



UNIVERSIDAD  
DE MÁLAGA

UNIVERSIDAD DE MÁLAGA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE  
TELECOMUNICACIÓN

TESIS DOCTORAL

VISION-BASED GESTURE  
RECOGNITION IN A ROBOT LEARNING  
BY IMITATION FRAMEWORK

AUTOR: Juan Pedro Bandera Rubio  
Ingeniero de Telecomunicación

2010



D. JUAN ANTONIO RODRÍGUEZ FERNÁNDEZ Y D. LUIS MOLINA-TANCO, PROFESORES DEL DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA DE LA UNIVERSIDAD DE MÁLAGA

CERTIFICAN:

Que D. Juan Pedro Bandera Rubio, Ingeniero de Telecomunicación, ha realizado en el Departamento de Tecnología Electrónica de la Universidad de Málaga, bajo nuestra dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

“VISION-BASED GESTURE RECOGNITION IN A ROBOT LEARNING BY IMITATION FRAMEWORK”

Revisado el presente trabajo, estimamos que puede ser presentado al Tribunal que ha de juzgarlo.

Y para que conste a efectos de lo establecido en la legislación vigente reguladora de los estudios de Tercer Ciclo-Doctorado, AUTORIZAMOS la presentación de esta Tesis en la Universidad de Málaga.

Málaga, 3 de Noviembre de 2009

Fdo. Juan Antonio Rodríguez Fernández  
Profesor de Tecnología Electrónica

Fdo. Luis Molina-Tanco  
Profesor de Tecnología Electrónica



Departamento de Tecnología Electrónica  
E. T. S. I. Telecomunicación  
Universidad de Málaga

TESIS DOCTORAL

VISION-BASED GESTURE RECOGNITION IN A  
ROBOT LEARNING BY IMITATION  
FRAMEWORK

AUTOR: Juan Pedro Bandera Rubio  
Ingeniero de Telecomunicación

DIRECTORES:

Juan Antonio Rodríguez Fernández  
Ingeniero de Telecomunicación  
Dr. por la Universidad de Málaga

Luis Molina-Tanco  
Ingeniero de Telecomunicación  
Dr. por la Universidad de Surrey



*A mi hermano*



# Agradecimientos / Acknowledgements

John Connor: Now, you gotta promise me you're not gonna kill anyone, right?  
Terminator: Right.  
John: Swear?  
Terminator: What?  
John: Just put up your hand and say, 'I swear I won't kill anyone.'  
Terminator: [Raises hand] I swear I will not kill anyone.

---

Como adelantaba la cita de más arriba, la robótica social abre un nuevo campo de aplicaciones, que por desgracia y como suele ocurrir, pueden estar orientadas a muy diversos fines. Quiero dar las gracias a todos los investigadores que se están esforzando para que estas aplicaciones sean, siempre, beneficiosas para la gente, y destinadas a hacernos vivir en un mundo más amable, cómodo y seguro.

Dicho ésto, los principales agradecimientos son, por supuesto, mucho más personales. En primer lugar, es una gran alegría ver que, en el transcurso de esta Tesis, he tenido la inmensa suerte de descubrir muy buenos amigos: Jesús Katrán, Rebeca Marfil, Pedro Núñez, Jose Manuel Pérez, Ricardo Vázquez-Martín,... Muchas gracias en primer lugar a todos ellos por el regalo de su amistad. Por supuesto, gracias también a los demás compañeros del laboratorio y del Departamento de Tecnología Electrónica. Es todo un placer trabajar junto a vosotros.

Por otro lado, he sido afortunado al tener como tutores no a uno, sino a dos buenos amigos: Juan Antonio Rodríguez Fernández y Luis Molina Tanco. Gracias a los dos por guiar una amalgama de ideas, publicaciones y notas hacia una Tesis de una pieza.

I would like to thank people from the ARICC centre, at Singapur, and from the ISR, at Coimbra, for their warm welcome and for the very good time I spent in these two really nice cities. Special thanks to Dr. Zhou Changjiu and Dr. Jorge Dias for their help, kindness and advices. I would also like to thank Professor Adrian Hilton and people from the CVSSP at Surrey for letting me use the Codamotion CX1 system, for all the help, and for make me feel like at home in England.

Por supuesto, sin mi familia para celebrar lo bueno, y para ayudarme en lo no tan bueno, todo esto hubiera resultado mucho más difícil. Aunque tengo la suerte de que seamos un montón y les estoy agradecido a todos, me gustaría en especial dar las gracias a mis padres, por todo lo que me han dado, a mi hermana por su alegría y, sobre todo, a mi hermano, Antonio Bandera, porque sin su ayuda esta Tesis no estaría aquí en estos momentos.

Finalmente a Rocío por quererme, por aguantarme... y por ser como es.



# Resumen

La robótica ha evolucionado en las últimas décadas hacia agentes cada vez más complejos, versátiles y capaces. Así, empiezan a aparecer robots capaces de trabajar en entornos cotidianos, dinámicos e imprevisibles. Es importante que dichos robots puedan interactuar de forma intuitiva y eficaz con las personas que encuentren en dichos entornos. Surge así el concepto de *robot social*, una de cuyas características principales es la capacidad de aprender de los demás. De entre los diferentes mecanismos de aprendizaje, el aprendizaje por imitación es una de las opciones más adecuadas para permitir a una persona suministrar fácilmente nuevos conocimientos al robot. Como quiera que el refuerzo verbal de las demostraciones realizadas en este tipo de aprendizaje puede no estar presente, es interesante contar con un mecanismo de aprendizaje por imitación basado puramente en visión. En esta Tesis, se presenta un sistema que implementa esta funcionalidad, permitiendo a un robot aprender gestos por imitación. La arquitectura de este sistema, en sí misma, supone la contribución fundamental de este trabajo, pues presenta alternativas y novedades tanto en su estructura como en la implementación de sus componentes.

La entrada sensorial del sistema está limitada a la información suministrada por un único par estéreo. A partir de estos datos, se extraen los movimientos de la parte superior del cuerpo de la persona usando un sistema propuesto en esta Tesis. Una vez estos movimientos han sido segmentados en gestos, son representados mediante una codificación basada en características tanto locales como globales. Dicha codificación mixta es otra de las aportaciones de esta Tesis. Una vez codificados, los gestos percibidos son comparados con el repertorio de gestos aprendidos por el robot. Las características locales se comparan utilizando algoritmos de programación dinámica, y los resultados obtenidos se refuerzan con la comparación de las características globales. Los resultados de estas comparaciones son utilizados por un componente de aprendizaje para valorar si se debe cambiar o no el repertorio de gestos aprendidos. Para solventar situaciones confusas, un cierto grado de supervisión por parte del usuario humano puede ser requerido.

Es importante destacar que todo el proceso anterior es independiente de las habilidades motoras del robot, lo que supone una importante variación de esta arquitectura frente a otras propuestas. Como consecuencia, las características físicas del robot no limitan la percepción, con lo que los movimientos de la persona que realiza las demostraciones pueden codificarse, reconocerse y aprenderse de manera más precisa. Sólo en aquellos casos en que se requiera que el robot imite físicamente los movimientos percibidos o aprendidos interviene el último módulo de la arquitectura, que traduce los gestos del espacio de movimientos de la persona al del robot, usando una estrategia combinada.

Los resultados experimentales incluyen una evaluación cuantitativa de los módulos de captura de movimientos y traducción. Por otro lado, tanto los métodos de representación como de reconocimiento han sido contrastados con otras alternativas y evaluados por separado. Finalmente, se han llevado a cabo pruebas del sistema completo en entornos cotidianos reales, que demuestran su validez de cara a ser utilizado en un robot social.



# Abstract

Robotics research has evolved in the last decades towards agents more complex, versatile, and capable. Thus, robots begin to appear that are able to work in dynamic, unpredictable environments such as houses, restaurants or museums. It is important for these robots to interact with people in their surroundings in an easy and efficient way. These requirements lead to the concept of *social robot*. One of the characteristics of such a robot is its required ability to learn from others. In this sense, learning by imitation appears as one of the most powerful mechanism a robot can use to learn socially from a human teacher. Speech learning reinforcement is useful but not always available, thus it may be worthy to consider a learning by imitation system that is based only in vision. This thesis presents a system that meets these requirements. The system itself is one of the main contributions, as its architecture includes novel concepts both in its structure and the implementation of its components.

The sensory input for the proposed system is restricted to the data provided by a pair of stereo cameras. The upper-body movements of the human performer are extracted from these data using a novel human motion capture system. Once captured, human motion is segmented into discrete gestures. Then, these gestures are codified using a proposed representation based on local and global features. Encoded perceived gestures are compared against the gesture repertoire of the robot, that contains all gestures the agent has already learnt. Local features are compared using dynamic programming alignment techniques. These local results are reinforced by the comparison of global features, performed using analytic algorithms. The final results are fed to a learning component, that determines whether the gesture repertoire should be modified or not. Uncertain situations are solved by asking for a small degree of human supervision.

It is important to consider that all previous modules are independent from the particular motor abilities of the used robot. This is an important difference respect to other approaches. In the proposed architecture, the physical characteristics of the robot do not constraint its perceptual abilities, thus perceived human movements can be represented, recognized and learnt more precisely. The last module of the architecture translates motion from the human motion space to the robot one, using a combined strategy. It is employed only when imitation of perceived of learnt gestures is required.

Experimental results presented in this thesis include a quantitative evaluation of the human motion capture and translation modules. Besides, representation and recognition methods have been compared against other alternatives and evaluated independently. The last experiments involved the complete system working in real environments. These tests validate the proposed system as an interesting element to be integrated in a social robot.



# Contents

<b>I</b>		<b>1</b>
	Resumen de la Tesis Doctoral . . . . .	3
	Conclusiones . . . . .	29
<b>II</b>		<b>1</b>
<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Motivation . . . . .	5
1.1.1	Social robots . . . . .	6
1.1.2	Constraints imposed by social environments . . . . .	9
1.2	Objectives of the thesis . . . . .	10
1.3	Contributions of the Thesis . . . . .	12
1.4	Organization of the Thesis . . . . .	13
<b>2</b>	<b>System architecture</b>	<b>15</b>
2.1	Outline of the chapter . . . . .	15
2.2	Introduction to Social Learning . . . . .	16
2.2.1	Culture . . . . .	16
2.2.2	Cultural evolution. Memes . . . . .	17
2.2.3	Individual learning . . . . .	17
2.2.4	Social learning . . . . .	18
2.2.5	Instinctive imitation . . . . .	19
2.2.6	Operant conditioning . . . . .	20
2.2.7	Cognitive imitation . . . . .	20
2.2.7.1	Attention . . . . .	21
2.2.7.2	Memorization . . . . .	21
2.2.7.3	Generation . . . . .	22
2.2.7.4	Motivation . . . . .	22
2.2.8	Factors that influence social learning . . . . .	23
2.3	Evolution of artificial social learning . . . . .	25
2.3.1	Early works. Program by Demonstration (PbD) . . . . .	25
2.3.2	Generalization of learned tasks . . . . .	26
2.3.3	Humanoid robots. From PbD to RLbI . . . . .	28
2.4	RLbI . . . . .	30
2.4.1	Schaal’s proposal . . . . .	33
2.4.1.1	Implementation . . . . .	34
2.4.2	Active imitation. Demiris and Hayes . . . . .	35
2.4.2.1	Implementation . . . . .	38

2.4.3	Breazeal’s architecture for a social robot . . . . .	38
2.4.3.1	Implementation . . . . .	40
2.4.4	The interactive perception filter of Mohammad and Nishida . . . . .	41
2.4.4.1	Implementation . . . . .	42
2.4.5	The task-level imitation learning system of Mülihg et al. . . . .	43
2.4.5.1	Implementation . . . . .	44
2.5	First approach to the considered RLbI system. Discussion . . . . .	45
2.6	Proposed RLbI system . . . . .	48
2.7	Map of the thesis . . . . .	50
<b>3</b>	<b>Human motion perception</b> . . . . .	<b>53</b>
3.1	Outline of the chapter . . . . .	53
3.2	Overview of the human motion capture system . . . . .	56
3.3	Face detection . . . . .	60
3.4	Silhouette extraction . . . . .	61
3.5	Pose estimation . . . . .	70
3.5.1	Anthropometry . . . . .	71
3.5.2	Human geometric model . . . . .	72
3.5.3	Estimation of torso pose . . . . .	77
3.5.3.1	Torso pose estimation using anthropometric relations . . . . .	78
3.5.4	Estimation of arms pose . . . . .	83
3.5.4.1	Hand trajectories preprocessing . . . . .	84
3.5.4.2	Inverse kinematics algorithm . . . . .	84
3.5.4.3	Detection of incorrect poses . . . . .	85
3.5.4.4	Avoidance of incorrect poses . . . . .	90
3.6	Stereo cameras mounted on the proposed RLbI system . . . . .	99
3.7	Evaluation of the HMC system . . . . .	100
3.7.1	Experimental setup . . . . .	100
3.7.2	Error measurement . . . . .	104
3.7.2.1	Ground-truth extraction . . . . .	106
3.7.2.2	Comparison of captured data against ground-truth . . . . .	111
3.8	Conclusion . . . . .	115
<b>4</b>	<b>Gesture representation, recognition and learning</b> . . . . .	<b>117</b>
4.1	Outline of the chapter . . . . .	117
4.2	State of the art . . . . .	118
4.2.1	Gesture representation . . . . .	119
4.2.2	Gesture recognition . . . . .	122
4.2.3	Gesture reconstruction and learning . . . . .	123
4.3	Gesture segmentation . . . . .	125
4.4	Reduction to a latent space of human motion . . . . .	127
4.5	Proposed approach . . . . .	128
4.6	3D trajectory representation . . . . .	131
4.6.1	Extraction of dominant points . . . . .	132
4.6.2	Validation . . . . .	136
4.7	Trajectory matching . . . . .	138
4.7.1	Local distance computation . . . . .	138
4.7.2	Distance functions evaluation . . . . .	140

4.7.3	Global features	143
4.7.4	Confidence Reinforcement	144
4.7.5	Reinforced confidence value evaluation	145
4.8	Knowledge update	147
4.9	Comparison between PCA+LDA and DPD+DTW	150
4.10	Conclusion	154
<b>5</b>	<b>Motion translation</b>	<b>157</b>
5.1	Outline of the chapter	157
5.2	The correspondence problem	158
5.3	Combined retargeting	162
5.3.1	Position retargeting	164
5.3.2	Angle retargeting	165
5.3.3	Combined retargeting	166
5.4	Evaluation of the retargeting module	168
5.5	Conclusion	171
<b>6</b>	<b>Testing the learning by imitation system</b>	<b>175</b>
6.1	Outline of the chapter	175
6.2	Robotic platforms used in testing	176
6.2.1	HOAP-1	177
6.2.2	NOMADA	179
6.2.2.1	Locomotion platform	180
6.2.2.2	Articulated arms	180
6.2.2.3	Perceptual system	180
6.3	Experimental setup	182
6.4	Experimental results	186
6.4.1	Gesture representation	186
6.4.2	Gesture recognition	187
6.4.3	Gesture learning by imitation	192
6.4.3.1	Gesture segmentation	192
6.4.4	Gesture learning	194
<b>7</b>	<b>Discussion and future work</b>	<b>203</b>
7.1	Outline of the chapter	203
7.2	Characteristics of the proposed RLbI architecture	203
7.3	Performance and limitations of proposed vision-based perception system	205
7.4	Usage of global features	206
7.5	Evaluation of proposed learning system	206
7.6	Further work	207
7.6.1	Increasing perceptual capabilities	207
7.6.2	Considering additional global features for gesture recognition	208
7.6.3	Using a more versatile learning module	208
7.6.4	Completing the construction of NOMADA and test the system in a working social robot	209
7.6.5	Integrating the proposed RLbI system in a higher level architecture	209
<b>8</b>	<b>Conclusion</b>	<b>211</b>

<b>A</b>	<b>Vision-based perceptual systems</b>	<b>233</b>
A.1	Outline . . . . .	233
A.2	Non-vision based perception systems . . . . .	233
A.3	Vision based perception systems . . . . .	234
<b>B</b>	<b>Face detection</b>	<b>241</b>
B.1	Features . . . . .	243
B.2	Integral image . . . . .	244
B.3	Classifier learning process: Adaboost algorithm . . . . .	244
<b>C</b>	<b>Canny edge detector</b>	<b>247</b>
<b>D</b>	<b>Tracking using BIPs</b>	<b>251</b>
D.1	Oversegmentation . . . . .	254
D.2	Template matching and target refinement . . . . .	254
D.3	Template updating . . . . .	255
D.4	Region of interest updating . . . . .	256
<b>E</b>	<b>FIR filter employed to smooth perceived hand motion</b>	<b>257</b>
<b>F</b>	<b>Inverse kinematics algorithm</b>	<b>259</b>
<b>G</b>	<b>Publications of the author</b>	<b>261</b>
G.1	Publications covered in this thesis . . . . .	261
G.2	Publications not covered in this thesis . . . . .	266

# Acronyms

**AIM** Active Intermodal Mapping.

**BIP** Bounded Irregular Pyramid.

**CMOS** Complementary Metal Oxide Semiconductor.

**CR** Compression Rate.

**CSS** Curvature Scale Space.

**DOF** Degree Of Freedom.

**DP** Dynamic Programming.

**DPD** Dominant Points Detector.

**DTW** Dynamic Time Warping.

**ED** Edit Distance.

**EDR** Edit Distance on Real Sequence.

**EMD** Earth Mover's Distance.

**ERP** Edit Distance with Real Penalty.

**FIR** Finite Impulse Response.

**FK** Forward Kinematics.

**FOM** Figure Of Merit.

**FOV** Field Of View.

**HMC** Human Motion Capture.

**HMI** Human Motion Imitation.

**HMM** Hidden Markov Model.

**HRI** Human Robot Interaction.

**ISE** Integral Square Error.

**IK** Inverse Kinematics.

- LDA** Linear Discriminant Analysis.
- LCSS** Longest Common Subsequences.
- LWPR** Locally Weighted Projection Regression.
- ML** Machine Learning.
- PbD** Program by Demonstration.
- PCA** Principal Component Analysis.
- RLbI** Robot Learning by Imitation.
- ROI** Region Of Interest.
- RNN** Recurrent Neural Network.
- SNM** Semantic Network Model.
- SVM** Support Vector Machine.
- VGA** Video Graphics Array.

# List of Figures

## RESUMEN DE LA TESIS DOCTORAL

### TESIS DOCTORAL

2.1	Different components that can be identified in a RLbI system. . . . .	32
2.2	Conceptual sketch of an imitation learning system (Schaal, 1999). . . . .	33
2.3	Biologically-plausible model for a learning by imitation system, proposed by Demiris and Hayes (2002). . . . .	37
2.4	Social robot cognitive architecture for learning and performing tasks and motor skills (Breazeal et al., 2004). . . . .	39
2.5	System architecture for a social robot presented in Mohammad and Nishida (2009). . . . .	41
2.6	Architecture for task-level imitation learning proposed by Mühlrig et al. (2009). . . . .	44
2.7	Architecture of the RLbI system proposed in Bandera et al. (2007). . . . .	45
2.8	System architecture. . . . .	51
2.9	System architecture, showing the chapters that describe each part. . . . .	52
3.1	Flow diagram of the Human Motion Capture (HMC) system. . . . .	59
3.2	Face detection in uncontrolled indoor environments ( $k = 5$ ). . . . .	62
3.3	Three examples of input images from the left camera. . . . .	64
3.4	Disparity images corresponding to left input images shown in Fig. 3.3. . . . .	64
3.5	Results obtained when Eq. 3.1 is applied to disparity maps depicted in Fig. 3.4. . . . .	64
3.6	Silhouettes extracted applying connected components to maps in Fig. 3.5. . . . .	65
3.7	Two examples of filtered disparity maps. The silhouettes extracted if only connected components are applied are marked as red regions. . . . .	65
3.8	Canny borders extracted from the disparity maps depicted in Fig. 3.4. . . . .	66
3.9	Regions in which silhouette borders of people depicted in Fig. 3.3 are searched. . . . .	67
3.10	Canny borders located inside the search regions depicted in Fig. 3.9. . . . .	67
3.11	Final results obtained by the proposed method for the disparity maps depicted in Fig. 3.7. . . . .	69
3.12	Silhouettes corresponding to input images from the left camera depicted in Fig. 3.3. . . . .	69
3.13	Silhouettes corresponding to input images from the left camera depicted in Fig. 3.3, containing skin color regions associated to human head and hands. . . . .	70
3.14	Vitruvian Man, drawing by Leonardo da Vinci (around 1487). . . . .	72
3.15	Anthropometric values used in this thesis. . . . .	73
3.16	(a) Illustration of the human upper-body kinematic model; and (b) human upper-body kinematic model showing the triangles used to model each mesh. . . . .	74

3.17	Scene graph used by the upper-body human model. The skeleton is composed by the ellipsoidal grey nodes. . . . .	75
3.18	Skeleton and DOFs for the upper torso of the human model. . . . .	76
3.19	Lateral flexion of human torso. . . . .	80
3.20	Forward/backwards flexion of human torso. . . . .	80
3.21	Geometric system showing the center points of the shoulder search regions. . . .	81
3.22	Torso medium axes extracted from the silhouette images depicted in Fig. 3.12. . .	83
3.23	Estimated torso medium axis and shoulder locations for the performers depicted in Fig. 3.3. . . . .	83
3.24	Coefficients of the FIR filter used to smooth one of the coordinates of a hand trajectory. . . . .	84
3.25	Movement of the torsion DOF along segment axis. Both configurations are equivalent as long as arm segment has cylindrical symmetry. . . . .	87
3.26	Consequences of the movement of the torsion DOF along segment axis if the arm is not symmetric: (a) Initial pose; (b) Right arm rotation (torsion DOF located in the shoulder); and (c) Right arm rotation (torsion DOF located in the elbow). . . . .	88
3.27	Collision check using projections of the OBBs over a separating axis. . . . .	89
3.28	RAPID collision detection: (a) Valid pose. (b) Collision. . . . .	89
3.29	Alternative elbow locations ( $N = 40$ , $\theta_{max} = 3\pi/4$ ) for two different hand positions. . . . .	92
3.30	Alternative elbow locations ( $N = 20$ , $\theta_{max} = 3\pi/4$ ) for two different hand positions. . . . .	92
3.31	Alternative elbow locations ( $N = 10$ , $\theta_{max} = 3\pi/4$ ) for two different hand positions. . . . .	93
3.32	Alternative elbow locations ( $N = 20$ , $\theta_{max} = \pi/2$ ) for two different hand positions. . . . .	94
3.33	Alternative elbow locations ( $N = 20$ , $\theta_{max} = \pi/8$ ) for two different hand positions. . . . .	94
3.34	Alternative elbow locations ( $N = 30$ , $\theta_{max} = \pi/8$ ) for two different hand positions. These are the parameters selected for most of the experiments. . . . .	95
3.35	Example of a virtual model tracking a perceived right arm movement (red spheres) that lies beyond its limits. Valid joint angles are not updated. As depicted, right arm motion stops until a valid pose is provided. . . . .	96
3.36	Example of a virtual model tracking a perceived right arm movement (red spheres) that lies beyond its limits. Valid joint angles are updated. It can be seen that the right arm motion tries to follow perceived position even when it lays beyond model's reachable workspace, thus errors are reduced respect to the ones obtained in Fig. 3.35. . . . .	97
3.37	Update algorithm to obtain more natural and efficient poses: (a) Initial human model pose; (b) the right arm moves to the left. Alternative elbow positions are adopted to follow the movement; (c) the right arm moves to the right. The elbow is located in the valid pose nearest to the previous one; and (d) the system looks for alternatives that locate the elbow in a lower vertical positions to obtain a more natural and efficient pose. . . . .	98
3.38	Pair of stereo cameras mounted on the HOAP-1 robot. . . . .	99
3.39	STH-DCSG-VAR-C cameras, Videre Design. . . . .	100
3.40	Codamotion CX1 motion capture system: a) CX1 camera unit; and b) Codamotion infra-red markers and one drive box used to power them. . . . .	102
3.41	Experimental setup used to evaluate the vision-based HMC system. . . . .	103
3.42	a) Left frame of a sequence captured under normal indoor lighting conditions; and b) Left frame captured during system evaluation. . . . .	104

3.43	a) the stereo vision system STH-DCSG-VARX from Videre Design and the CODA motion capture system at the Centre for Vision, Speech and Signal Processing, at the University of Surrey; and b) distribution of markers. . . . .	105
3.44	a) Upper-body human model with attached virtual markers (green spheres); and b) real (red spheres) and virtual (grey spheres) markers for a perceived frame. . .	105
3.45	Positions of the CODA marker located in the left shoulder during a test movement (handshake) lasting 40 seconds. . . . .	106
3.46	Results obtained after applying outlier removal to the movement shown in Fig. 3.45. . . . .	107
3.47	Positions of one of the CODA markers located in the right wrist for the movement depicted in 3.45. . . . .	107
3.48	Results obtained after applying outlier removal to the movement shown in Fig. 3.47. . . . .	108
3.49	Comparison between real CODA markers (red dots) and virtual markers attached to the 3D human model (grey dots): a) before spatial alignment; and b) after spatial alignment. . . . .	112
3.50	Interpolated positions of the virtual marker located in the left shoulder for the movement depicted in Fig. 3.45. . . . .	113
3.51	Interpolated positions of the virtual marker located in the right wrist for the movement depicted in Fig. 3.47. . . . .	114
3.52	Comparison between Codamotion CX1 and the proposed vision-based HMC system. Mean errors and standard deviations associated to the different markers. . .	114
3.53	Percentage of visibility for the CODA markers during execution of test sequences. . . . .	115
3.54	Comparison between Codamotion CX1 and the proposed vision-based HMC system. Differences between position estimated from both capture systems for a virtual marker located: a) near the image center; and b) near the image border (red, green and blue lines show the differences in the x, y and z coordinates, respectively). . . . .	116
4.1	Overview of the proposed gesture recognition system. . . . .	129
4.2	Calculation of the maximum length of trajectory presenting no significant discontinuity on the right side of point $i$ ( $K_f[i]$ ). The graph shows different values for $\sum l_j - dE$ , where $\sum l_j = l(i, i + K_f[i])$ and $dE = d(i, i + K_f[i])$ . In this case, the chosen $K_f[i]$ value is equal to 4.0. . . . .	133
4.3	a-c) $\kappa_{XZ_i}$ values associated to three different demonstrations of an example gesture composed by only one trajectory. . . . .	135
4.4	a-b) Curvature-based descriptors associated to typical gestures. . . . .	135
4.5	Dominant points extraction using different curvature functions. (a) $k=50$ ; (b) Adaptive curvature; and (c) $k=500$ . . . . .	137
4.6	Upper-body social gestures used to test the gesture recognition and learning system. The trajectories of the left and right hands have been marked over the frontal view of a 3D model of the human performer. . . . .	142
4.7	Effects concerning the use of a matching threshold: (a) Blue and red trajectories are considered to match perfectly; and (b) Due to a small global variation in the red trajectory, blue and red trajectories are now considered completely different. . . . .	143
4.8	Global similarities between different gestures. . . . .	146
4.9	Dataflow of the knowledge update algorithm employed to test the RLBI system in real scenarios. . . . .	148

4.10	a-b) PCA projections of the gesture descriptors in Figs. 4.4a-b	152
5.1	Effects of different environments on the same behaviour -lay the box on the table- for two individuals that share the same embodiment.	160
5.2	Illustration of the combined retargeting approach of Shin et al. (2001): (a) Joint angles preserved in absence of external objects; and (b) End-effector positions preserved when external objects are close.	162
5.3	Combined retargeting system.	163
5.4	Position retargeting.	165
5.5	3D models showing the local coordinate frames, the left shoulder position and the length of the stretched left arm for: (a) Human model; and (b) Robot model.	165
5.6	Angle retargeting.	165
5.7	Example of angle retargeting function.	166
5.8	$\alpha$ values ( $d=50$ cm.).	168
5.9	Position errors (Right hand): (a) $\alpha = 0.0$ (position retargeting); (b) $\alpha = 1.0$ (angle retargeting); and (c) $\alpha = 0.5$ (combined retargeting).	170
5.10	Joint angle errors (Right arm): (a) $\alpha = 0.0$ (position retargeting); (b) $\alpha = 1.0$ (angle retargeting); and (c) $\alpha = 0.5$ (combined retargeting).	171
5.11	Perceived movement retargeted to two different robotic platforms using different retargeting strategies: (a) $\alpha = 1.0$ (angle retargeting); (b) $\alpha = 0.0$ (position retargeting); and (c) dynamic $\alpha$ (combined retargeting).	172
6.1	HOAP-1 humanoid robot.	177
6.2	DOFs for the upper torso of the HOAP-1.	178
6.3	Virtual model of the NOMADA social robot.	179
6.4	NOMAD 200 from Nomadic.	181
6.5	DOFs for the upper torso of the NOMADA.	181
6.6	(a) Virtual model of the NOMADA arm; and (b) real NOMADA arm mounted by the ISIS research group at Malaga University.	182
6.7	Biclops pan-tilt-vergence head.	183
6.8	Upper-body social gestures used to test the proposed RLbI system. The trajectories of the left and right hands have been marked over the left frame.	184
6.9	Real indoor scenarios used to test the proposed HMC system.	185
6.10	Trajectories of the right hand, captured for the same gesture using the Codamo- tion system and the vision-based system, respectively.	188
6.11	Trajectories of the right hand, captured for the same gesture using the Codamo- tion system and the vision-based system, respectively.	189
6.12	Right hand XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.	194
6.13	Left hand XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.	195
6.14	Head XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.	195
6.15	Right shoulder XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.	196
6.16	Left shoulder XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.	196

6.17	Human performer facing the stereo cameras and executing a gesture while the NOMADA robotic arm imitates his right arm movements. . . . .	199
B.1	Haar-like features used to detect faces: (a) Vertical two-rectangle; (b) horizontal two-rectangle; (c) three rectangle; and (d) four-rectangle. . . . .	243
B.2	Memory accesses needed to compute: (a) a two-rectangle feature; (b) a three-rectangle feature; (c) a four-rectangle feature. For example the two-rectangle feature is computed as: $B - A = (5 - 6 - 3 + 4) - (3 - 4 - 1 + 2)$ . . . . .	244
C.1	Directions in which the Sobel argument is grouped. . . . .	248
C.2	Search regions used to obtain $g_h$ value, depicted over: (a) left input image; and (b) disparity map. . . . .	249
D.1	Binary BIP level generation: (a) regular step (non-orphan vertices of level $l$ have been marked); (b) parent search; (c) intra-level twining at level $l$ ; and (d) intra-level edge definition at level $l + 1$ (marked in black). . . . .	253
D.2	Levels of a template, modeled as a BIP structure, that represents a tracked hand. . . . .	253
D.3	Data flow of the tracking algorithm. . . . .	254
E.1	Coefficients of the FIR filter used to smooth one of the coordinates of a hand trajectory. . . . .	258
F.1	Kinematic model of the human arm showing local coordinate frames and elbow circle. . . . .	260



# List of Tables

## RESUMEN DE LA TESIS DOCTORAL

1	Promedio de los errores de posición, obtenido a partir de más de 5300 imágenes.	19
---	---	----

## TESIS DOCTORAL

3.1	Number of triangles used to model each body part.	73
3.2	Anthropometric values used to constraint the torso and shoulder search regions.	79
3.3	Collision pairs defined for the human model depicted in Fig. 3.16.	90
3.4	Main specifications of the STH-DCSG-VAR-C stereo cameras.	101
3.5	Tracking errors averaged over 5300 frames.	113
4.1	Compression rates and execution times for different trajectories.	137
4.2	Distance functions (Chen et al. 2005).	140
4.3	Evaluation of different distance functions to measure local similarity between gestures represented as sets of trajectories.	141
4.4	Evaluation of different distance functions to measure local similarity between gestures represented as sets of trajectories, including global reinforcement.	147
4.5	Evaluation of the different recognition approaches.	154
5.1	Right arm mean errors ( $E(q)$ ) and standard deviations ( $\sigma_q$ ) for an imitation sequence performed by HOAP-1 robot.	169
5.2	Right arm mean errors ( $E(q)$ ) for dynamic gestures imitated by NOMADA robot.	170
5.3	Right arm mean errors ( $E(q)$ ) for static gestures imitated by NOMADA robot.	171
6.1	Description of the social gestures performed to test the RLbI system.	184
6.2	Compression rates and execution times for different trajectories.	187
6.3	Evaluation of different distance functions. The motion has been perceived using the proposed vision-based motion capture system.	190
6.4	Deviation values obtained for different gesture datasets.	191
6.5	Confusion matrices before correcting stored gestures. Motion perceived using a Codamotion CX1 system.	200
6.6	Confusion matrices before correcting stored gestures. Motion perceived using the vision-based motion capture system presented in Bandera et al. (2006).	200
6.7	Confusion matrices after correcting stored gestures. Motion perceived using a Codamotion CX1 system.	201
6.8	Confusion matrices after correcting stored gestures. Motion perceived using the vision-based motion capture system presented in Bandera et al. (2006).	201

6.9	Description of the gestures given to performers testing the gesture learning by imitation system. . . . .	202
6.10	Quantitative temporal evaluation of the proposed RLbI system. . . . .	202
B.1	Percentages of rightly classified faces, false positives and false negatives. . . . .	245

# Part I



Departamento de Tecnología Electrónica  
E. T. S. I. Telecomunicación  
Universidad de Málaga

RESUMEN DE LA TESIS DOCTORAL

VISION-BASED GESTURE RECOGNITION IN A  
ROBOT LEARNING BY IMITATION  
FRAMEWORK

AUTOR: Juan Pedro Bandera Rubio  
Ingeniero de Telecomunicación

DIRECTORES:

Juan Antonio Rodríguez Fernández  
Ingeniero de Telecomunicación  
Dr. por la Universidad de Málaga

Luis Molina-Tanco  
Ingeniero de Telecomunicación  
Dr. por la Universidad de Surrey



# Resumen de la Tesis Doctoral

En esta primera parte del presente documento se expone un resumen, escrito íntegramente en español, de la Tesis titulada: “Vision-based gesture recognition in a robot learning by imitation framework”. A lo largo de las siguientes secciones se irán describiendo de forma general los distintos elementos que componen el sistema propuesto. Se presentarán resultados que ayuden a evaluar la validez y alcance de las soluciones propuestas. Para una descripción más amplia y profunda de los temas abordados en esta Tesis, la segunda parte de este documento amplía la información presentada en este resumen, incluyendo las formulaciones matemáticas empleadas, y descripciones detalladas de las pruebas realizadas y los resultados obtenidos.

## 1 Introducción

La investigación en el campo de la robótica está realizando en las últimas décadas un gran esfuerzo por llevar los robots a la vida cotidiana. Así, se pretende evolucionar desde los robots industriales, encargados de tareas repetitivas realizadas en entornos controlados, a otro tipo de agentes, más parecidos a la visión original de Čapek (1920), el escritor de fantasía que acuñó el término *robot*. El objetivo para estos nuevos agentes es que puedan trabajar en entornos cotidianos no controlados, tales como un restaurante, un museo, unas oficinas, una casa o incluso en las calles de una ciudad. La naturaleza de las tareas que se proponen para estos robots también cambia: así, se pretende que estos agentes sean capaces de colaborar e interactuar con personas de una manera natural e intuitiva, y que puedan también adaptarse a nuevas situaciones, o aprender a realizar nuevas tareas.

Una de las grandes cuestiones que surgen en el camino hacia esta nueva generación de robots es la del aprendizaje. La opción de programar *a priori* al robot se descarta de partida, pues es imposible predecir la variedad de situaciones que un robot puede encontrar en entornos sociales no controlados. Es por tanto necesario dotar al agente de un mecanismo que le permita aprender de la experiencia, de la observación. Dicho mecanismo debería también permitir a

las personas enseñar al robot nuevas tareas o comportamientos de manera fácil e intuitiva. Uno de los mecanismos más poderosos para satisfacer estas cuestiones es el del Aprendizaje por Imitación (o Robot Learning by Imitation (RLbI)) (Schaal, 1999). Sin embargo, a pesar de que las investigaciones en este campo han sido intensas en la última década, existen serias dificultades a la hora de crear un sistema práctico, que permita a un robot aprender en los entornos previamente mencionados, de personas no familiarizadas con su uso, y usando sólo sus sistemas de percepción.

Esta Tesis aborda la implementación de un sistema completo de aprendizaje de gestos basado en visión estéreo. Dicho sistema percibirá, aprenderá e imitará movimientos de la parte superior del cuerpo de la persona -sin considerar expresiones faciales o gestos realizados con los dedos-, satisfaciendo las necesidades previamente expuestas, en escenarios no controlados. Antes de formalizar estos objetivos, se van a describir los robots sociales, y a enumerar las restricciones típicas de los escenarios en que dichos agentes se utilizan.

## 1.1 Robots sociales

A pesar de la utilidad de los robots industriales (Craig, 1986) y teleoperados (Ohya et al., 2009), el concepto primigenio de robot consideraba agentes mucho más versátiles y capaces. Así, los robots de ficción descritos por Čapek (1920) y otros autores son entidades que perciben y trabajan en entornos complejos y cambiantes. Estos agentes reconocen objetos y personas, y pueden discernir lo que dichas personas están haciendo. También pueden interactuar con personas y robots de una forma natural e intuitiva, utilizando la voz, el tacto y, sobre todo, la visión. Pueden cooperar para ayudar a resolver tareas complejas. Y pueden adaptarse a nuevas situaciones, para lo cual utilizan mecanismos de aprendizaje por observación, por demostración o incluso por revisión interna de conductas previamente aprendidas.

La utilidad de un 'compañero robótico' que fuese capaz de exhibir todas las características previamente mencionadas está fuera de toda duda. Dejando a un lado los robots imaginados en fantasías futuristas, lo cierto es que la rama de la robótica en que se desarrolla esta Tesis tiene como objetivo, precisamente, la consecución práctica de este tipo de robots, a los que se ha dado en llamar *robots sociales*.

El término *robot social* puede resultar excesivamente vago y, de hecho, Walter (1950) lo empleó hace ya más de cincuenta años para denominar robots incapaces, en realidad, de desempeñar ninguna de las funciones previamente comentadas. Décadas después, Dautenhahn and Billard (1999) formularon una definición de robot social que sigue siendo, hoy día, muy em-

pleada. Según esta definición *“los robots sociales son agentes que forman parte de una sociedad heterogénea: una sociedad de robots o humanos. Estos robots sociales son capaces de reconocer a los miembros de esta sociedad, y de establecer relaciones sociales con ellos. También poseen historias (perciben e interpretan el mundo según su experiencia), y se comunican y aprenden de los demás de forma explícita”*.

Más recientemente, con el desarrollo de nuevas aplicaciones y plataformas robóticas, se han propuesto diferentes categorías de “comportamiento social” que un robot puede exhibir. Así, Breazeal et al. (2003) distingue entre: (a) robots *socialmente evocativos*, con capacidades de percepción y movimiento muy limitadas; (b) robots que proporcionan un *interfaz social* pero son incapaces de aprender o cooperar con las personas; (c) robots *socialmente receptivos*, capaces de aprender, pero carentes de motivaciones u objetivos propios; y (d) robots *sociables*, definidos como *“robots capaces de comunicarse e interactuar con nosotros, comprendernos y relacionarse con nosotros de una forma personal. Deberían ser también capaces de entendernos en términos sociales. Por nuestra parte, deberíamos poderlos entender en esos mismos términos”* (Breazeal, 2002).

Como se ve, esta definición resulta menos genérica que la de Dautenhahn and Billard (1999), pues considera sólo robots que interactúan con personas (y excluye, por tanto las sociedades constituidas exclusivamente por robots). Lo cierto es que la inmensa mayoría de robots sociales están diseñados para trabajar en sociedades humanas, en las que la presencia de otros robots es testimonial, en el mejor de los casos. Al igual que otros autores como Fong et al. (2003), esta Tesis sigue esta particularización, y propone la siguiente definición de robot social: *los robots sociales son robots que trabajan en entornos sociales reales, y que son capaces de percibir, interactuar con y aprender de otros individuos, que suelen ser personas pero pueden ser, también, otros robots sociales*.

## 1.2 Escenarios de RLbI

Los entornos en los que un robot social debe actuar poseen determinadas características. La mayor parte de ellas representan, *de facto*, restricciones que deben tenerse en cuenta cuando se diseña uno de estos robots. En esta Tesis se consideran las siguientes:

- *Variabilidad estructural*. Nuevos objetos pueden aparecer en el entorno. Los objetos existentes pueden desplazarse, sustituirse o incluso ser eliminados.
- *Iluminación dinámica*. Las fuentes de luz pueden aparecer y desaparecer de forma impre-

vista. Las características de las luces tampoco están controladas y son variables.

- *Distancia entre la persona y el robot desconocida.* La distancia a la que se establecen las interacciones entre el robot y las personas de su entorno no es fija. El robot no puede imponer a las personas que se sitúen a cierta distancia para interactuar con él.
- *Conocimiento previo limitado.* El robot social debe adaptarse a nuevas situaciones y aprender gestos, tareas y comportamientos. Por tanto, aunque en casos concretos puede proporcionarse al agente cierto conocimiento previo, éste debe poder ser extendido o modificado según sus experiencias.
- *Percepción limitada.* La cantidad, variedad y calidad de los sensores que se pueden montar en un robot social es limitada.
- *No invasividad.* No se debe imponer a la gente que interactúa con un robot social que se coloquen marcadores especiales en el cuerpo, ni tampoco prendas específicas o parches de color. La interacción debe ser natural, intuitiva y cómoda.
- *Usuarios no entrenados.* No se debe exigir un entrenamiento especial para utilizar un robot social. Por el contrario, para comunicarse con uno de estos agentes una persona debería poder usar los mismos mecanismos que utiliza para comunicarse con otras personas. En este sentido, se refuerza la idea de utilizar el aprendizaje por demostración (o por imitación) como método para enseñar nuevos comportamientos a un robot social.
- *Usuarios imprevistos.* Aunque su comportamiento puede variar cuando se relaciona con personas desconocidas, lo cierto es que un robot social debe considerar la posibilidad de que nuevos usuarios aparezcan en el entorno e intenten comunicarse con él.
- *Respuesta rápida.* Las personas esperan recibir realimentación y contestaciones rápidas de sus interlocutores. Un agente que tarde demasiado en proporcionarlas producirá rápidamente una pérdida de interés en las personas que se relacionen con él. Por tanto, es necesario para un robot social ser capaz de respetar los ritmos utilizados en interacciones sociales entre personas.

En ocasiones, es posible reducir o eliminar alguna de estas restricciones. Por ejemplo, los trabajos que se centran en el estudio del aprendizaje pueden considerar que el robot cuenta con datos completos y fidedignos del movimiento de una persona (Calinon, 2007). En esta Tesis, sin embargo, se pretenden considerar todas las restricciones o, al menos, las principales, de cara a obtener un robot que pueda ser utilizado en la práctica como robot social. A lo largo de esta

Tesis, cuando se hable de “escenarios de RLbI”, se estarán designando entornos en los que hay que considerar todas las restricciones anteriores.

### 1.3 Objetivos de la Tesis

La voz y los gestos juegan un papel crucial en las interacciones sociales y en el aprendizaje social. Sin embargo, en el caso concreto del aprendizaje por imitación la comunicación verbal es muchas veces usada, simplemente, para reforzar las demostraciones prácticas visuales. De hecho, en ocasiones la demostración visual de una tarea es suficiente para que dicha tarea pueda ser imitada y aprendida. Por tanto, es interesante desarrollar un sistema de aprendizaje por imitación que utilice sólo información visual. La implementación de un sistema de estas características es el principal objetivo de esta Tesis. Más concretamente, en esta Tesis se implementa un *sistema de aprendizaje por imitación de gestos sociales basado en visión*.

Para alcanzar este objetivo general es necesario afrontar una serie de objetivos parciales que conciernen a diferentes etapas del proceso, desde la percepción de los movimientos de una persona hasta el envío de órdenes a los motores del robot. Estos objetivos se enumeran a continuación:

- Encontrar un sistema de percepción visual que pueda montarse en un robot social, y que pueda percibir personas que se encuentren a distancias típicas en interacciones sociales.
- Implementar un sistema que pueda extraer, de estas imágenes, el movimiento de una persona, y descomponerlo en gestos.
- Codificar los gestos usando una representación simplificada, de baja dimensionalidad. Implementar un método que permita reconocer estos gestos de forma rápida y eficaz.
- Definir una forma adecuada de trasladar el movimiento percibido de la persona al robot social.
- Implementar un mecanismo de aprendizaje que permita modificar el repertorio de gestos.

### 1.4 Contribuciones de la Tesis

Como se ha comentado, el principal objetivo de este trabajo es la realización de un sistema de aprendizaje de gestos por imitación, basado en visión, e implementado para un robot social. La arquitectura propuesta para este sistema, y detallada más adelante, incorpora sensibles

diferencias respecto a otras contribuciones, y constituye una de las principales aportaciones de esta Tesis. La implementación de esta arquitectura involucra diversos elementos. El resto de contribuciones se localizan en algunos de estos elementos.

Así, para poder aprender gestos por imitación usando sólo información visual, era necesario desarrollar un sistema de Captura de Movimientos (Human Motion Capture (HMC)) que cumpliera las restricciones mencionadas para escenarios de Robot Learning by Imitation (RLbI). Este sistema supone una nueva aproximación al problema, cuyos resultados han sido cuantitativamente evaluados para probar su validez.

La percepción es sólo el primer paso en la imitación. Es necesario encontrar una representación de los gestos que permita construir una base de conocimientos compacta pero útil. En este texto se propone una nueva forma de representación basada en características tanto locales como globales.

No todos los gestos percibidos son nuevos. De hecho, transcurrido un cierto tiempo un robot social debería ser capaz de reconocer la mayor parte de los gestos que percibe. En esta Tesis se propone un método basado en el algoritmo Dynamic Time Warping (DTW) para comparar características locales. Los resultados de esta comparación se refuerzan con una comparación de características globales basada en algoritmos analíticos.

Las contribuciones de esta Tesis han cristalizado en diferentes publicaciones, que se listan en el apéndice G.

El trabajo presentado en este documento ha sido financiado por los Proyectos TIN2004-05961, TIN2005-01359 y TIN2008-06196, del Ministerio de España de Ciencia y Tecnología (MCYT) y fondos FEDER, los Proyectos P06-TIC-2123 y P07-TIC-03106, de la Junta de Andalucía, y el Proyecto VISOR de la European Robotics Research Network (EURON). La Tesis se ha desarrollado en el Grupo ISIS (*Ingeniería de Sistemas IntegradoS*), de la Universidad de Málaga, en España.

## 2 Arquitectura del sistema

La habilidad para aprender por imitación está presente en los humanos y en muchas otras especies animales. Dicha habilidad juega un papel muy importante en el desarrollo social de los individuos (Mosterín, 2005). Aunque en sus comienzos los sistemas de RLbI eran un escalón más en una serie de avances puramente tecnológicos, actualmente suelen estar inspirados, hasta

cierto punto, en las habilidades naturales previamente mencionadas. Así, los sistemas de RLbI pueden ser considerados, hoy día, el producto de una investigación interdisciplinar, que mezcla conceptos de neurología y ciencias sociales con la informática, la ingeniería industrial y las tecnologías de la comunicación y el tratamiento de señales. El capítulo 2 de la tesis ofrece una introducción al aprendizaje social biológico, con reseñas a la utilidad que estos mecanismos tienen en el aprendizaje en robots. En este resumen, nos limitaremos a indicar estas influencias cuando sea pertinente, mientras se describe la arquitectura implementada.

Los sistemas de RLbI pretenden proporcionar al robot mecanismos de aprendizaje que le permitan aprender de las demostraciones que hace un ser humano, sin necesidad de ser teleoperado por éste. Por tanto, se requerirá, ineludiblemente: (a) un aumento en las capacidades perceptuales del robot (por ejemplo, sistemas de Human Motion Capture (HMC) basados en visión) (Kojo et al., 2006; Hecht et al., 2009); y (b) una traducción interna de movimientos de persona, a movimientos de robot (Schaal, 1999; Shin et al., 2001). Esta traducción, llamada *retargeting*, permite al robot imitar, y aprender, los movimientos percibidos a pesar de sus diferencias físicas con la persona que los demuestra.

Un inconveniente importante relacionado con los sistemas de RLbI es el de la *generalización*. En robots industriales, que deben ejecutar sólo tareas predefinidas, este problema no es tan importante (Muench et al., 1994). Los robots sociales, sin embargo, deben adaptarse a las condiciones variables de los escenarios de RLbI. Para estos robots, la generalización de las tareas aprendidas es un requisito mucho más importante. Esta importancia ha hecho que en el campo de la robótica social la descripción de tareas a nivel de trayectorias (Ude et al., 2004; Calinon, 2007), como un flujo continuo de datos, se haya impuesto a la larga sobre la descripción a nivel *simbólico* (Alissandrakis et al., 2007), que representa a las tareas como sucesión de estados, y que resulta más abstracta y difícil de generalizar. Las descripciones basadas en trayectorias también son más fácilmente modificables como consecuencia de nuevas demostraciones, o de procesos de revisión interna de comportamientos (Calinon, 2007), cuya importancia en los sistemas de aprendizaje biológico ha sido destacada por autores como Bandura (1969).

En cuanto a los mecanismos para conseguir la generalización, actualmente se destacan dos. El primero de ellos es la supervisión (Muench et al., 1994). Gracias a la ayuda de una persona que conoce las características, objetivos y particularidades de la tarea es posible guiar el proceso de aprendizaje hacia una codificación mejor. El segundo mecanismo consiste en la utilización de algoritmos (algoritmos de Machine Learning (ML)) que permitan reconocer automáticamente patrones complejos y tomar decisiones basadas en los datos percibidos (Thrun and Mitchell, 1993). Mientras que estos últimos mecanismos permiten automatizar el proceso de

generalización, la supervisión normalmente ofrece mejores resultados, y es de hecho un elemento importante en los procesos de aprendizaje biológicos (Mosterín, 2005).

## 2.1 Componentes de un sistema de RLbI

Un sistema de RLbI para un robot social, por tanto, debe ser capaz de percibir las tareas realizadas por una persona sin exigir un elevado grado de supervisión (por ejemplo, sin requerir que el usuario teleopere al robot). Luego, debe representar esas tareas (o, usando un término más general, comportamientos) de una forma adecuada, siendo en principio las representaciones basadas en trayectorias las más adecuadas de cara a la generalización del aprendizaje. Las tareas aprendidas deben ser almacenadas, y comparadas con la tarea percibida. Finalmente, se hace necesario contar con mecanismos que permitan traducir las tareas percibidas, realizadas por una persona, al espacio de movimientos del robot.

La necesidad de satisfacer estos requisitos hace que en cualquier arquitectura de RLbI puedan distinguirse una serie de componentes principales, que se listan a continuación.

- **Entrada.** En este componente se incluyen todas las señales que aportan información al robot social, tanto sobre su entorno como sobre su propio estado interno.
- **Percepción.** Es necesario procesar la información suministrada por el componente de entrada, para extraer de ella sólo los datos que puedan ser útiles. Este procesado se realiza en el componente de percepción. Así, por ejemplo, en escenarios de RLbI el robot debería extraer, de las imágenes percibidas, sólo la pose de la persona que realiza la demostración, y la posición y orientación de los objetos utilizados.
- **Conocimiento.** Este componente contiene todos los elementos que el robot social utiliza para almacenar unidades de información. Estas unidades pueden ser preprogramadas (en cuyo caso se asemejan a los *genes* biológicos) o aprendidas (*memes*, en el campo de la biología (Dawkins, 1976)). También se incluyen en este componente los elementos encargados de codificar esta información, organizarla y transformarla a los formatos requeridos por otros componentes.
- **Aprendizaje.** Este componente proporciona al robot social la capacidad de aprender nuevas tareas, o modificar la información almacenada en el componente de conocimiento de acuerdo con nuevas observaciones o procesos de revisión interna.
- **Generación de movimientos.** El objetivo del aprendizaje por imitación es, en definitiva,

enseñar al robot social a *ejecutar* determinados comportamientos. Es necesario, así, que el robot sea capaz no sólo de percibir y aprender, sino también de *imitar* lo aprendido. El componente de generación de movimientos se encarga de realizar todos los procesos necesarios para generar el movimiento de los motores del robot.

- **Salida.** Los comandos de movimiento generados por el componente previo son recibidos por este componente, que usa las habilidades motoras del robot para ejecutarlos físicamente.

Existen diversos ejemplos de arquitecturas de RLbI, y en todas ellas es posible identificar estos componentes. Algunas de estas arquitecturas han inspirado de una manera u otra la propuesta de esta Tesis. Otras son contribuciones más actuales, cuya comparación con el sistema presentado aquí aporta valiosos resultados. El capítulo 2 detalla algunas de estas arquitecturas y su relación con la aportación presentada en esta Tesis. En concreto, se analizan las aportaciones de [Schaal \(1999\)](#), [Demiris and Hayes \(2002\)](#), [Breazeal et al. \(2004\)](#), [Mohammad and Nishida \(2009\)](#) y [Mühlig et al. \(2009\)](#).

## 2.2 Sistema de RLbI propuesto

La arquitectura del sistema de RLbI propuesto en esta Tesis (Fig. 2.8 del capítulo 2) presenta una serie de componentes fundamentales, detalladas a continuación.

- **Entrada.** En esta Tesis se presenta un sistema de RLbI que está basado exclusivamente en visión. Más concretamente, y tras evaluar otras alternativas, se optó por utilizar como fuente de información un par de cámaras estéreo. La separación entre las cámaras se asemeja a la separación entre los ojos de una persona, de forma que el robot esté físicamente preparado para percibir a personas que se encuentren a distancias típicas de interacción social (entre 1,5 y 2 metros).
- **Percepción.** El componente de percepción está orientado a extraer la pose de la persona que demuestra los gestos, a partir de la información suministrada por el par de cámaras estéreo. Así, en este componente se distinguen tres elementos enlazados secuencialmente. El primero de ellos extrae la silueta de la persona, y la posición de su cara y sus manos, de las imágenes de entrada. Estas posiciones se actualizan para cada fotograma percibido usando un algoritmo rápido de seguimiento implementado en el módulo de *tracking*. Finalmente, el último elemento de este componente implementa el algoritmo de HMC basado en modelo propiamente dicho que obtiene, a partir de la información anterior, la pose de la persona para cada fotograma.

- **Conocimiento y aprendizaje.** Inspirados en gran medida por las evidencias biológicas, que apuntan a la existencia de representaciones senso-motrices en el cerebro (por ejemplo, las investigaciones con *neuronas espejo* de Rizzolatti et al. (1996)), numerosos trabajos previos han buscado la unificación de la percepción y la acción en robots sociales. En el campo del aprendizaje de gestos, esta unificación se ha conseguido mediante la codificación de los gestos percibidos en el espacio de movimientos del robot. Por tanto, se traducen los gestos de la persona al robot *antes* de codificarlos en la base de conocimientos (Schaal, 1999; Demiris and Hayes, 2002; Lopes and Santos-Victor, 2005). En una implementación previa de la arquitectura propuesta en esta Tesis (Bandera et al., 2007), se siguió esta misma aproximación. Los resultados obtenidos fueron limitados. Las restricciones físicas del robot hacían que gestos distintos, que se podían distinguir sin problemas en el espacio de movimientos de la persona, se codificasen de manera muy parecida en el espacio de movimientos del robot. Ilustrando este caso con un ejemplo extremo, según esta estrategia un robot social que sólo tenga un brazo derecho no podrá distinguir gestos realizados por una persona con su brazo izquierdo. Y esta limitación no viene impuesta por el sistema perceptivo del robot, sino por sus capacidades motrices. Por tanto, mientras que la unificación de percepción y acción es una opción válida cuando el robot es físicamente muy parecido a la persona, resulta inadecuada cuando las diferencias entre el robot y la persona pueden ser considerables. Revisando estudios biológicos, se encuentran evidencias (Meltzoff and Moore, 1989) de que un niño encuentra problemas al intentar imitar los movimientos realizados por una máquina. Es razonable pensar que el caso opuesto, el de una máquina intentando imitar los gestos de una persona, también presente problemas.

En esta Tesis se propone, por tanto, seguir una estrategia diferente, que guarda cierta relación con el método Stanislavsky utilizado en el mundo del teatro (Stanislavsky, 1936). Esta estrategia puede resumirse en la frase “usar un modelo de persona para representar movimientos de personas”. Dicho de otro modo los gestos, caso particular de esta Tesis, se perciben, comprenden y aprenden sobre el modelo de persona usado en el componente de percepción. De esta manera las particulares capacidades motrices del robot no afectan a su percepción de los interlocutores humanos. El módulo de traducción o *retargeting* sólo se ejecuta cuando el robot imita los gestos aprendidos, lo que supone, como efecto adicional, un aumento en la eficiencia del sistema.

- **Generación de movimientos y salida.** Los gestos que el robot va a imitar no se traducen directamente en comandos para sus motores. En lugar de ello, el movimiento traducido es enviado antes a un modelo virtual del robot. Dicho modelo se basa en los mismos algoritmos que se usan para animar el modelo de persona usado previamente, y se

encarga de validar las poses que serán finalmente enviadas a los motores.

### 3 Percepción de los movimientos de la persona

En esta Tesis se propone el uso de un sistema de HMC, basado en modelo, para inferir la pose de la parte superior del cuerpo de una persona, situada delante de un robot social equipado con un sistema de visión estéreo. Dicho sistema satisface las condiciones previamente descritas para los escenarios de RLbI, con lo que no se impone a la persona que se coloque marcadores, ni vista prendas especiales. Los datos de entrada para el sistema de HMC propuesto se limitan a las imágenes de color y disparidad proporcionadas por el par estéreo.

El primer paso ejecutado por el sistema de HMC propuesto es buscar una persona que tenga intención de interactuar con el robot. Para localizar dicha persona, se busca en las imágenes de color proporcionadas por el par estéreo una cara que esté mirando al robot, a una distancia no excesivamente grande. Para encontrar dicha cara se utiliza el detector propuesto por [Viola and Jones \(2001\)](#).

#### 3.1 Extracción de la silueta

Una vez se detecta una cara, se calcula su posición 3D usando información acerca de su disparidad media. El siguiente paso es extraer la silueta de la persona de las imágenes estéreo de entrada. Para ello, se utiliza un algoritmo que combina los siguientes dos procesos:

- Una segmentación por profundidad, realizada en el mapa de disparidad. Este proceso, detallado en el capítulo 3 de esta Tesis, primero selecciona de entre todas las zonas de la imagen sólo aquellas cuya distancia a la posición 3D de la cara está por debajo de ciertos umbrales. Luego, se descartan todas las regiones que, aunque cumplen este requisito, están desconectadas de la cara.
- Una detección de bordes realizada en el mapa de disparidad usando el algoritmo de [Canny \(1983\)](#). Como se detalla en el capítulo 3, este algoritmo se ejecuta sólo en la vecindad de la silueta extraída en el proceso anterior.

Los resultados de estos dos procesos se combinan mediante una OR lógica. Al resultado de esta combinación se lo somete a un proceso de *closing* (o “rellenado”), que combina erosión y

dilatación de la silueta para reducir huecos que pueden aparecer erróneamente, debido al ruido presente en el mapa de disparidad.

Una vez concluido este paso, se buscan dentro de la silueta la cara y las manos de la persona. Para ello, se buscan las tres regiones de color piel más grandes localizadas dentro de la silueta. La cara se asocia a la región de color piel más cercana a la cara detectada en imágenes anteriores. Tras etiquetar la región de la cara, las manos izquierda y derecha se asocian, respectivamente, a las zonas de color piel detectadas a la izquierda y a la derecha de la cara. Como se ve, sólo se impone que al principio de la interacción la persona no esté con las manos cruzadas. A partir de ese momento, se realizará el seguimiento o *tracking* de cara y manos como regiones de color piel. El algoritmo de tracking empleado en esta Tesis se detalla en el apéndice D.

### 3.2 Estimación de la pose global del torso

A partir de la silueta previamente obtenida, se calculan en primer lugar la inclinación y rotación del torso. Para ello, se utilizan tablas antropométricas extraídas de los estudios de [Contini \(1972\)](#). Dichas tablas contienen las proporciones medias de una persona, referidas a su altura. Estos datos son, por supuesto, generales, pero pueden aplicarse para cálculos aproximados referidos a personas adultas de cualquier sexo.

Aparte de estas tablas, en la obtención de la pose se utiliza un modelo virtual de la persona, que representa la parte superior del torso de un humano. En la estimación de la pose del torso, este modelo se utiliza simplemente para comprobar que las inclinaciones y rotaciones estimadas caen dentro de los límites humanos. Como se verá en el siguiente apartado, el papel de este modelo virtual es más importante a la hora de generar una pose para los brazos.

Para estimar la pose del torso, se utiliza un algoritmo analítico que ajusta la silueta percibida a la correspondiente pose mediante una serie de pasos ampliamente detallados en el capítulo 3 de esta Tesis. En primer lugar se calcula el eje medio de la silueta; a continuación, según dicho eje medio, se calcula la inclinación, tanto lateral como hacia adelante o hacia atrás, del torso. Posteriormente, usando de nuevo el eje medio como referencia, se obtienen las regiones de la silueta donde probablemente se sitúan los hombros, según las tablas antropométricas. Finalmente, la información de disparidad de los hombros se usa para calcular el grado de rotación del torso.

### 3.3 Generación de la pose de los brazos

Una vez que la pose del torso ha sido calculada, se pasa a generar la pose de los brazos, según las posiciones de las manos. Para ello, se utiliza un algoritmo de Cinemática Inversa (Inverse Kinematics (IK)) basado en modelo. Antes de ejecutar este algoritmo, el torso en el modelo se ajusta y gira según los valores previamente estimados. Luego, se toman las posiciones 3D de las manos, obtenidas a partir de sus posiciones en las imágenes de color y sus datos de disparidad asociados, y se filtran usando un filtro FIR detallado en el apéndice E. Dicho filtrado reduce los efectos del ruido de disparidad y otros errores.

Lo siguiente es utilizar la posición de las manos para inferir una pose del brazo. Ésto se hace mediante el uso de un algoritmo de Inverse Kinematics (IK) detallado en el apéndice F. La naturaleza analítica de dicho algoritmo le permite ofrecer sus resultados con rapidez, lo que lo hace más indicado para escenarios de RLBI que otros algoritmos probabilísticos (Moeslund et al., 2006). Sin embargo, la pose ofrecida como resultado no tiene por qué satisfacer los límites impuestos por las articulaciones humanas y las colisiones entre diferentes segmentos corporales.

Por tanto, tras la ejecución del algoritmo de IK se evalúa la validez de la pose obtenida para el brazo, utilizando datos antropométricos para comprobar la adecuación a los límites de las articulaciones, y un algoritmo de detección rápida de intersecciones entre mallas de triángulos (Gottschalk et al., 1996) para comprobar la existencia de colisiones. Si la pose obtenida para el brazo es válida según ambos criterios, se adopta como pose final. Si no lo es, entonces es necesario ejecutar un algoritmo de búsqueda de posiciones alternativas para el brazo.

El capítulo 3 de esta Tesis detalla el algoritmo de búsqueda de poses alternativas que se ha implementado. Dicho algoritmo, dada una pose incorrecta, busca configuraciones válidas del brazo que varíen la posición del codo, pero sitúen la mano (el efector final) en la posición deseada. Si no se encuentra alternativa según esta búsqueda, entonces el algoritmo busca una pose lo más parecida posible a la deseada, modificando sólo ciertas articulaciones.

Finalmente, cabe mencionar que el algoritmo de búsqueda de alternativas también se ejecuta cuando la pose es válida, simplemente para buscar alternativas que sitúen el codo a una altura menor. Esto da como resultado poses normalmente más naturales, y en cualquier caso más eficientes desde un punto de vista energético.

### 3.4 Evaluación del sistema de HMC

El sistema de HMC propuesto en esta Tesis se evaluó durante su desarrollo de forma cualitativa, pero se precisaba una evaluación más precisa de sus posibilidades de cara a validarlo como un sistema útil en los escenarios de RLbI considerados. En resumen, es necesario suministrar medidas cuantitativas de error.

Para realizar una evaluación cuantitativa del sistema de HMC propuesto, se han capturado los movimientos realizados por una persona con dos sistemas distintos de HMC. El primero de ellos es el sistema propuesto, basado en visión estéreo. El segundo es un sistema que sirve de referencia. Dicho de otro modo, los datos capturados por este segundo sistema se consideran fidedignos, y contra ellos se contrastan los datos proporcionados por el sistema evaluado. En esta Tesis, como sistema de referencia se ha utilizado un Codamotion CX1 basado en marcadores activos. Este sistema usa marcadores infrarrojos colocados en diferentes partes del cuerpo. La resolución de los datos depende de la visibilidad y de otros factores, pero se garantiza un error de posición por debajo de 1.5 mm. a distancias de 2 metros, en condiciones de incidencia normal y visibilidad buena.

El capítulo 3 de la Tesis detalla el escenario montado para realizar las pruebas, la posición de los marcadores y los mecanismos que se utilizaron para alinear las capturas y filtrar los errores puntuales que presentó el sistema Codamotion. La tabla 1 proporciona las medidas cuantitativas de error obtenidas tras realizar todas las pruebas. Puede comprobarse cómo los codos, cuya posición no es obtenida ni estimada a partir de las imágenes percibidas, sino determinada por el método, acumulan errores mayores. Es también interesante mencionar que un estudio más detallado de estos errores reveló que, a pesar de utilizar en todo caso un sistema estéreo calibrado, los errores crecían al acercarse los objetos a los bordes de la imagen. Es por ello que los errores de las manos, que se acercan más a dichos bordes, son mayores que los de los hombros, la cabeza o el abdomen. Como se ha comentado antes, en cualquier caso los errores cometidos se mantienen en unos límites aceptables, válidos para los escenarios de RLbI considerados en esta Tesis.

## 4 Representación, reconocimiento y aprendizaje de gestos

Para aprender por imitación, un robot social no sólo debe percibir a la persona que realiza la demostración. También debe ser capaz de extraer las partes relevantes de dicha demostración y descartar lo que no es importante, representar las tareas (en esta Tesis, gestos) observadas de una forma eficiente y comparar dicha representación con aquellas que pueda tener memorizadas

Table 1: Promedio de los errores de posición, obtenido a partir de más de 5300 imágenes.

<b>Marcador</b>	Hombro izdo.	Codo izdo.	Mano izda.
<b>Error medio (cm)</b>	5.74	12.53	11.51
<b>Desviación estándar (cm)</b>	3.13	6.06	6.55
<b>Marcador</b>	Hombro dcho.	Codo dcho.	Mano dcha.
<b>Error medio (cm)</b>	6.72	12.41	11.47
<b>Desviación estándar (cm)</b>	5.01	6.94	7.63
<b>Marcador</b>	Cabeza izda.	Abdomen	Cabeza dcha.
<b>Error medio (cm)</b>	7.03	7.76	6.51
<b>Desviación estándar (cm)</b>	5.41	1.18	5.13

para dilucidar si la tarea percibida es nueva o conocida. Finalmente, el robot debe disponer de algún mecanismo que le permita actualizar su base de conocimientos de acuerdo con las demostraciones percibidas. En esta sección se describen brevemente los métodos de segmentación, representación, reconocimiento y aprendizaje desarrollados en esta Tesis para implementar estas funcionalidades.

#### 4.1 Segmentación

El robot social percibe una secuencia continua de movimientos, que debe ser en primer lugar segmentada en un conjunto discreto de gestos. En esta Tesis, se considera que los puntos de inicio y fin de gesto se marcan con una pausa en la demostración. El algoritmo encargado de detectar estas pausas actúa de forma distinta según se esté buscando el inicio o el fin del gesto, con el objetivo de poder detectar el inicio del movimiento con una precisión alta, al tiempo que se reducen las falsas detecciones de fin de gesto debidas a ruido o a pequeñas pausas en la demostración. El empleo de estas pausas para segmentar los gestos permitirá en el futuro incorporar al sistema la detección de gestos estáticos, que se corresponderán precisamente con las poses de la persona durante estas pausas.

#### 4.2 Representación

Como se ha visto en secciones anteriores, la percepción en el sistema propuesto proporciona la pose de la persona percibida para cada fotograma de la secuencia de entrada. En esta Tesis, cada una de las poses de esta secuencia se representa como un conjunto de trayectorias 3D seguidas por ciertas partes del cuerpo. La necesidad de respuesta rápida por parte del robot obliga a usar una tasa de imágenes por segundo tan elevada como sea posible. Ésto hace que las secuencias

percibidas tengan una dimensión considerable, al menos en aquellos gestos que requieran cierto tiempo para ser ejecutados. Para poder procesar dichos gestos de forma rápida, es necesario reducir su dimensionalidad.

Técnicas como el Análisis de Componentes Principales (Principal Component Analysis (PCA)) (Jolliffe, 1986) o la Regresión por Proyección Ponderada Localmente (Locally Weighted Projection Regression (LWPR)) (Vijayakumar et al., 2005) son aproximaciones genéricas al problema de la reducción de la dimensionalidad. En esta Tesis, sin embargo, se quiere aplicar dicha reducción a un conjunto de datos muy concreto: las trayectorias 3D seguidas por ciertas partes del cuerpo (cara y manos). Como quiera que son un caso muy concreto de trayectoria, podrían aprovecharse algunas de sus características para encontrar una representación particular que mejore las anteriores, más genéricas.

Diferentes autores (Croitoru et al., 2005; Asfour et al., 2006a; Alajlan et al., 2007) proponen la extracción de ciertas características de las trayectorias, que se pueden utilizar para codificarlas de forma simplificada. Tal y como indica Alajlan et al. (2007), se pueden emplear tanto características globales como locales. Las globales son menos sensibles al ruido pero puede ser complicado representar con ellas los detalles de la trayectoria. Las características locales no tienen este problema. Sin embargo, dependen mucho de la calidad de la segmentación, requieren un ajuste muy fino de los parámetros y son más sensibles al ruido (Calinon, 2007).

En esta Tesis se propone un nuevo sistema de representación que utiliza tanto características locales como globales para representar gestos, percibidos como secuencias de trayectorias 3D.

- Las características *locales* se obtienen como las secuencias de puntos dominantes de las trayectorias 3D percibidas. La extracción de puntos dominantes comienza con el cálculo de la función de curvatura para cada trayectoria. De acuerdo con Calinon (2007), el uso de métodos estándar de detección de curvatura ha causado problemas de segmentación en escenarios de RLbI. En esta Tesis se propone utilizar un nuevo método de detección adaptativa de curvatura en trayectorias 3D. Esta técnica adaptativa permite recoger las pequeñas variaciones de la trayectoria al tiempo que se mantiene una eficacia aceptable en el filtrado del ruido (Bandera et al., 2000). En el capítulo 4 puede encontrarse una descripción detallada del algoritmo implementado.

Una vez obtenidas las curvaturas 3D, es necesario distinguir en ellas una serie de puntos dominantes. En una primera iteración, puntos dominantes serán los máximos locales de curvatura y los puntos de inicio y fin de la trayectoria. Una vez marcados estos puntos,

una segunda iteración comprueba el tiempo transcurrido entre puntos dominantes. Cuando este tiempo supera cierto umbral, se inserta un nuevo punto dominante. De esta forma se garantiza que se va a disponer de una mínima cantidad de puntos dominantes incluso en aquellos casos en los que el movimiento es suave y no presenta picos abruptos de curvatura. La validez de estas secuencias de puntos dominantes como representación local de las trayectorias ha sido evaluada comparándolas con las secuencias extraídas mediante otros métodos de curvatura fija, así como con el Curvature Scale Space (CSS) (Mokhtarian and Mackworth, 1986), considerado un descriptor estándar. Como secuencias de prueba, se usaron gestos capturados usando el sistema Codamotion CX1 basado en marcadores activos descrito en la sección anterior. Los resultados, detallados en el capítulo 4, muestran que el método propuesto proporciona resultados similares a los anteriores en términos de tasas de compresión, pero es considerablemente más rápido, y codifica las trayectorias con errores sensiblemente menores.

- El segundo tipo de características utilizadas para representar los gestos percibidos son características *globales*. En esta Tesis, se utilizan dos características globales, relacionadas ambas con las amplitudes de las trayectorias percibidas. La primera de ellas se obtiene como la diferencia entre los valores máximos y mínimos alcanzados en cada coordenada X, Y, Z para cada trayectoria. La segunda característica global almacena el movimiento relativo de unas trayectorias percibidas respecto a otras. Así, por ejemplo, dado un cierto movimiento de la mano derecha en un gesto determinado, la primera característica global permite dilucidar si dicho movimiento fue amplio o no, mientras que la segunda indica si dicho movimiento fue más o menos amplio que el movimiento de otras partes del cuerpo, como la mano izquierda.

El cálculo de estas características globales se detalla en el capítulo 4. Su simplicidad hace que se calculen en un tiempo muy reducido, despreciable en las escalas que se consideran en el sistema de RLbI propuesto.

### 4.3 Comparación y reconocimiento

Los gestos percibidos se han representado mediante un conjunto de características locales y globales. Del mismo modo estarán codificados aquellos gestos almacenados en el repertorio del robot social. Es necesario ahora para dicho robot contar con un mecanismo que le permita comparar el gesto percibido con los gestos conocidos. De esta forma, el robot puede reconocer un gesto previamente demostrado. También puede utilizarse el resultado de esta comparación

para modificar el repertorio de gestos conocidos añadiendo nuevos gestos, o cambiando las representaciones internas de gestos conocidos con los datos obtenidos de las nuevas demostraciones del gesto.

En esta Tesis, se propone que la comparación de gestos se realice en dos etapas: en la primera de ellas, se realiza una comparación de las características locales (secuencias de puntos dominantes). El resultado de dicha comparación local se refuerza, en una segunda etapa, con los resultados obtenidos al comparar las características globales de los gestos.

- Existen diferentes métodos para realizar comparaciones de trayectorias 3D (en este caso, secuencias de puntos dominantes). En el caso de los gestos, se deben tener en cuenta ciertas características representativas como son las posibles diferencias en tasas de muestreo y en temporizaciones relativas de un gesto respecto a otro, la presencia de ruidos o la posibilidad de tener que comparar secuencias de diferente longitud (Croitoru et al., 2005). Actualmente, los Modelos Ocultos de Markov (Hidden Markov Model (HMM)) pueden considerarse el estado del arte en lo que se refiere a reconocimiento de gestos (Kojo et al., 2006; Asfour et al., 2006a; Aleotti and Caselli, 2006). Sin embargo las HMMs presentan, entre otros, dos inconvenientes importantes: (a) el número de estados que se utilizan en el modelo está limitado por la complejidad de los algoritmos de entrenamiento e inferencia; y (b) se requieren una gran cantidad de datos de entrenamiento. En los escenarios de RLbI considerados, la necesidad de una respuesta rápida va a forzar el uso de modelos necesariamente simples e incompletos. Además, en dichos escenarios no tiene por qué disponerse de la cantidad de datos de entrenamiento requeridos por estos algoritmos.

Otras aproximaciones al problema de la comparación de gestos, representados como trayectorias, consideran el uso de técnicas de programación dinámica (Croitoru et al., 2005; Chen et al., 2005). Estos algoritmos se basan en la comparación del gesto de entrada con un repertorio de gestos conocidos y clasificados. Hay diferentes maneras de calcular las distancias entre puntos de las trayectorias, lo que origina distintos métodos. Los tiempos de respuesta, así como la robustez y validez de los resultados de estos métodos, varían según el tipo de trayectorias que se estén comparando. Sin embargo, dichos algoritmos de programación dinámica no requieren una simplificación adicional de los datos percibidos, al no estar limitados por un número restringido de estados. Además, requieren menos entrenamiento que las HMMs. En esta Tesis se optó por utilizar dichos algoritmos para realizar la comparación de las características locales de los gestos. Aparte de la comparación directa basada en distancias euclídeas, se evaluaron diferentes algoritmos de programación dinámica para calcular dichas distancias entre secuencias de puntos dominantes. En con-

creto, el Dynamic Time Warping (DTW), el Edit Distance on Real Sequence (EDR), el Edit Distance with Real Penalty (ERP) y el Longest Common Subsequences (LCSS) (Chen et al., 2005; Croitoru et al., 2005). Los detalles acerca de cómo se han implementado estos algoritmos en esta Tesis se encuentran en el capítulo 4. Para evaluarlos, se utilizó una base de datos de once gestos que fueron ejecutados varias veces por seis personas diferentes. Los movimientos de estas personas se registraron utilizando el sistema Codamotion CX1 descrito anteriormente. Estas pruebas representan un escenario controlado, donde las secuencias de entrada pueden considerarse como libres de ruido (una vez filtradas las muestras incorrectas puntuales) y, por tanto, es más fácil establecer una correspondencia entre los resultados obtenidos y la validez del método utilizado para obtenerlos. El proceso de evaluación, detallado en el capítulo 4, concluye que tanto el DTW como el Edit Distance with Real Penalty (ERP) son, en principio, válidos para comparar las características locales de los gestos. Hay una serie de características comunes a estos dos métodos que hacía, en cierto modo, previsible este resultado: ninguno de los algoritmos está basado en el empleo de un umbral fijo, y ambos permiten un acomodamiento elástico de las trayectorias comparadas, con lo que son menos sensibles a los efectos de las diferencias relativas de temporización.

- En esta Tesis las características globales no se pueden utilizar realmente para discriminar entre un gesto u otro. En cambio, son menos sensibles que las locales al ruido y a los errores puntuales, y se utilizan para reforzar las comparaciones basadas en características locales. Los detalles acerca de los algoritmos analíticos empleados, en esta Tesis, para realizar esta comparación global se dan en el capítulo 4. Como allí se demuestra, el efecto del refuerzo global mejora sensiblemente los resultados de la comparación local. A cambio, el aumento en el tiempo requerido para realizar la comparación es despreciable.

#### 4.4 Aprendizaje

La comparación de un gesto percibido con aquellos almacenados en el repertorio del robot social origina un cierto valor de parecido. Dicho “parecido”, siguiendo la nomenclatura de Demiris and Hayes (2002), se denomina *valor de confianza*. El aprendizaje de gestos utiliza estos valores de confianza, obtenidos en el proceso de reconocimiento, para modificar o no el repertorio de gestos aprendidos.

Las secuencias de puntos dominantes usados para representar los gestos pueden resultar dispersas y ruidosas. Es, por tanto, necesario utilizar un criterio de decisión robusto. En esta

Tesis se propone un algoritmo basado en dos umbrales: uno absoluto y otro relativo. El umbral absoluto selecciona todos aquellos gestos en el repertorio que se parecen al percibido. El umbral relativo, por otro lado, compara los dos valores de confianza más altos obtenidos en la comparación del gesto percibido con los conocidos. Este segundo umbral sólo será superado por aquellos gestos que sean suficientemente parecidos a uno de los almacenados, pero suficientemente distintos del resto. La inclusión de nuevos gestos en la base de datos, en cualquier caso, se realizará siempre contando con la supervisión de una persona.

Tras realizar algunos experimentos, se comprobó que los momentos iniciales del aprendizaje, en los que el robot social aún no cuenta con un repertorio amplio de gestos, son críticos. Para adaptarse a esta situación el algoritmo de aprendizaje, como se detalla en el capítulo 4, aumenta el grado de supervisión humana en las primeras etapas del proceso.

## 5 Traducción del movimiento

En el contexto de la imitación y el aprendizaje social, el *problema de la correspondencia* se puede definir como el problema de encontrar una traducción de los movimientos realizados por el demostrador al cuerpo del observador (Nehaniv and Dautenhahn, 2002). A esta traducción se la denomina usualmente *retargeting*. Si los cuerpos del demostrador y el observador son muy parecidos, este *retargeting* se puede implementar de forma obvia con una correspondencia directa. Pero si estos cuerpos son distintos, entonces el problema requiere soluciones más complejas. En los escenarios de RLbI que consideramos en esta Tesis, en los que un robot social intenta imitar y aprender observando los movimientos de una persona, se da esta última circunstancia. Los cuerpos de la persona y el robot son muy distintos. Es necesario, por tanto, implementar un algoritmo de *retargeting* que pueda adaptarse adecuadamente a estas diferencias.

En esta Tesis, se ha optado por seguir las propuestas de autores como Shin et al. (2001), e implementar un *retargeting* combinado, en donde la traducción que finalmente se aplica es una mezcla ponderada de los resultados obtenidos por diferentes estrategias. A la hora de elegir dichas estrategias, es interesante considerar las investigaciones de Smyth and Pendleton (1990) en el campo de la biología. (Smyth and Pendleton, 1990) diferencian movimientos *localizados* y *configurados*. En los primeros lo que realmente importa es la posición del efector final, o de otra parte igualmente relevante del cuerpo. En los segundos, lo que importa es preservar el movimiento relativo y la configuración de cada parte del cuerpo. En esta Tesis se implementa un sistema de reconocimiento de gestos sociales individuales, que no involucran objetos ni a otros individuos. Por ello, las estrategias que se mezclan en el algoritmo de *retargeting* combinado

propuesto son dos, atendiendo a la división de [Smyth and Pendleton \(1990\)](#).

- **Retargeting de posiciones.** Esta estrategia trata de que las posiciones de las manos de la persona, referidas a su cintura y normalizadas según la longitud del brazo de la persona, se traduzcan fielmente a posiciones de los efectores finales del robot, referidas a su cintura y normalizadas según la longitud de su brazo.
- **Retargeting de ángulos.** En este caso, lo que se trata de preservar en la traducción es el conjunto de ángulos de las articulaciones del brazo. Más concretamente, se enfatiza la conservación del movimiento de estas articulaciones, aún cuando estos movimientos produzcan diferentes posiciones en los efectores finales debido a las importantes diferencias existentes entre la persona y el robot.

La mezcla de estas estrategias se basa en emplear un factor cuyo valor depende de la amplitud de los movimientos percibidos. Cuando los movimientos son amplios, predomina el *retargeting* de ángulos. En movimientos pequeños o gestos estáticos, predomina el *retargeting* de posiciones. El capítulo 5 da una descripción detallada de estas dos estrategias, así como de la forma en que se mezclan de forma ponderada para generar la pose final del robot. Una ventaja del algoritmo propuesto es que nuevas estrategias pueden añadirse fácilmente a la mezcla ponderada, con lo que es posible extender dicho algoritmo a escenarios en los que haya más de un demostrador, intervengan objetos, etc. Otra ventaja es que se trata de un algoritmo analítico, que además no depende de la cantidad de datos que se estén usando para percibir a la persona, y que se ejecuta de manera independiente para cada brazo.

## 5.1 Evaluación del sistema de traducción

Para la evaluación del sistema de *retargeting* se compara la pose adoptada por el modelo de persona con la pose traducida al modelo de robot. La comparación de las poses se realiza con dos medidas distintas: (a) la variación entre la posición de la mano de la persona, y la posición del efector final del robot. Dichas posiciones están normalizadas según la altura del robot; y (b) la variación en los ángulos de las articulaciones. Se consideran tres grados de libertad en el hombro y uno en el codo.

Los resultados de las pruebas realizadas, detallados en el capítulo 5 de esta Tesis, muestran cómo la estrategia de *retargeting* combinado se acerca al *retargeting* de posiciones para gestos estáticos, y al de ángulos para gestos dinámicos, adaptándose por tanto a la estrategia más adecuada en cada caso.

## 6 Resultados de las pruebas sobre el sistema completo de aprendizaje por imitación

En esta sección se describen los resultados obtenidos en las pruebas finales del sistema de RLbI completo. Dichas pruebas han implicado la realización de un conjunto de gestos por parte de seis personas diferentes. Estos gestos involucran a la parte superior del cuerpo, y fueron descritos a las personas que los realizaban con una frase corta, acompañada de una única demostración.

Las demostraciones se realizaron a una distancia variable de las cámaras estéreo, de 1.30 a 1.80 metros. Para satisfacer las condiciones previamente descritas para escenarios de RLbI, las cámaras se situaron en diferentes lugares de los laboratorios donde trabajan los miembros del Grupo ISIS.

La primera batería de pruebas consideró que el robot ya disponía de un repertorio de gestos conocidos. De esta forma, se pudo evaluar la validez de los sistemas de captura de movimientos, representación y reconocimiento de gestos. Para construir dicho repertorio se almacenaron tres demostraciones de cada gesto. Los resultados de dichas pruebas demostraron que el algoritmo propuesto era capaz de clasificar correctamente el 100 % de los gestos, cuando el criterio de éxito utilizado era el  $1 - NN$  (vecino más cercano). El uso de un criterio más restrictivo  $3 - NN$  arroja un porcentaje de éxito del 79%, que puede considerarse un resultado adecuado a las necesidades del sistema propuesto, especialmente si tenemos en cuenta que el repertorio aprendido sólo incluye tres demostraciones de cada gesto. De hecho, las tasas de reconocimiento obtenidas son parecidas a las obtenidas en otros trabajos previos, donde las condiciones estaban más controladas (Kojo et al., 2006; Calinon, 2007).

El segundo conjunto de pruebas realizadas sobre el sistema de RLbI propuesto representa la verificación de toda la arquitectura propuesta. Así, partiendo de una situación en la que el robot social no tiene conocimiento previo alguno, se realizan demostraciones de movimientos que el robot tiene que dividir en gestos, representar, almacenar, reconocer e imitar. Como actualmente sólo se dispone de un brazo para el robot social en el Grupo ISIS, la imitación física se limita a dicho brazo, aunque el modelo virtual es el del robot completo.

Gracias en gran medida al aumento en la supervisión de las etapas iniciales del aprendizaje, previamente comentado, el sistema de RLbI propuesto fue capaz de ir incrementando adecuadamente la base de datos de gestos, y de reconocer todos los gestos ejecutados que ya se habían incluido en el repertorio. Las principales limitaciones del sistema no provienen de los módulos de reconocimiento o aprendizaje, sino de la percepción. En términos generales,

puede afirmarse que, en el sistema de RLbI propuesto, si la percepción es buena el gesto será normalmente representado, reconocido y, si es el caso, aprendido de forma adecuada.

## 7 Trabajo futuro

Para finalizar este resumen, se enumeran las principales líneas de trabajo futuro que emanan de esta Tesis. Dichas líneas se detallan en el capítulo 7, donde también puede encontrarse una discusión sobre algunos aspectos generales de la Tesis.

- Incremento de las capacidades perceptuales. Este incremento se refiere tanto a la mejora del sistema basado en visión, como a la inclusión de nuevos sistemas perceptivos, tales como el auditivo, el táctil, o sistemas basados en sensores láser.
- Consideración de más características globales, tales como las propuestas por [Cooper and Bowden \(2007\)](#).
- Uso de un módulo de aprendizaje más versátil.
- Realización de pruebas sobre un robot social real completo.
- Integración del sistema de RLbI en una arquitectura de alto nivel. Para un robot social, es importante poder decidir *cuándo, qué y cómo* imitar ([Schaal, 1999](#)). En esta Tesis se ha respondido parcialmente a las dos últimas preguntas. La respuesta a la primera es responsabilidad de capas superiores de decisión, que pueden también modificar las respuestas a las otras dos preguntas.



Departamento de Tecnología Electrónica  
E. T. S. I. Telecomunicación  
Universidad de Málaga

CONCLUSIONES DE LA TESIS DOCTORAL

VISION-BASED GESTURE RECOGNITION IN A  
ROBOT LEARNING BY IMITATION  
FRAMEWORK

AUTOR: Juan Pedro Bandera Rubio  
Ingeniero de Telecomunicación

DIRECTORES:

Juan Antonio Rodríguez Fernández  
Ingeniero de Telecomunicación  
Dr. por la Universidad de Málaga

Luis Molina-Tanco  
Ingeniero de Telecomunicación  
Dr. por la Universidad de Surrey



# Conclusiones

Esta Tesis propone un nuevo sistema de aprendizaje por imitación para robots sociales. Dicho sistema está basado exclusivamente en visión y tiene como objetivo fundamental el satisfacer las condiciones encontradas en procesos de interacción en entornos reales, una situación denominada en esta Tesis “escenario de RLbI”. De forma más precisa, el sistema propuesto se encarga de la percepción, representación, reconocimiento y aprendizaje de gestos realizados con la parte superior del cuerpo por una persona que interactúa con el robot. Para implementar esta funcionalidad, ha sido necesario dar una serie de pasos que pueden resumirse de esta manera: (a) percibir a la persona; (b) capturar sus movimientos; (c) segmentar los movimientos en gestos discretos; (d) codificar dichos gestos de forma eficiente; (e) reconocer el gesto si ya ha sido aprendido por el robot; (f) aprender, o modificar el repertorio de gestos del robot, de acuerdo con los resultados del reconocimiento; y (g) si se requiere, imitar físicamente los gestos aprendidos o percibidos.

El capítulo 1 de la Tesis describe su motivación, define el concepto de robot social y enumera las características propias de los escenarios de RLbI. También se presenta la propuesta de esta Tesis, así como sus principales contribuciones.

En el capítulo 2, se detalla la arquitectura del sistema de RLbI propuesto. Esta arquitectura se inspira en conceptos derivados de las teorías de aprendizaje social biológico, así como en otras arquitecturas anteriores de RLbI. La propuesta de esta Tesis tiene, sin embargo, ciertas particularidades e incorpora una serie de mejoras, novedades y alternativas respecto a trabajos previos y contemporáneos en este campo. De estas contribuciones, una de las principales es la implementación del conocimiento en el espacio de movimientos de la persona, en lugar de en el espacio del robot. Por tanto, un modelo interno de persona sustituye al robot imitador en el proceso de aprendizaje por imitación. Ésto puede describirse como “utilizar un modelo humano para representar movimientos humanos”, y supone una importante diferencia respecto a la mayoría de contribuciones previas. Este cambio permite no sólo incrementar la capacidad discriminativa del sistema de reconocimiento de gestos, sino también mejorar la eficiencia del

sistema ya que el módulo encargado de traducir el movimiento al espacio del robot sólo se ejecuta cuando se requiere que el robot imite físicamente lo percibido.

Los requisitos de los escenarios de RLbI hacen que la mayor parte de robots sociales confíen en la visión como principal entrada sensorial. El capítulo 3 detalla el sistema de percepción de movimientos basado en visión desarrollado en esta Tesis. Este sistema extrae el movimiento de la persona de las imágenes captadas por un par de cámaras estéreo montadas en la cabeza del robot. A continuación, se estima la pose de la parte superior del cuerpo de la persona utilizando, para ello, un nuevo método de estimación basado en tablas antropométricas. Una vez situado el torso, se calcula la posición de los brazos utilizando un algoritmo de Cinemática Inversa modificado para que pueda encontrar alternativas a poses erróneas. Finalmente, la última parte de este capítulo proporciona una extensa valoración cuantitativa del sistema de captura de movimientos, marcando sus ventajas e inconvenientes y validándolo como parte de una arquitectura de RLbI.

El capítulo 4 presenta los módulos de representación, reconocimiento y aprendizaje de gestos que han sido implementados en esta Tesis. Antes de esto, se detalla el método utilizado para segmentar un movimiento en gestos. Una vez el movimiento percibido ha sido dividido en un conjunto discreto de gestos, dichos gestos se representan usando un nuevo método basado en características locales y globales. Las características locales son, básicamente, secuencias de puntos dominantes extraídos de funciones de curvatura 3D adaptativa, que a su vez han sido obtenidas a partir de las trayectorias percibidas para diferentes partes relevantes del cuerpo. Esta representación, ligeramente menos eficiente que las que se pueden obtener usando Principal Component Analysis (PCA), proporciona sin embargo mejores resultados en la fase de reconocimiento, como se demuestra en el capítulo 4. También presenta importantes ventajas respecto a otros descriptores tales como el Curvature Scale Space (CSS) o las curvaturas fijas. El segundo conjunto de características usadas para representar el gesto son las globales. Dichas características incluyen una serie de medidas simples sobre las amplitudes absolutas y relativas del movimiento percibido.

Una vez que un cierto gesto ha sido codificado, es necesario, para reconocerlo, compararlo con el repertorio de gestos almacenados por el robot. La necesidad de conseguir implementar esta funcionalidad sin utilizar fases intensas de entrenamiento previno contra el uso de soluciones probabilísticas basadas en HMMs. En lugar de ello, diferentes técnicas de cálculo de distancias, basadas en algoritmos de programación dinámica, han sido evaluadas en esta Tesis para comparar secuencias de puntos dominantes. Los experimentos destacan el DTW sobre las demás. El reconocimiento de gestos, sin embargo, no sólo compara las secuencias de puntos

dominantes (características locales), sino también las características globales. En esta Tesis se propone un sencillo método analítico para comparar estas características. El resultado final de la comparación, expresado como un cierto valor de confianza, es una combinación de las distancias locales y globales. Los valores de confianza obtenidos cuando un gesto percibido se compara con los gestos en el repertorio del robot son las entradas del módulo de aprendizaje propuesto en esta Tesis. Los gestos no reconocidos pueden representar nuevos gestos que deben ser incluidos en el repertorio. Sin embargo, también pueden representar gestos incorrectos, o movimientos ruidosos. Por ello, el sistema de aprendizaje solicita cierto grado de supervisión por parte de la persona, para solventar estas situaciones confusas, detectadas mediante el empleo de un método de doble umbral.

En un sistema de RLbI, se requiere del robot que sea capaz no sólo de percibir, representar y aprender gestos, sino también de imitarlos. En la arquitectura propuesta la imitación implica una traducción de los movimientos de la persona al robot. El algoritmo de traducción híbrido implementado se inspira en conceptos de animación gráfica, es flexible y eficiente, y permite incorporar con facilidad nuevos criterios de traducción sin cargar excesivamente el sistema. El capítulo 5 detalla este algoritmo, y lo evalúa considerando distintos tipos de gestos.

El capítulo 6 presenta los resultados obtenidos cuando la arquitectura de RLbI completa se utiliza en escenarios reales. Finalmente, el capítulo 7 incluye una discusión sobre algunos aspectos destacados de la Tesis, y describe las líneas de investigación futura que emergen de ella.

La principal contribución de esta Tesis es el sistema de RLbI completo. Este sistema se ha validado mediante experimentos ejecutados en escenarios de RLbI reales. Incluye diferentes módulos, desde la percepción de los movimientos de la persona hasta el aprendizaje, en los cuales se han realizado aportaciones específicas. La arquitectura del sistema, en sí misma, presenta varias novedades en lo referente a su estructura y componentes. Los resultados de los experimentos demuestran que las principales limitaciones del sistema provienen del sistema de percepción utilizado, aunque por otro lado el sistema de captura de movimientos propuesto es capaz de seguir los movimientos de la persona en la mayoría de las situaciones, imponiendo sólo muy leves condiciones de inicialización. Los sistemas de reconocimiento y aprendizaje ofrecen resultados con rapidez, y no requieren entrenamiento previo. La persona, por otro lado, interviene en el proceso de aprendizaje supervisando el comportamiento del sistema en situaciones confusas. Aunque aún falta trabajo por hacer el sistema de RLbI propuesto, en definitiva, constituye una pieza que puede ser integrada de forma efectiva en un robot social, y combinada con elementos adicionales para alcanzar comportamientos de más alto nivel.



## Part II



Departamento de Tecnología Electrónica  
E. T. S. I. Telecomunicación  
Universidad de Málaga

TESIS DOCTORAL

VISION-BASED GESTURE RECOGNITION IN A  
ROBOT LEARNING BY IMITATION  
FRAMEWORK

AUTOR: Juan Pedro Bandera Rubio  
Ingeniero de Telecomunicación

DIRECTORES:

Juan Antonio Rodríguez Fernández  
Ingeniero de Telecomunicación  
Dr. por la Universidad de Málaga

Luis Molina-Tanco  
Ingeniero de Telecomunicación  
Dr. por la Universidad de Surrey



# Chapter 1

## Introduction

Researchers in the field of robotics have put a big effort in recent years towards including robots in everyday life. Thus, after decades of industrial use, the idea of robot begins to move closer to the artificial beings conceived by the first fiction authors who wrote about robots (Čapek, 1920). There are, however, many issues that must be addressed before robots are able to be used in real, day-to-day, human environments. One of the main of these issues is the impossibility to predict all situations that such robots will face. Solutions to this problem involve the use of learning mechanisms that allow the robot to adapt to dynamic conditions, new tasks, or human users. Biologically-inspired learning by imitation has become one of the main lines of research in robotics. In this thesis, a novel architecture for imitation learning is proposed, detailed and tested.

This chapter begins with a brief description of the context of the thesis: social robotics and its challenges. Section 1.2 sets the concrete objectives defined for the proposed system. Section 1.3 summarises the main contributions, and, finally, section 1.4 presents the organization of the thesis.

### 1.1 Motivation

Robots have been widely used in industrial environments to perform repetitive, predictable tasks that may be dangerous, disengaging or boring for human workers (Craig, 1986). The nature of these tasks allows to provide the robot with a complete *a priori* knowledge database. Industrial robots are reprogrammed only if the task they are performing changes.

The evolution in the control systems and the manipulation capabilities of the robots allowed industrial robots to evolve towards more complex systems, able to perform more complex

tasks. Robots have started to substitute humans in some dangerous tasks, such as cleaning mine fields, inspecting volcano craters or sealing submarine pipes. Although many researchers are addressing the topic of providing a higher degree of autonomy, these robots are usually teleoperated by a person (Ohya et al., 2009).

More than thirty five years ago, a new generation of robots began to appear. Japan, the country with the highest proportion of robots per person, started to invest in researching humanoid robots. The government, the industry and the universities began different research lines concerning the development of humanoid robots (Inoue et al., 2001). The general idea and main objective of this research was to include robots in everyday life, as partners (Breazeal, 2002), work companions (Kaneko et al., 2004), waiters (Maxwell et al., 1999), etc. Such robots, that have to relate to humans and work in human environments, could benefit from sharing certain human characteristics. Thus, bipedal walking, stereo vision-based perceptual systems or complex multi-fingered manipulators become important objectives for this research topic. Since the first results (Kato, 1973), research in humanoid robots has evolved towards meeting these objectives, contributing in this process with new platforms, controllers and architectures. These systems have also begun to be applied to different research fields such as medical rehabilitation, ergonomics, exoskeletons or human enhancement (Folgheraiter et al., 2009).

Despite these advances, it is debatable whether robotics research has succeeded at introducing robots in the home, the hospital or the museum. Commercially successful robots designed to work in these environments include robotic hoovers, pool cleaners, and entertainment devices (Simoncelli et al., 2000; Forlizzi and DiSalvo, 2006). Some of the most visionary researchers in robotics are pushing in this direction by introducing the concept of *social robot*. The most sophisticated social robot would be able to participate in society, interacting with us through standard social channels. The achievement of this long-standing goal of robotics still requires advancements in artificial intelligence, computer science and mechanical engineering. This thesis hopes to contribute towards this goal by introducing a novel architecture for imitation learning which allows a robot to learn —see, recognise and imitate— gestures in a social environment.

### 1.1.1 Social robots

Despite the undoubted utility of industrial and teleoperated robots, the early concept of what a robot should be considers much more versatile capabilities. The robots described in Čapek (1920) and other fiction sources are able to work in real uncontrolled human environments. They are able to perceive its surroundings and recognize objects and people, and the behaviours

they are exhibiting. They can relate to humans, and cooperate with them in solving everyday tasks. They can also learn from observation, from direct teaching, or even from inner revisions of perceived behaviours.

The usefulness of robotic companions able to meet at least some of the previous requirements is worth to consider. Robots that could react to uncontrolled environments could safely walk around in streets, offices or houses. Robots that interacted with people in a natural way would be able to become useful companions. If these robots were provided with dexterous abilities and multisensory input, they would be able to help people solving complex tasks. Probably the most important ability for these robots would be their learning capability. Robots that could learn from its observations and experiences, and from human teachers, would be able to adapt to new situations and perform new tasks, or improve already known ones.

As pointed out by [Breazeal \(2002\)](#), it is difficult to define a term that encapsulates all these characteristics and specifications. In this sense, the concept of *social robot* was firstly used more than fifty years ago to designate robots that were aware of its surroundings, although in these pioneer systems the robots did not recognize each other nor established explicit communication channels ([Walter, 1950](#)). From these contributions, two main research lines have been followed. The first of them considers collective or swarm robots, that conform anonymous groups of individuals that do not explicitly relate to each other but achieve indirect cooperation using biological principles like stigmergy (indirect communication via modifications in the shared environment) ([Fong et al., 2003](#)). The second research line considers robots that work in individualized societies. In these societies each individual has its own motivations, interest and goals, and sets its own relationships. These relationships conform a complex social network constrained by social conventions and norms that the agent knows and applies -or not- depending on each particular situation. Social robots are currently related to this second type of society, as depicted in the following definition proposed by [Dautenhahn and Billard \(1999\)](#):

Social robots are embodied agents that are part of a heterogeneous group: a society of robots or humans. They are able to recognize each other and engage in social interactions, they possess histories (perceive and interpret the world in terms of their own experience), and they explicitly communicate with and learn from each other.

According to this definition, social robots are not only aware of their surroundings, but also able to recognize, learn and communicate with other individuals. There are, however,

different levels of social behaviour that may exhibit one of these robots. Thus, [Breazeal \(2003\)](#) describes different classes of social robots:

- *Socially evocative.* These robots focus on evoking nurture, care or empathy feelings, more than on offering complex social behaviours.
- *Social interface.* These robots provide a natural and intuitive interface for humans, but they do not offer deeper social capabilities. It is possible at that level to implement robots that can substitute biological pets ([Ryokai et al., 2009](#)), even in therapeutical scenarios ([Wada and Shibata, 2007](#)).
- *Socially receptive.* These robots are able to obtain benefits from interaction (e.g. learn by imitation), although they do not have their own motivations and goals, and thus they do not actively engage with humans unless told to.
- *Sociable.* According to the definition proposed in [Breazeal \(2002\)](#), "A sociable robot is able to communicate and interact with us, understand and even relate to us, in a personal way. It should be able to understand us and itself in social terms. We, in turn, should be able to understand it in the same social terms". These robots require complex cognition structures.

It can be seen that [Breazeal \(2003\)](#) describes robots that interact with people. This is a more particular situation than the generic 'individualized societies' described by [Dautenhahn and Billard \(1999\)](#). The reality is that the vast majority of social robots are designed to work in human societies, where the presence of other social robots is usually reduced, if any. Thus, these 'others' should be understood as 'people'. In this thesis we follow this practical concept of social robot, that has to interact with humans. [Fong et al. \(2003\)](#) also follow this concept. They also complement the previous classes with the following ones.

- *Socially situated.* Robots that are placed in social environments, and that are able to perceive and react to them.
- *Socially embedded.* These robots are very similar to previous ones, but they can also be partially aware of human interactive behaviours.
- *Socially intelligent.* These robots are provided with complex detailed models of human cognition and social competence. They may show aspects of human social intelligence. These robots are still more a long term objective than a practicable realization.

The previous taxonomy shows that social robots conform a wide, complex group, that includes very different agents. They share, in any case, certain common characteristics, thus it may be possible to provide a definition that contains all of them. In this thesis, social robots are understood as *robots that work in social environments, and that are able to perceive, interact with and learn from other individuals, being these individuals people or other social agents*.

It is in the context of social learning and, more specifically, of learning by imitation, where this thesis contributes. As it is detailed in the next chapter, social learning in people mainly relies on two perceptual inputs: speech and vision (Schaal, 1999). In order to fit the requirements imposed for social robots, regarding intuitive and natural interaction and adaptability to human social environments, social robots should also focus on these perceptual channels. Speech recognition and synthesis has been deeply and successfully addressed in previous work (Breazeal et al., 2004).

Visual perception presents important issues with respect to speech. One of the main differences is that sound signals can be coded with high fidelity using a relatively low bandwidth, while vision signals require more resources to be stored and processed (Moeslund et al., 2006). However, the vision system is a key part in a social robot, that has to detect and recognize potential interaction partners and objects of interest, and perceive, recognize and learn social gestures. These tasks have also to be achieved in complex real scenarios.

This work aims to provide a framework for recognition, imitation and learning of gestures for a robot in specific conditions that we believe are close to those that may be encountered in *real*, day-to-day scenarios such as the home or the office. These environments impose a set of constraints which are revised in the following section.

### 1.1.2 Constraints imposed by social environments

The environments in which social robots are supposed to be used impose special constraints on them. Not all constraints will apply to all situations, and it is probably recommendable to focus on dealing with only a subset of them (Calinon, 2007). This thesis, however, proposes a concrete architecture that takes into account the following list of constraints imposed by social environments:

- *Structural variability*. New objects may appear in the environment. Existing objects may be moved, replaced, or removed.
- *Dynamic lighting*. Light sources may appear and disappear in unpredicted locations. The

characteristics of existing light sources are variable.

- *Uncontrolled human-robot distance.* Interaction distances are not fixed. The social robot should identify potential interaction partners located at different distances and react to them accordingly, without imposing the users to stand at certain distances.
- *Limited a priori knowledge.* The social robot has to adapt to new situations, and learn new tasks, gestures or behaviours. Thus, although in certain situations it may be possible to provide it with certain *a priori* knowledge, it should be able to extend it from its experience.
- *Limited perceptual capabilities.* No perfect sensors are employed in the social robot. Calibration errors, distortion and noise affect the quality of the perceptual inputs.
- *Non-invasiveness.* People that interact with social robot should not be required to wear specific markers, garments or color patches.
- *Untrained users.* People should interact with the social robot in a natural and intuitive way. No specific training should be required. In this sense, learning by imitation appears as a powerful mechanism to allow the robot learning new behaviours.
- *Interaction with unpredicted users.* Social robots should not only be aware of known people in their surroundings. They should also be able to interact with novel users, even if these interactions may be constrained in certain robots.
- *On-line response.* People expect to receive fast feedback and responses from their interaction partners. The social robot should be able to work at human interaction rates.

These constraints describe the scenarios in which the social robot will have to see, recognise and imitate a set of gestures as part of its social interaction with humans. In the remaining of these thesis we will refer to this scenarios as Robot Learning by Imitation (RLbI) scenarios.

The characteristics of RLbI scenarios have guided the development of the architecture for imitation learning presented in this thesis. The tests and experiments designed to verify this architecture will show the degree of immunity to the challenges imposed by social environments.

## 1.2 Objectives of the thesis

Social robots that have to work in RLbI scenarios need to adapt to new situations, and to learn from human teachers in natural and intuitive ways. A basic assumption is that learning by

imitation appears as one of the most adequate mechanisms to achieve these objectives (Schaal, 1999). Both speech and vision play an important role in social learning in general. However, in learning by imitation speech is usually employed only to reinforce the visual demonstration. Sometimes this visual demonstration does not use speech reinforcement at all. Thus, it is interesting to implement a learning by imitation system that uses only vision, and that can be later extended to include speech commands. This is the main objective of this thesis: the implementation of a vision-based gesture RLbI system, that can be used by a social robot. This general objective is decomposed in the following sub-objectives:

- Develop a vision-based human motion capture module that uses as only input the images obtained by a stereo vision system mounted in the head of a social robot. This system should be able to detect people at human interaction distances and generate a representation of the gestures they are performing in RLbI scenarios.
- Develop a low dimensionality representation of perceived gestures that stores all relevant information for recognition and imitation.
- Find a method to achieve gesture recognition in scenarios in which no intense training is available, and *on-line* response is required. The method should use a learning mechanism that allows for augmentation and modification of the gesture repertoire.
- Define a method to translate the perceived human motion to robot motor commands, so that *imitation* of the gestures is possible when required.

The implementation of a complete social robot is a very complex endeavour. Many of the indispensable elements of a social robot lie beyond the scope of this thesis, which focuses on gesture recognition and learning. Some of the main issues that will not be covered are detailed below.

- Speech recognition and synthesis is not considered.
- This thesis does not deal with the physical implementation of the social robot.
- Gestures considered in this thesis do not involve interaction with objects, and are executed by only one performer simultaneously. They involve the upper-body part of the body, but leave out facial gestures.
- The system is not tested in outdoor environments. Some issues that are present in these situations (e.g. direct sunlight) require further work to be successfully addressed.

- Higher level cognition layers are required to decide whether a certain perceived gesture should be imitated or not, or whether a certain learnt gesture should be executed in certain situations or not. These levels lie beyond the objectives of this thesis, that focuses instead on the gesture learning mechanism itself.

This work will hopefully serve as the basis over which some of these issues can be addressed. For example, higher level decision layers will benefit from using a robust and functional lower level recognition layer.

### 1.3 Contributions of the Thesis

This thesis focuses on the implementation of a complete RLbI system. The architecture of the system itself differs from previous approaches in the field of social robotics. Thus, instead of following the extended idea of encoding the perceived gestures in the robot motion space, this thesis proposes to use a human model. While this forces the social robot to have a deeper knowledge of human kinematics, it increases gesture recognition rates and allows social robots that are not humanoid to successfully perceive and interpret human gestures. Translation of perceived motion from human to robot becomes an independent task respect to perception, and thus it is not necessary to execute it unless the robot is required to imitate the movements.

It is complex to fit RLbI scenarios requirements using a perceptual system composed only by a pair of stereo cameras. In this thesis a novel vision-based HMC system is proposed and quantitatively tested. This method incorporates a torso pose estimator based on anthropometric tables and disparity maps, and also an IK algorithm to pose the arms, that includes a novel alternative pose evaluation method.

Once human motion is successfully perceived, it is still necessary to encode it in an efficient space. This thesis proposes a novel gesture representation system based on different features that consider both global and local characteristics of the gesture.

Gesture recognition is also achieved adapting the widely used Dynamic Time Warping (DTW) technique to the comparison of local features. Comparison of global features reinforce these local distances. The results of this proposed method prove that it is a robust and interesting alternative to previous approaches.

Finally, a new combined strategy, based on computer graphics animation techniques, is used to translate the motion from the human to the robot. This approach is easily scalable and

adapts to different types of gestures.

It is important to remark that special care has been put in testing the system in RLbI scenarios. While it has limitations that are discussed further, the proposed RLbI system has been able to adapt to different untrained users, and to dynamic real indoor environments. It has been tested in several live demos performed at different locations. It is able to learn from few or none training samples, and can adapt to different types of gestures.

The previous contributions have produced several publications. A complete list of these publications is given in appendix [G](#).

The work of this thesis was developed in the context of the TIN2004-05961, TIN2005-01359 and TIN2008-06196 projects by the Spanish Ministerio de Ciencia y Tecnología (MCYT) and FEDER funds, projects P06-TIC-2123 and P07-TIC-03106 by Junta de Andalucía, and project VISOR by the European Robotics Research Network (EURON), in the ISIS group (*Grupo de Ingeniería de Sistemas IntegradoS*) at the University of Málaga (Spain).

## 1.4 Organization of the Thesis

The main contribution of this thesis is the RLbI architecture itself. Chapter [2](#) describes previous approaches and discusses their advantages and disadvantages. Then, the proposed architecture is deeply described.

Chapter [3](#) describes the Human Motion Capture (HMC) system needed to extract human motion from perceived images. The chapter also provides a quantitative evaluation of this system.

Captured human motion has to be segmented into discrete gestures. Then, these gestures have to be encoded in an efficient representation. Memory and learning components use this representation to recognize the gesture, update the knowledge database of the robot and learn new gestures. Chapter [4](#) details these steps of the RLbI process.

The robot should not only be able to perceive, recognize and learn human gestures. It should also be able to imitate these gestures, i.e. translate them to its own motion space, and reproduce them. Chapter [5](#) describes and tests the algorithms proposed in this thesis to achieve these tasks.

The components of the proposed RLbI architecture has been independently detailed

and tested, but experiments involving the complete system working in RLbI scenarios are also required. Chapter 6 details the integration of the different components in the complete architecture, and presents the results obtained when the proposed system is used in RLbI scenarios.

Chapter 7 discusses some general aspects related to vision-based RLbI systems. Finally, chapter 8 concludes the thesis and summarizes the main obtained results.

## Chapter 2

# System architecture

### 2.1 Outline of the chapter

This chapter describes the RLbI system proposed in this thesis. As detailed in the previous chapter, RLbI mechanisms represent a key addition to any social robot as it needs to be provided with a learning system that is natural and intuitive for human teachers, and that allows to quickly and efficiently incorporate new knowledge units to the repertoire of the robot.

Social learning mechanisms are present in biological entities. These natural mechanisms have largely inspired the ones proposed for artificial beings. Thus, section 2.2 briefly introduces social learning, and discusses the adequacy of some of the introduced concepts to social robotics.

Section 2.3 starts from the first approaches to learning in robotics and describes some of the main contributions in this research field. These contributions have driven artificial learning systems from purely engineering architectures to biologically inspired systems. One of the main influences for these last systems are the social learning mechanisms detailed in section 2.2.

The increasing interest in social robotics have motivated the apparition of different RLbI architectures. Section 2.3 describes some of the main of these architectures. These approaches inspired a first RLbI architecture that was developed in the context of this thesis, and that is detailed in [Bandera et al. \(2007\)](#). The process of implementing and testing this approach revealed certain issues and opened a discussion process, summarized in section 2.5, that pointed towards the necessity of including certain modifications in the proposal. These modifications have been recently implemented, and finally produced the RLbI architecture presented in this thesis, that is detailed in section 2.6.

## 2.2 Introduction to Social Learning

The ability to imitate and learn from imitation is present in humans and other animal species since childhood. It plays a very important role in the social development of an individual, to the point that in some animal species, individuals that are not able to imitate, or that have not learnt from imitation, do not survive unless breed in captivity (Mosterín, 2005).

The RLbI systems developed for social robots are inspired by these natural social learning mechanisms. Thus, it deserves to briefly introduce the main concepts, classifications and theories about these mechanisms before describing RLbI systems.

### 2.2.1 Culture

The Latin verb *colere* means 'to cultivate'. That root originated terms like 'agriculture', from *agrum colere* (land-growing) or 'viticulture' (vine-growing). The term implies to put special and constant care in a certain task, and thus the Latin substantive *cultus* began to be applied to the careful actions the priests performed to take care of gods. This religious meaning of the word favored in later centuries the metaphor that compared the soul or spirit of a rude man with a non-cultivated field. In opposition, 'cult people' were people which souls had been cultivated by religion and knowledge or, in general, by an educational process provided by the society in which these people lived.

Neglecting the recent use of 'culture' as 'pastimes of well-educated people', that can be found in some contexts, here the scientific -anthropological and biological- meaning of culture is used. From this point of view, 'culture' is "*the set of activities, procedures, moral concepts and ideas that are transmitted by social learning and not by genetic heredity*" (Mosterín, 2005). Thus, culture becomes a wide concept, that in fact includes all the previously mentioned ones (agriculture techniques, religion, pastimes, etc.).

It can be seen that this scientific definition of culture emphasizes its social, learnt character. Thus, it is opposed to the innate or congenital abilities, the natural (from the Latin word *nasci*, meaning 'to be born') abilities a certain individual possesses since its birth. In the field of social robotics these concepts can be directly translated. The *natural* abilities of a social robot would include its motor commands, perceptual channels and some related processing modules. Some authors include the mapping from perception to action, or visuo-motor mapping, among the natural abilities of a social robot (Lopes and Santos-Victor, 2005). The *culture* of a social robot, on the other hand, would be the set of behaviours, activities, procedures, tasks, gestures,

etc. that it learns from observation of its environment and, specially, the people with whom it interacts.

### 2.2.2 Cultural evolution. Memes

The term *meme* was firstly introduced by Dawkins (1976), as a complementary concept to 'genes'. While genes are the units of genetic information, memes are the units of cultural information. While biological evolution is, fundamentally, the evolution of genes, cultural evolution is the evolution of memes. But genes are transmitted only vertically, from parents to children. Memes, on the other hand, are transferred both vertically and horizontally from one individuals to others, conforming a cultural net that contains multiple individuals. This net changes dynamically due to different influences, or forces, that include communication, isolation, individual elections, culture decay and coaction.

One long term objective for social robotics may be the inclusion of social robots in the cultural nets that its human partners have already established. It could also be possible to imagine cultural nets that are exclusive to robots. These goals, however, are usually considered to be still too far away from the current objectives and possibilities of social robots. On the other hand, social learning mechanisms have proven to be useful to efficiently transmit knowledge to the robots (Schaal, 1999), and thus these mechanisms are more deeply described in this section.

### 2.2.3 Individual learning

It is possible for a certain individual to acquire information by itself, i.e. without requiring a social context. This individual learning procedure includes trial and error methods, imprinting, classical conditioning, etc. (Mosterín, 2005). These forms of learning benefit from natural predisposition. In general, animals are preprogrammed to feel pleasure when performing beneficial behaviours, and pain when performing potentially dangerous or not convenient ones.

There are, however, certain problems in individual learning. While trial and error method is a powerful mechanism, it may be also very dangerous if some of the alternatives to explore are potentially deadly. Even if this is not the case, the animal has still to spend a considerable amount of time and energy in the process. Approaches that apply this learning method to artificial agents that cooperate with humans have to consider not only the previous disadvantages, but also the risks that involve for people in its surroundings a machine that learns by using trial and error (Kuniyoshi et al., 2003).

### 2.2.4 Social learning

Culture is transmitted by social learning. This learning procedure includes all the information a certain individual acquires from others through imitation, communication or teaching. Social learning drastically reduces the risks implicit to trial and error methods. It also allows to learn new tasks and behaviours faster, as a certain animal can just imitate a demonstrator instead of going through individual imitation. These advantages are not only applied to each individual, but also to the society itself as a group (Mosterín, 2005). Thus society, thanks to social learning, can obtain nearly immediate (from the point of view of evolution) benefits from the ideas and experiments of its most imaginative or curious individuals.

As commented above, there are different types of social learning. Mosterín (2005) describes the following.

- **Imitation.** Social learning by imitation is the process in which an imitator learns by observing the behaviours performed by other animal of the same specie. It is interesting to emphasize this last point: an animal learns from its kind, as they share necessities, innate abilities and perceptual and motor skills. In this sense, imitation as a learning mechanism in social robots faces an important issue: the robot imitator is very different to the human performer. These differences affect the perceptual capabilities, the kinematics and dynamics systems, the cognitive structures, etc. Thus, robotic imitation should carefully consider that either the social robot perceives itself as a human (i.e. it uses an inner representation of a human to imitate the human performer), or its imitation learning mechanism considers the differences between the perceived performer and the robot (Meltzoff and Moore, 1989).
- **Teaching.** In imitation learning the performer or model is passive, and there is no evaluation of imitated behaviours. The teaching process, on the other hand, involves social learning scenarios in which the imitated performer is active, and distributes positive reinforcements (rewards) and negative reinforcements (punishments), according to the accuracy of executed imitations. Thus, in teaching processes exhibition of proper behaviours is positively reinforced, while incorrect behaviours are negatively reinforced. Reinforcement, and the degree of supervision it implies, is an interesting addition to the majority of RLbI scenarios (Breazeal, 2002).
- **Telecommunications.** In humans, it is possible a third form of social learning, in which the information transmitted by a certain performer can be simultaneously acquired by

many imitators that are located in very different places. This social learning procedure includes transmission of knowledge through books, radio, television, internet, discs, etc. This procedure, however, is not usually integrated in social robots, probably due to the current lack of situations in which one single human has to communicate with many social robots. It is also difficult, when using telecommunication learning, to control the learning process to an adequate degree.

The previous classification correspond to a certain theory about social learning and imitation. There are, however, other authors from different research fields that have proposed and followed different theories. These proposals are hindered by the limited knowledge about the psychological processes involved in imitation learning. In absence of consensus, the main of these theories about social learning (Huertas, 1992) are further explained. They are similar in certain aspects, but they differ in others. This thesis does not select a particular theory, but it analyzes whether the different categorizations and steps they propose are more or less suitable to be integrated in the cognitive architecture of a social robot.

### 2.2.5 Instinctive imitation

Humans, and other animals, imitate from early ages. Meltzoff and Moore (1989) demonstrate that babies aging between 2 and 3 weeks are able to perform gesture imitation. Many psychologists stated that this ability can be explained only if a natural tendency and ability to imitate is present in these children by nature.

The theory of instinctive imitation is also focused on the study of imitation in animals (Galef, 1988). In this area, Thorpe (1963) proposed the three following categories of social learning.

- **Social facilitation.** A behaviour that is already present in the repertoire of the observer is stimulated by the observation of a certain demonstrator. E.g. an animal may learn that a certain food can be eaten if it sees another animal of its kind eating that food.
- **Focalization.** Even if no explicit demonstration is present, the behaviour of a model can drive the attention of the observer to certain objects, places or events. This increases the speed in the learning process.
- **True imitation.** According to Thorpe (1963), this is the imitation of an unusual behaviour, that was not present in the repertoire of the observer and that could have been

hardly acquired by trial and error.

While some previous knowledge can be stored *a priori* in a social robot, it is not possible to predict all possible situations this agent will face. Thus, it is necessary to equip it with the ability to perform previously mentioned true imitation. Focalization and social facilitation, on the other hand, may be very interesting additions to the capabilities of a social robot, that make it become a more curious and autonomous being, able to learn from non-explicit demonstrations. However, these categories of social learning imply important practical issues. For instance, to achieve focalization the robot needs not only to perceive the human performer, but also to share his/her focus of attention, identify referenced objects, etc. (Breazeal et al., 2004).

### 2.2.6 Operant conditioning

Miller and Dollard (1941) propose to consider social learning as a specific case of operant conditioning, that follows the scheme 'discriminative stimulus - response - reinforcement'. Thus, the behaviour of the model is the discriminative stimulus, and the imitator behaviour the response. The probability of imitation increases when positive reinforcement is applied, and viceversa.

Some authors argue that it is not possible to explain certain social learning situations using this theory, e.g. a reinforcement applied over the model affects the behaviour of the imitator (Bandura, 1969). Some of these arguments have been refuted by other authors. However, it is still difficult to explain, using this theory, how an observer learns from behaviours that are not executed by itself, or how these behaviours can be reinforced by stimulus that are not directed to the observer.

### 2.2.7 Cognitive imitation

Bandura (1969) proposed a social learning theory that he named *cognitive-social*. The basis of this theory is that the model behaviour has an informative function. In the social learning process the observer, then, obtains a set of symbolic representations of the behaviours performed by the model. These representations can later be adapted by the observer, to address similar situations in which it plays the role of the model perceived before.

This process is composed by the parts detailed below (Bandura, 1969).

### 2.2.7.1 Attention

It is necessary, to achieve social learning, that the observer focuses its attention in the significant components of the model behaviour, and that perceives them correctly. The characteristics and state of both the observer and the model influence perception and attention. Thus, this should be considered an active process.

Perception, including attentional mechanisms, is a key component of social robots. In fact, the difficulties that imply trying to process all data provided by the robot visual sensors impose the use of the latter, to achieve *on-line* responses. For instance this thesis, that deals with RLbI, will use an attention mechanism that extracts only the motion of a human performer, from the images perceived by the robot visual input.

### 2.2.7.2 Memorization

An observed behaviour may be of little use for the imitator if it is not able to remind it later. The memorization of behaviours requires to encode them in a certain symbolic representation, in which all important data are preserved, while superfluous information is discarded. Revision of learnt behaviours, in which the observer imagines itself in the perceived situation, are useful to refine these representations, and to increase the performance of the observer in executing these behaviours. Thus, [Smyth and Pendleton \(1990\)](#) state that in motion learning the observer needs to perceive the performed motion, process spatial and temporal relations between different body parts, and retain these data. They introduce an interesting difference between *location movements*, where the objective is to define a point in the space (e.g. pointing) and *configured movements*, in which it is more important to preserve relative motions and body parts configurations (e.g. dancing). This classification will be later used in this thesis to transfer different types of movements from the perceived human to the robot imitator. [Smyth and Pendleton \(1990\)](#) demonstrate that the codification of configured movements is disturbed by other configured movements being perceived at the same time. However, the interference is lower if only location movements are perceived. This implies that cognitive revision is present in social learning, and that these processes are partly independent from other low term cognitive tasks, thus they may be considered a differentiated part in the *cognitive-social* theory of [Bandura \(1969\)](#).

[Bandura \(1969\)](#) states that social learning is based on two representation systems: images and speech. In humans, images are more important in the first phases of the development, although speech gives a strong boost to learning processes in later stages. A social robot, thus,

should consider both systems and understand both gestures and verbal commands (Breazeal, 2002).

### 2.2.7.3 Generation

The third type of processes involved in social learning translates the stored symbolic representations to actions. This translation includes corrective adjustments of executed behaviours, to approach them to stored representations. In order to perform this feedback comparison between memorized behaviours and actions, it is necessary to map executed actions in the same symbolic representations used to store perceived behaviours. This leads to different architectural alternatives when designing social robots, as will be detailed in further sections.

### 2.2.7.4 Motivation

The last element in cognitive social learning as defined by Bandura (1969) is motivation. This process explains why learnt behaviours are not always executed even when the circumstances match the ones observed when the behaviour was learnt. Thus, the execution of a certain behaviour is influenced by three different stimulus: direct, vicarious or observed, and produced by the individual itself. The probabilities of reproduction of a certain perceived behaviour increase if it provides positive results for the observer or the model, or if it satisfies the inner necessities or preferences of the individual.

It is not easy to determine whether motivation should be part of a social robot or not. If social robots are understood not as partners but tools, motivation may appear as an unnecessary, or even disturbing, element (e.g. it would not be desirable to use a hammer that could decide whether it struck the nail or not depending on its inner state). On the other hand, from a purely theoretical point of view, it can be argued that the social robot should possess these motivations in order to increase its resemblance to humans and ease interactions and cooperation. People may prefer to relate to social robots that have their own motivations, goals and even necessities, as they can be more easily understood in a social context. E.g. a social robot serving tea in a restaurant should not only attend the orders of the clients. It should also check its batteries, the tea temperature, the closing time of the restaurant, previous orders, etc. and prioritize its behaviours according to all these motivations. Finally, from a practical point of view, motivations may offer the social robots new cognitive and behaviour possibilities. This can increase its capabilities and usefulness.

### 2.2.8 Factors that influence social learning

The results of social learning can vary drastically depending on the characteristics of the scenario, the model or the imitator itself. Huertas (1992) describes different factors that may influence the social learning process. Between these factors the following are highlighted.

- **Model.** The characteristics of the model strongly influence its probabilities of being imitated. Dominant people, charismatic leaders, or individuals that provide more rewards (positive reinforcements) are more probably selected as models to imitate. There are evidences that individuals that occupy a prestigious or popular role in the society are, in fact, imitated even in the behaviours in which they are not particularly dexterous. In these cases imitation is clearly stimulated by the characteristics of the model itself, regardless the convenience of the imitated behaviour for the imitator.

It is also important to consider that the resemblance between the model and the imitator is also an important factor in imitation processes. Observers tend to imitate those models whose characteristics are closer to them, e.g. children usually prefer to imitate other children of the same sex and interests. The similarities between demonstrator and imitator are again highlighted as an important factor for successful imitation, as in Meltzoff and Moore (1989) or Mosterín (2005).

- **Observer.** The characteristics of the observer itself seem to play a certain role in the imitation process. Small children tend to imitate older classmates. As pointed out by Bandura (1969), this does not mean that imitation is more commonly found in insecure people. Some authors state that this may be true when dealing with simple tasks, but it is not true, in any case, if complex tasks are considered.
- **Situation.** Uncertain or unknown situations stimulate the use of imitation (Huertas, 1992). Besides, it is more probable that an individual imitates tasks of medium difficulty. Easy tasks do not require social learning, while too complex tasks are not interesting for the observer as it is not able to reproduce them (Mazur, 1986).
- **Functional value of the imitated behaviour.** It is usual to imitate these behaviours that provide certain success or satisfaction, even when this reward is not directly linked to the behaviour (e.g. social popularity). This last factor may easily override the rest ones in certain scenarios. Thus, a behaviour that implies negative effects for the observer may not be imitated even when the characteristics of the model are ideal. The prestige of the model may also be affected in these cases.

In the cognitive architecture of a social robot, these factors should be included as high-level parameters that influence the imitation process. The last factor, however, raises very similar problems to those encountered when considering inclusion or not of motivation.

These issues may be generalized in the following dilemma: social robots should follow orders even if they may produce negative effects for them. But they may possess certain degree of survival instinct. The three laws of [Asimov \(1942\)](#) are currently the most popular answer to this problem in the robotics community. These laws are subject to discussion, and many authors in the robotics research field argue that they are correct for literature, but not valid in real applications (e.g. if the robots have to obey the First Law, they should govern the people to avoid wars). They are, in any case, not practical as no robots are currently able to reach the knowledge and perceptual levels required to meet these laws. On the other hand, they imply that social robots should reach a certain equilibrium between self-interest and functionality, and this may be an useful concept to be incorporated to the design of social robots.

## 2.3 Evolution of artificial social learning. From Programming by Demonstration towards socially interactive robots

Social robots are the last step in a progressive research line that started years ago in a very different context. Industrial robots are the base over which complex humanoid robots have been developed. Similarly, techniques proposed decades ago to ease software development have been the inspiration for the current RLBI systems mounted on social robots.

Social robots have evolved from these roots in the last years. Their current controllers, motor systems and sensory inputs differentiate them from industrial systems. Their knowledge and learning systems have also become complex architectures inspired not only by engineering concepts, but also by insights taken from neuroscience and social sciences.

This sections starts introducing early works in Program by Demonstration (PbD), and details how these systems evolved to the current state-of-the-art in social robotics.

### 2.3.1 Early works. Program by Demonstration (PbD)

Program by Demonstration (PbD) appeared nearly two decades ago to ease software development. The idea behind this technique is to create software that is able to learn, from human demonstrations, how to solve a certain task (Cypher, 1993; Mitchell et al., 1994). This approach was inspired by the observation that most computer users are skilled at controlling it at an interface, application level, while only a few of them are able to re-program or modify the contents of that computer. Instead of trying to provide easier programming interfaces, the PbD approach proposes to provide the computer with mechanisms to learn differently. In these new learning scenarios the user simply solves a certain task, while the computer observes, records and analyzes human actions in order to infer the processes that lead to the solution of the problem (Cypher, 1993). These processes are usually encoded as a set of "if-then" rules, that the computer should be able to generalize to slightly different scenarios or situations, e.g. by using *feed-forward neural networks* (Mitchell et al., 1994).

The robotics community soon demonstrated an understandable interest in the PbD paradigm. Industrial robots required a complex and careful programming process to be able to perform even simple tasks, that people can reproduce easily. It was a desirable approach, then, to provide these robots with the ability to learn from human demonstrations. There was, however, an important difference between software PbD scenarios and robot PbD scenarios. In the former, the demonstrations and the reproductions are performed on the same medium (the

computer interface). On the other hand, the latter imply a different *embodiment* from the performer to the imitator. It may be difficult to imitate human actions using an industrial robot for which kinematics and dynamics are completely different. Social learning theories already predicted these issues (Meltzoff and Moore, 1989; Huertas, 1992; Mosterín, 2005), that were referred in the robotics and computer graphics communities as the *correspondence problem* (Nehaniv and Dautenhahn, 2002). Thus, any PbD system for a robot has to address this problem of translating a motion or task performed in the human motion space, to the robot motion space. Different approaches have been proposed to deal with this problem (Alissandrakis et al., 2007), being one of the first ideas to move the robot, through a dedicated interface or manually, through a set of relevant configurations or key points, that the robot should reach sequentially to solve the particular task. This teaching method is described in Calinon (2007) as *kinesthetic teaching*, and it is still widely used, in scenarios in which it is possible to count with a high degree of human supervision during the learning process. There are, however, other learning mechanisms that may not require a so direct, or intrusive, participation of the human performer, but may require more complex robot perceptual capabilities (e.g. vision-based HMC systems) (Kojo et al., 2006; Azad et al., 2007a; Hecht et al., 2009).

### 2.3.2 Generalization of learned tasks

The previously mentioned PbD methods can be applied to program the movements of an industrial robot in a controlled environment (Muench et al., 1994). However, the results are not usable if the robot or the environmental conditions change. New demonstrations should be required to adapt to each of these variations. On the other hand, it may be a more reasonable and powerful approach to consider some degree of *generalization* in the learning process. The first attempts of generalizing the learned skill involved queries about the intentions and goals of the user (Heise, 1989). These goals, and the characteristics of the task itself, may be defined at different levels of abstraction. These levels have been finally classified in two categories: *symbolic level* and *trajectory level* (Calinon, 2007).

A certain task can be described at a symbolic level by a set of predefined sequential states or action primitives (Alissandrakis et al., 2007). This approach allows to produce more abstract representations of a perceived task. However, obtained representations depend too much on pre-determination of observed cues and segmentation. Besides, it is complex to set the optimal granularity level to represent generic tasks. Thus, this approach faces important issues when addressing generalization.

Trajectory level representations, on the other hand, encode a certain task as a continuous stream of sensed data (Ude et al., 2004). These data may include end-effector or relevant body parts positions, joint torques, end-effector orientations, etc. The levels of abstraction that are achieved when using this approach are usually lower than the ones reached when using representations at symbolic level. Besides, encoding a task at trajectory level usually involves a higher amount of data, that usually have to be filtered using dimensionality reduction techniques. However, trajectory level representations are more suitable to achieve generalization, as they are not constrained by the predefined segmentation units. They are also easier to refine when new demonstrations of the tasks are observed, or by cognitive revision procedures (Calinon, 2007).

While human supervision is an important element in the learning process, generalization of observed tasks may be favored by a robot that is able to automatically learn to recognize complex patterns and make decisions based on data. Thus, Machine Learning (ML) algorithms began to be used in robot PbD systems (Thrun and Mitchell, 1993). This tendency allows to test ML techniques with multiple example of input/output (sensors/actuators), while robotics benefitted from the abilities of ML to deal with generalization and multivariate data. One important contribution in this combined research field is the proposal of Muench et al. (1994), who suggested to use what they called Elementary Operations (EOs) to encode tasks at a symbolic level. EOs are integrated in more complex and general structures in a learning process. This process selects only EOs that appear repeatedly in the demonstrations. On the other hand, the integration is achieved by relating EOs using, mainly, conditional branches. There is, then, a search for the underlying structure behind the demonstrations, that Muench et al. (1994) describe as a key part of the learning process. The extraction of relevant dependencies and relations at a symbolic level is usually defined as the *functional induction* problem (Dufay and Latombe, 1984), and has been widely addressed since its formulation (Nicolescu and Mataric, 2003; Saunders et al., 2006).

As pointed out by Muench et al. (1994), there are important differences between generalizing examples at higher levels, and generalizing elementary operations such as force-controlled movements or contour-tracking operations. The later requires the learning of continuous numerical functions, in which completeness or sufficiency of the examples may be more important, but more difficult to achieve. Muench et al. (1994) propose to increase the importance of the human supervision. The user, then, constraints the demonstrations to those that are understandable by the robot, takes care of providing enough demonstrations, preprocesses the examples, guides the learning process and evaluates the learning results. These requirements may be adequate for industrial robots. However, they may be very constraining in other scenarios, even when

social learning points towards the importance of a certain degree of supervision in the learning process.

### 2.3.3 Humanoid robots. From PbD to RLbI

While not all the social robots have to be humanoid robots, the rise of humanoid robots effectively marked the beginning of social robotics. The construction of the first humanoid robot, the WABOT-1 (Kato, 1973), more than thirty-five years ago, started a process that has evolved towards a new concept of robot. Research in humanoid robotics seeks to move robots from complex mechanisms to engaging interaction partners, from potentially dangerous machines to useful work companions, from industry buildings to restaurants. Humanoid robotics aims at developing robots intended to cooperate with humans in everyday task, in real human environments. The long term goals for humanoid robots include robots that assist elder or disabled people (Mohan et al., 2008), work as diplomats<sup>1</sup>, prepare meals or clean the house (Asfour et al., 2006a), work as restaurant waiters (Maxwell et al., 1999) or become an engaging opponent for a professional football team<sup>2</sup>. It is interesting to consider that these ambitious objectives have been present in this research field nearly from its beginning, probably due to the high prospects that inevitably arise when facing a robot that resembles a human being. The development of a humanoid robot, however, has to face many complex issues before being able to address these high level goals.

As commented above, humanoid robots are intended to work in real environments, and to cooperate with humans in everyday tasks. It may be interesting, to achieve these objectives, to provide the robot with a bipedal locomotion system, instead of wheels or other options, as it may ease adaptation to human environments. Biped locomotion presents, however, complex issues to researchers regarding stability, energy consumption and robustness. The Zero Moment Point (ZMP) stability criterion was widely used in the first steps of this research (Vukobratovic and Borovac, 2004). However, it reached its limits years ago, as it was not possible to use it, for instance, to model a running gait. In a recent contribution, Sugihara (2009) refers to previous work to state that ZMP criterion is not sufficient, not even necessary to grant stability. It may still be used as an approximative measurement in certain situations. In any case, researchers are currently proposing new biped locomotion algorithms based on different criteria, such as

---

<sup>1</sup>Probably the most famous example of protocol robot is the ASIMO, from Honda <http://world.honda.com/ASIMO/>.

<sup>2</sup>The objective of Robocup<sup>®</sup> project is: 'By the year 2050, develop a team of fully autonomous humanoid robots that can play and win against the human world champion soccer team'. More information in <http://www.robocup.org/>

linear model predictive control (Dimitrov et al., 2009) or central pattern generation (Cheng et al., 2006). These approaches allow not only to generate stable gaits, but to grant balance when the robot suffers from unpredicted disturbances (e.g. a person pushes the robot). In any case, the absence of a definitive solution to the biped locomotion issue moves many researchers to substitute the legs by wheeled platforms (Asfour et al., 2006b), that provide the robot with enough locomotion abilities as to work in most real urban environments.

Manipulation skills are also very important for a humanoid robot. The problem of catching a box of a certain shape using a grip becomes the problem of understanding and reproducing the grasping human movements. A body of work concentrates on evaluating different types of grasping (Aleotti and Caselli, 2006; Faria et al., 2009), considering the object to grasp and other factors (e.g. in some cultures the social role of the individual determines how he/she should grasp a cup of tea). It is also important to evaluate the pressure that has to be applied to ensure grasping without breaking the object, but also without letting it slip (Yussof et al., 2007).

It is finally important to consider the perceptual capabilities that should be provided to a humanoid robot. The main constraint imposed is that these perceptual systems have to be mounted on the robot. Thus, they should be light, efficient and robust. They should also provide the robot with the ability to perceive and interact with its human partners. Thus, the WABOT-1 was provided with a vision system and a conversational system. These two sensory inputs (audio and video) are mounted in nearly all humanoid robots since then (Metta et al., 2000; Breazeal et al., 2004; Asfour et al., 2006b). In order to provide humanoid robots with the ability to perform 3D perception, their vision system usually consists of a pair of stereo cameras (Breazeal et al., 2001; Kojo et al., 2006; Azad et al., 2007b; Bandera et al., 2007). The microphones used to perceive sound are also usually configured to detect the sound sources (Breazeal, 2002). More recently, tactile sensors that were originally used only in certain body parts (i.e. fingertips) have evolved towards the creation of complete sensitive skins (Kerpa et al., 2003) that may provide the humanoid robot with more complete and precise tactile sensations (Asfour et al., 2006b).

The kinematics structure of a humanoid robot, their manipulation skills and their perceptual capabilities should be adapted to the environments in which it is going to be used. As commented above, these robots are designed to work in human environments. Thus, it is understandable that some characteristics of a humanoid robot resemble human ones. In fact, human environments are created by humans, for humans. It is worthy in the design of a humanoid robot, then, to follow not only engineering requirements, but also to study and analyze human characteristics (Metta et al., 2000). The design of learning mechanisms for humanoid robots can

be currently considered an interdisciplinary approach, influenced by studies about the learning mechanisms used by people. These studies are extracted from different research fields such as neuroscience and social sciences. For instance, mirror neurons (Rizzolatti et al., 1996) inspired the architecture of Demiris and Hayes (2002), that is detailed in section 2.4. Imitation and social learning mechanisms found in biological beings have also inspired authors in the robotics research field. Thus, the contribution of (Meltzoff and Moore, 1989), who appealed for an unified representation of perception and action in humans, is the basis of the visuo-motor mapping system proposed by Lopes and Santos-Victor (2005) for social robots. On the other hand, as pointed out by Mosterín (2005), the imitation process is just one of the possible social learning scenarios. Translated to robotics, teaching scenarios offer the human user the possibility to supervise robot learning, and evaluate the performance of imitated movements (Breazeal, 2002; Calinon, 2007). 'Focalization', as described by Thorpe (1963), is employed by authors as Scasselatti (1999), or Breazeal et al. (2005), to drive the attention of the robot using pointing gestures and gazing. The influence of social learning in social robotics has been revised in section 2.2, and will be further discussed in section 2.4. To summarize, the rise of humanoid robots moved PbD from a purely engineering perspective to an interdisciplinary approach. In fact, the term 'Program by Demonstration' is being progressively replaced in the robotics community by the biologically inspired term 'Learning by Imitation' (Billard and Dillmann, 2006). In order to avoid confusion with the biological process, in this thesis these robotic learning processes are referred as Robot Learning by Imitation (RLbI). Different control architectures have been proposed in the last years to implement RLbI systems in humanoid robots in particular, or social robots in general. Next sections introduce, describe and discuss about these architectures, and detail the one presented in this thesis.

## 2.4 Robot Learning by Imitation

There have been many proposals of RLbI systems in the last decade, due to the increasing interest in autonomous agents, humanoid robots and social robotics. In the previous section the evolution towards these RLbI systems has been described. In this one different architectures, that have inspired the approach proposed in this thesis, are detailed.

In order to achieve RLbI, most authors agree that it is necessary to describe the system as a set of modules, or elements, that address concrete tasks and that are connected between them, conforming a certain architecture. These architectures differ in their objectives, considered perceptual inputs, number of modules, levels of abstraction or structure. However, it is possible

to identify some key components that are common to most of these architectures. Thus, it is in the elements inside each of these components, and the relations that are established between these elements, where the differences between RLbI architectures lay.

While other descriptions of a RLbI architecture may be possible, this thesis considers a set of key components, that are related to the processes of cognitive imitation detailed in section 2.2. The components, depicted in Fig. 2.1, are described below.

- **Input.** This component includes all the sensory inputs that are available for the architecture. While visual input is necessary to achieve imitation in RLbI scenarios, some authors propose the use of additional perceptual channels, such as auditive or proprioceptive -own state- perception.
- **Perception.** The perception component contains all modules that are used to extract useful information from available perceptual channels. Raw data provided by sensors are useless for the robot. Thus, the audio information should be filtered to select only important messages. It is also necessary to extract only the relevant information from the huge amount of visual input data. Focusing attention is necessary to offer *on-line* response for the artificial agent. It is also convenient to ease social interaction and provide the social robot with human-like features (e.g. people interacting with the robot appreciate that it focus on the interaction process, instead of distracting itself processing continuously every signal or object).
- **Knowledge.** People use their memories continuously in their daily life. For instance, facial expressions or social gestures are necessary to achieve social interaction. Thus, a social robot should know how to identify and execute them. The knowledge component represents the memory of the social robot. This component contains all elements that are used to store information units, both learnt (memes) or preprogrammed (genes). It also includes elements used to organize these data, reduce their dimensionality or transform them to a format that can be useful for other components.
- **Learning.** RLbI scenarios involve real environments and users. It may be very difficult to preprogram every single situation a social robot can face in such scenarios. Instead, as commented before, a more reasonable strategy is to provide the robot with some learning mechanisms that allow it to adapt and learn from these new situations. The learning component is a key component of a RLbI architecture. It mainly affects the knowledge component, adding new items to the knowledge database, but also modifying already stored items or deleting old ones.

- **Motion generation.** RLbI requires the social robot not only to perceive, recognize or learn human behaviours. The robot has also to be able to imitate these behaviours. Imitation involves translating the perceived or learnt motion to the robot, and generating a sequence of motion commands. It is interesting to consider that this motion generation component contains all elements that are responsible of generating a motion output in the robot. Thus, not only gestures but also expressions and speech commands can be generated in this component.

It may be argued that speech can be generated by a sound synthesizer that requires no motor to be moved, thus it should be included in a different component, or the name of this component should be modified. However, a sound synthesizer requires to generate a signal that effectively moves a physical system (a speaker). Besides, there are other strategies to generate sound, i.e. people use mouth and tongue movements to do it. Thus, the name of this component has been maintained, as it is responsible of generate a set of motion commands.

- **Output.** Motion commands are received by this component of the RLbI architecture, that uses the abilities of the social robot to execute them.



Figure 2.1: Different components that can be identified in a RLbI system.

The RLbI architectures that have inspired the proposal presented in this thesis, and other contemporary architectures that may be interesting to compare with, are detailed below.

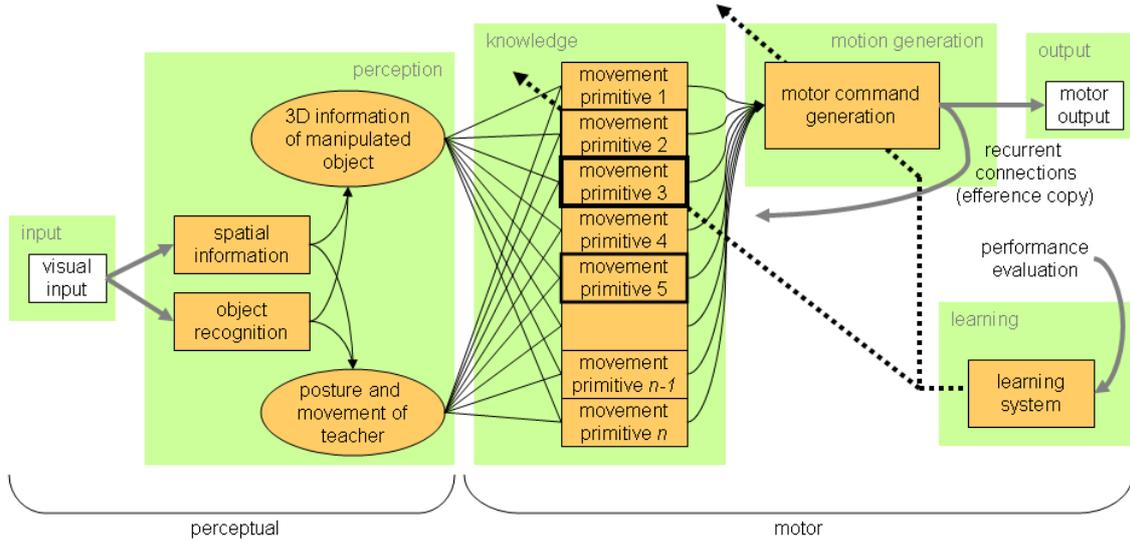


Figure 2.2: Conceptual sketch of an imitation learning system (Schaal, 1999).

### 2.4.1 Schaal’s proposal

One decade ago, S. Schaal proposed to address the problem of imitation learning in humanoid robots from a pragmatic point of view (Schaal, 1999). Based on concepts extracted from neuroscience and biological learning processes, the author proposed a conceptual sketch of a RLbI architecture that has inspired many further works and has been largely referenced in the subsequent literature. The original architecture itself is still commonly used as the basis of very recent contributions in this research area (Pastor et al., 2009). The proposal presented in this thesis is also strongly related to this architecture, as will be discussed below.

Fig. 2.2 depicts the RLbI system detailed in Schaal (1999). The architecture is divided into two major parts: perceptual and motor. The first of these parts contains the input and perception components described above. It can be seen that the visual input is immediately parsed, after capture, into (i) information about objects; and (ii) their spatial location in an internal or external coordinate system. As detailed in a posterior contribution of the author (Schaal et al., 2003), this organization is largely inspired by the dorsal (what) and ventral (where) streams, discovered in neuroscience research. From these two data sources it is possible to extract 3D information about objects, and also the posture and movements of a human teacher.

Once the perceptual part of the process is completed, a key problem arises about how

perceived information can be translated to useful robot actions (Schaal et al., 2003). As depicted in Fig. 2.2, movement primitives are used to help in this translation. Movement primitives are the basic units stored in the knowledge component of this architecture. Schaal (1999) defines these movement primitives as sequences of actions that allow to accomplish a goal-directed behaviour. These sequences could be simple or complex depending on the implementation, but in any case they should allow generalization. The author suggests to use them to encode complete temporal behaviours instead of low-level basic motion units, that may find difficulties in being scaled to systems with many degrees of freedom. The non-linear differential equations used by Pastor et al. (2009) are a good example of how these movement primitives can be obtained.

As depicted in Fig. 2.2, the movement primitives are defined in the motor side of the architecture. Thus, they are encoded as sequences of robot actions. On the other hand, the system perceives human movements. As the bodies of the human and the robot may be very different, it is necessary to solve the *correspondence problem*, that is defined as the problem of translating, or retarget, perceived human movements to robot motor motion (Nehaniv and Dautenhahn, 2002; Alissandrakis et al., 2007). This translation is not included in Schaal’s architecture. It could be implemented as an intermediate element between the ‘posture and movements of the teacher’ element and the movement primitives defined as robot actions.

Once the movement primitives are available, the system should be able to match perceived data against these primitives and select the ones that better satisfy this matching process. The learning component of the architecture affects the knowledge database by adding new primitives if no existing primitive is a good match for the observed behaviour. It also enables self-improvement as stored primitives may update their contents depending on their differences to the observed behaviour, the matching results and the evaluation of performance obtained after imitation is executed. As Fig. 2.2 depicts, the learning system also affects the motion generation component, as motor commands can be refined depending on the performance evaluation.

#### 2.4.1.1 Implementation

The architecture proposed by Schaal (1999) represents an useful conceptual scheme, elaborated using insights from different disciplines. It includes the main components that can be differentiated in most RLbI architecture, and has inspired many further contributions, and the current thesis.

The paper of Schaal (1999) does not pretend to expose a practical implementation of a RLbI system. In fact, as the author states, this paper address RLbI from the motor end,

assuming all necessary perceptual information is available. Thus, this contribution does not deal with problems derived from the practical implementation. It also does not consider, not even from a theoretical point of view, the issues that appear in RLbI regarding perception. Finally, as detailed above, it does not address the problem of translating the motion from human to robot.

Further works that use, or are inspired by, this architecture have contributed with practical implementations of the knowledge and motion generation components (Calinon, 2007; Pastor et al., 2009). But these approaches still do not deal with the problems derived from limited perceptual capabilities. Thus, Calinon (2007) uses a motion capture suit or directly moves the robot motors, set to passive mode, to obtain accurate and reliable inputs for the knowledge component. Pastor et al. (2009) presents a limited scenario in which a robot is taught one-arm movements. The characteristics and goals for these movements are known *a priori*. A manually operated robotic arm, that is provided with 10 Degrees of Freedom (DOFs), is used to record motion at 480 samples per second. These contributions do not consider the inclusion of a re-targeting module, as in these particular scenarios the input data are forced to be very similar to the output motor commands. No experiments are performed in these contributions that deal with more generic RLbI scenarios or limited perceptual cues.

#### 2.4.2 Active imitation. The biologically-plausible approach of Demiris and Hayes

Demiris and Hayes (2002) remind that the base of the ability to imitate is a mechanism that matches perceived external behaviours with equivalent internal behaviours. This mechanism uses information extracted from perceptual, motor and memory systems. If this mechanism fails or has some kind of malfunction in a certain person, it produces pathological disorders such as autism or some forms of apraxia (Demiris and Hayes, 2002). Problems or inability to imitate are in fact used as detectors of these disorders in people.

The RLbI architecture developed by Demiris and Hayes (2002) is useful for a social robot to acquire new behaviours. But the previous facts moved the authors to propose a biologically-inspired architecture to be used as a tool for further experiments, not only in robotics but also in other research fields.

Demiris and Hayes (2002) establish a main division between passive imitation and active imitation. These two concepts represent two very different approaches to imitation, but they are used jointly, providing an architecture that correlates better to human imitation than the

ones that are based only in one of the approaches.

- **Passive imitation.** This approach to imitation goes through a 'perceive - recognise - reproduce' cycle, where the motor systems of the imitator -the robot- are only used in the last step of the process. This form of imitation is a powerful tool to ease robot training and learning (Kaiser and Dillmann, 1996). It presents two characteristics that, according to Demiris and Hayes (2002), may limit it: (i) there is no substantial interaction between the three stages of the process; and (ii) the motor system of the robot is only involved in the last part of the process.
- **Active imitation.** Instead of going through the previous, passive cycle, active imitation involves the motor system in the process from the perceptual stage. Thus, at the same time it is perceiving the demonstration, the robot internally generates possible behaviours in parallel, selected among the ones that most probably match the observations. Predictions about the next state of the performer are computed from these internal representations of behaviours using forward models. The comparison between predictions and perception provides an evaluation value that is used to select the matching behaviour. When compared against passive imitation, the active approach produces faster results, adapts better to partial demonstrations and also matches better to the characteristics exhibited in adult human imitation (Calinon, 2007). On the other hand, its main disadvantage is that it is not capable of imitating behaviours that are not in the knowledge database.

Demiris and Hayes (2002) propose to combine both approaches in the RLBI architecture depicted in Fig. 2.3. It can be seen how the knowledge database is composed by 'behaviours', very similar to the 'movement primitives' described by Schaal (1999), and how predictions are used to select the correct stored behaviours according to comparison against the perceived state of the performer. A learning component is added to allow including new behaviours in the repertoire. This learning component executes passive imitation on the sequence of perceived states, and adds this information to the knowledge component, as a new behaviour, if the active comparisons do not produce a positive result.

Fig. 2.3 shows that the behaviours are represented in the motion space of the robot, and thus motor commands can be directly extracted from them. On the other hand, as in Schaal (1999), the perception stage explicitly includes only the elements that are necessary to extract performed pose from input images. But it would be necessary to include also a retargeting element that extracts a certain robot pose from this perceived human pose. This element is

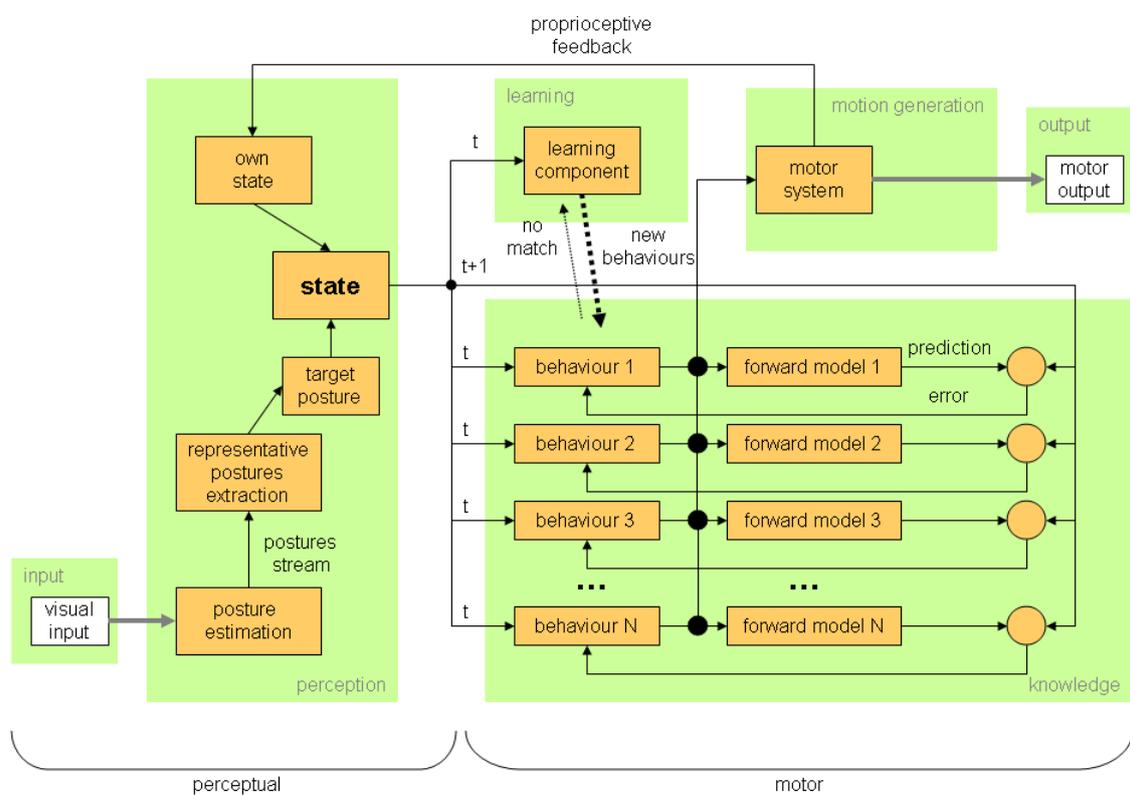


Figure 2.3: Biologically-plausible model for a learning by imitation system, proposed by Demiris and Hayes (2002).

implicit in these architectures, although it can be considered only a trivial step if the bodies of the human and the robot are very similar.

As depicted, in the architecture of Demiris and Hayes (2002), the resulting robot pose is combined with proprioceptive information to conform the current state of the imitator. This state is finally fed to the knowledge component and the learning component.

#### 2.4.2.1 Implementation

Demiris and Hayes proposes a RLbI architecture that is closely linked to concepts taken from neuroscience and social learning. The use of a dual-route structure for imitation matches with behaviours, pathologies and observations obtained from biological entities. The architecture proves to be able to imitate and learn unknown, partially known and fully known sequences of movements.

The experiments presented in Demiris and Hayes (2002) are executed over a virtual robot equipped with thirteen DOFs in its upper-body. All performed experiments involve only simulated data. The authors do not consider the problems derived from limited perceptual inputs, and they do not address the retargeting problem, assuming the *representative postures* extracted from perceived data can be directly mapped into the robot motion space. In order to emulate errors in the vision and the proprioceptive systems, an uniformly distributed random noise is added to the generated inputs. But this strategy does not consider the practical errors that arise in RLbI scenarios regarding occlusions or partial data. Finally, the recognition process is able to perform some temporal normalization to adapt to movements performed at different speeds. However, this algorithm is not able to deal with local temporal shifts. Thus, a certain motion is required to maintain its relative velocities at all moments in order to be recognized.

#### 2.4.3 System architecture for a social robot. The contribution of Breazeal et al.

While the previously detailed architectures are exclusively oriented to motion learning by imitation, the system described here presents a complete cognitive architecture for a social robot. This architecture, proposed by Breazeal et al. (2004), differs from the previous ones in that it models the complete functionality of a social robot. Thus, it includes additional elements and functionalities. The main characteristic of this architecture is that it is designed for a robot that cooperates with humans. As stated by Breazeal et al. (2004), it is important in this context that the social robot is able to interact with humans, and learn quickly and efficiently from

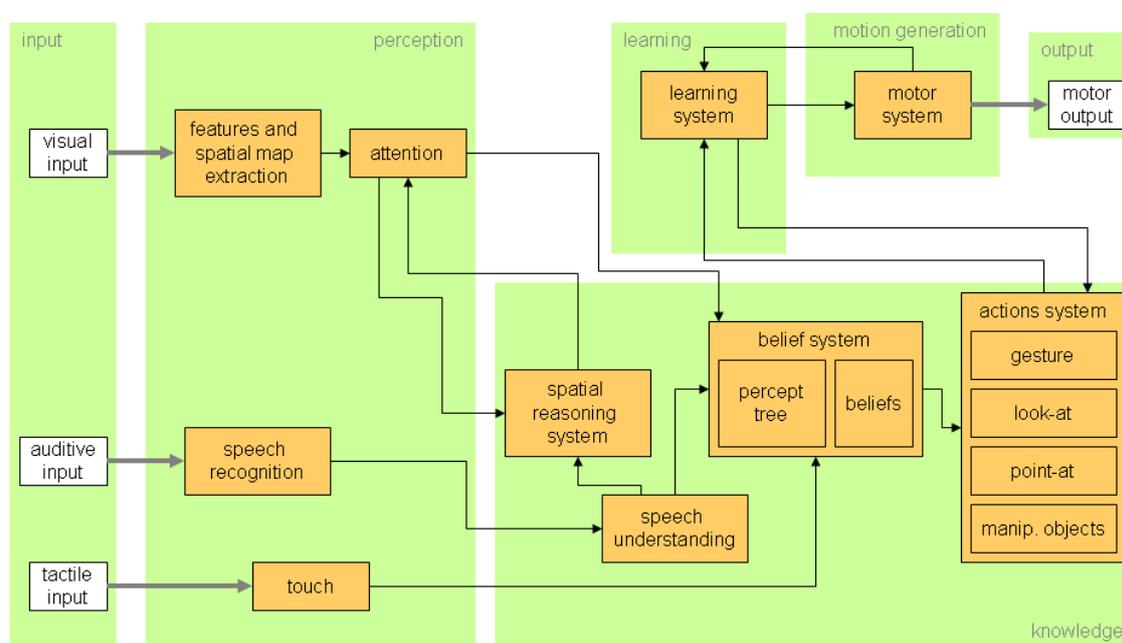


Figure 2.4: Social robot cognitive architecture for learning and performing tasks and motor skills (Breazeal et al., 2004).

natural human instructions. The robot should also be intuitive and engaging to communicate and interact with. It is also important that it is able to detect where the human is driven his/her attention to. This allows sharing human and robot focuses of attention, a key requirement in cooperative scenarios.

In order to provide natural communication channels from the human to the robot, as depicted in Fig. 2.4, the input component includes not only visual perception, but also auditive and tactile stimulus. The knowledge database is also more complex than in the previous approaches, including not only the actions of previous architectures, but also a 'belief system', that provides a higher-level knowledge about the environment and perceived stimulus. Thus, while *seeing* reflects the act of directly perceiving the world around, *beliefs* are representations that do not depend on direct, immediate observations. For instance, if the robot sees someone entering the toilet, it should *believe* that this person will go out after some time. While beliefs are very related to perception, they imply a certain level of previous knowledge, thus this element has been included in the knowledge component.

One of the main elements of the RLbI architecture proposed by Breazeal et al. (2004) is the set of 'percepts', or 'perceptual units'. Percepts may correspond to multiple sensed data:

vision, tactile information, audio and proprioceptive feedback. Sensory inputs that are detected by the perception system activate a hierarchical structure called the 'percept tree' that classifies this sensory information. The belief system, then, uses this output to generate new beliefs or update existing ones.

[Breazeal et al. \(2004\)](#) explain that the actions stored in the knowledge component are motion sequences performed by a human demonstrator. This demonstrator wears a motion capture suit able to measure joint angles at over 40 different points in the upper-body of the human, using potentiometers and gyros, at more than 120 frames per second. Each of the actions in the database has some degree of generalization thus it can adapt to different situations, e.g. in the particular scenario described in [Breazeal et al. \(2004\)](#), it is possible to perform a 'push-button' motion even if the 3D position of the particular button changes, as the task is described according to the changes it produces in the environment. As depicted in Fig. 2.4, a learning component is used to include new actions in the database or improve already known actions by evaluating the performance of the executed motion. Feedback is provided from the motor system to the learning system. This allows to perform the evaluation process, providing information about the robot inner state.

Finally, it is interesting to remark that the relations between different components of this architecture are slightly more complex than in the previous ones. For instance, the attention element in the perception component helps detecting the human focus of attention, computed by the spatial reasoning element in the knowledge component. But this last element also influences where the robot should drive its attention to.

### 2.4.3.1 Implementation

[Breazeal et al. \(2004\)](#) implements their proposed RLbI architecture in a social robot, Leonardo. This robot has 65 DOFs and is equipped with a stereo vision system, tactile sensors, speech recognition and generation systems, and an expressive face able to provide natural and intuitive feedback to the human user.

The experiments performed in [Breazeal et al. \(2004\)](#) demonstrate the ability of the robot to interact with people, using both visual and auditive perceptual channels. However, as commented above, the learning process relies on the use of a motion capture suit that has to be worn by the human demonstrator, thus visual input is effectively not used for learning purposes.

Generalization of learnt skills is demonstrated in constrained scenarios. Performed tests

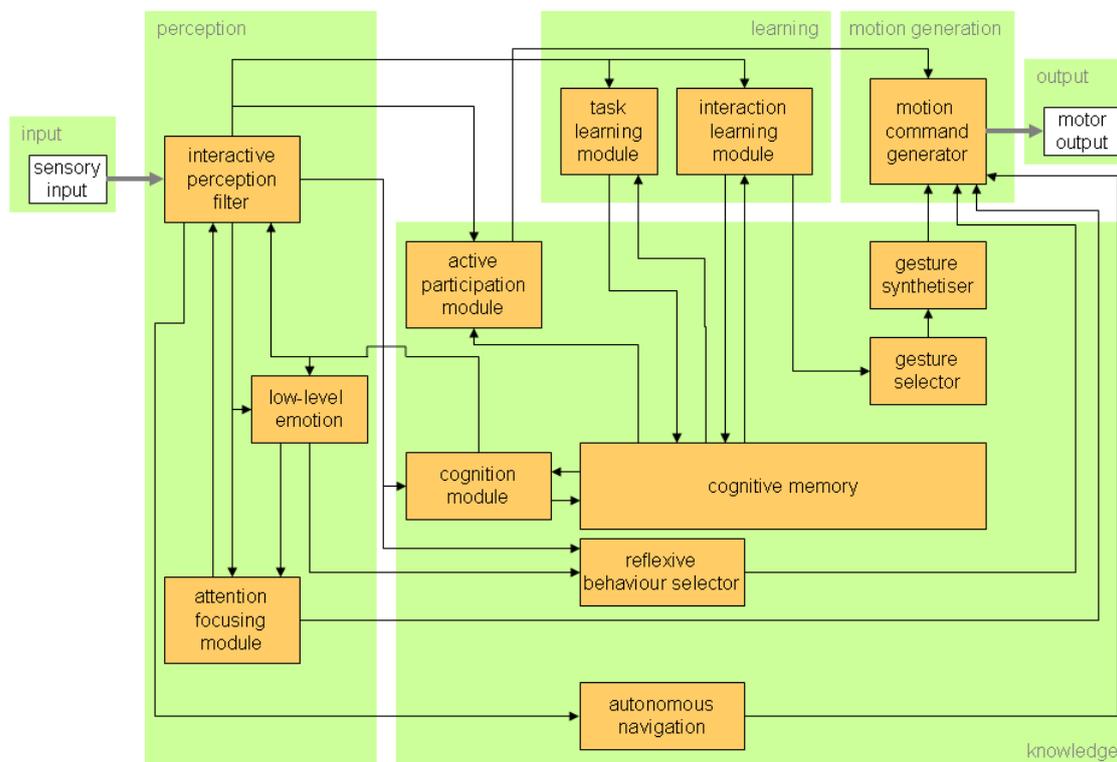


Figure 2.5: System architecture for a social robot presented in Mohammad and Nishida (2009).

also show that the robot is able to identify certain predefined objects, and to share its focus of attention with the human user. In summary, the RLbI architecture presented in Breazeal et al. (2004) represents an important step towards the creation of robots that can cooperate with humans in social terms. The current implementation of this architecture proves that it is able to provide successful results in concrete scenarios, where it is possible to use accurate motion data to achieve learning, and performed tasks conform a constrained repertoire.

#### 2.4.4 Reinforcing intended behaviours against unintended ones. The social robot architecture of Mohammad and Nishida

The architecture proposed by Mohammad and Nishida (2009) for a social robot is depicted in Fig. 2.5. The main element of this architecture is the 'interactive perception filter'. It can be seen that this element is highly coupled with nearly all remaining elements. It is the first and main processing unit of the social robot, and the mediator between the agent and the human user.

The most important function of the interactive perception filter is to translate perceived information to the high-level reasoning modules of the robot. While this translation may be easy when dealing with passive objects, static or not, the problem becomes more difficult when considering intelligent agents (humans or other robots). In this case, the authors argue that it is not enough to perceive and encode sequences of perceived states, but if true RLBI has to be achieved, it is necessary for the robot to *understand* the goals, or intentions behind perceived actions.

The usual strategy is to leave the goal-directed behaviours to higher-level reasoning systems. However, [Mohammad and Nishida \(2009\)](#) state that this approach can complicate too much the higher-level layers. They remind that goal-directed learning has to deal with noise attenuation, enforcement of intended behaviour respect to unintended ones, context analysis, interaction atmosphere detection and alignment with other agents. The authors propose to deal with these problems in the perception component. They state that this especially eases perceiving intended behaviours.

As Fig. 2.5 depicts, this architecture divides each component into more elements than the previously presented architectures. It is, in general terms, more complex in elements and relations. From the point of view of this thesis it is interesting to consider how gestures are generated here. It can be seen in Fig. 2.5 that the 'interaction learning module' is the responsible of matching perceived and stored gestures. From this comparison a certain gesture is selected. Then, the selected gesture is synthesized using previous knowledge about the robot kinematics and the result is fed to the motion command generation. This element also receives more inputs to deal with reflexive behaviours, autonomous navigation or social requirements, thus it becomes a more complex and versatile element than the motion generation components presented in previous architectures.

#### 2.4.4.1 Implementation

[Mohammad and Nishida \(2009\)](#) test their interactive perception filter component using different experiments that involve not only robotics but also human-computer interaction. For this thesis, only the experiments performed in the field of social robotics are considered.

The authors use a simulated robot that tries to mimic performed one-hand movements. These movements are perceived using a position sensor attached to the human finger. In order to model noisy perception, a white noise component with a magnitude of 20% is mixed with the original signal in two ways, to model a nonlinear influence: a portion of the noise component

is multiplied, the other is added. As in [Demiris and Hayes \(2002\)](#), occlusions and partial data are not considered. The virtual robot can provide interactive feedback to the human using the screen of a computer as interface. As the authors state, these experiments are related only to the interactive part of the RLbI process. The rest of the module of the proposed architecture are not evaluated nor extensively detailed.

#### 2.4.5 The task-level imitation learning system of [Mülihg et al.](#)

The last architecture presented here is a simple task-level imitation learning system, proposed by [Mülihg et al. \(2009\)](#). They divide the architecture into three differentiated parts: observation, learning and reproduction. The observation part can be identified as the perception part of other architectures, while reproduction of tasks is what has been called 'motor' part in other architectures. Learning component is directly identified as the learning part. Knowledge component is not explicitly depicted in the structure presented in [Mülihg et al. \(2009\)](#). However, it is implicitly understood in the paper that both tasks and the gestures that compose them are learnt and stored in a certain knowledge database.

[Fig. 2.6](#) depicts the resulting RLbI architecture that matches the proposal of [Mülihg et al. \(2009\)](#). It is interesting to see how perceived data is mapped into the task space, thus the perception of the social robot is constrained, or polarized, by the concrete task being learnt or recognized. Another important characteristic of this architecture is that the task representation does not depend on temporal variations. The authors state that temporal normalization is crucial, as even the same person will perform multiple demonstrations of the same task at different speeds. While the use of a left-right Hidden Markov Model (HMM) is mentioned as an option, the authors prefer Dynamic Time Warping (DTW) to achieve this temporal normalization, as it does not introduce additional undesired smoothing effects in the perceived trajectories.

The motion generation component of this architecture, finally, shows a slightly different approach to the problem, respect to the previous architectures. In this case the motion is generated using not motor commands or kinematics, but dynamics. Thus, the executed task sets attractors for the robot end-effectors or relevant body locations. The movements of the agent tend to reach these attractors in an optimization procedure.

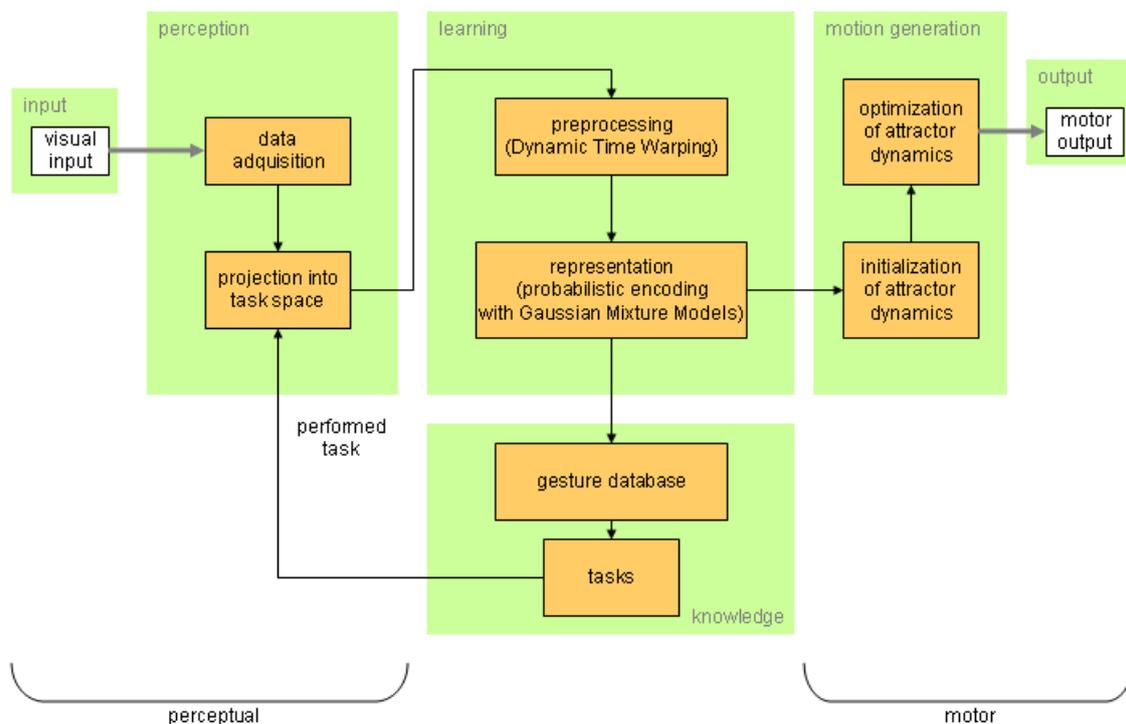


Figure 2.6: Architecture for task-level imitation learning proposed by Mühlhig et al. (2009).

### 2.4.5.1 Implementation

The architecture proposed by Mühlhig et al. (2009) has been tested using a real social robot, the ASIMO from Honda. For the experiments presented in the paper, only one specific task is considered: the *pouring* task, in which the robot has to pour liquid from one jar that it grabs in its right hand, to a glass grabbed by the left hand. Both glass and jar have specific colours. The robot focus attention on the position and orientation of both objects. These data are captured using only the visual perception system mounted on ASIMO.

The results of the experiments show that this RLbI architecture is able to learn trajectory-defined tasks. It also achieves interesting generalization levels due to the use of Gaussian Mixture Models and DTW, and an IK system able to consider and avoid incorrect poses. As the author conclude, this contribution is a valuable starting point to research in imitation learning. Learning different, not previously defined tasks, meet the conditions imposed in real RLbI scenarios (dynamic environments, untrained users, etc.) or achieve spontaneous, non-guided interactions are some of the issues the authors consider to address in further work.

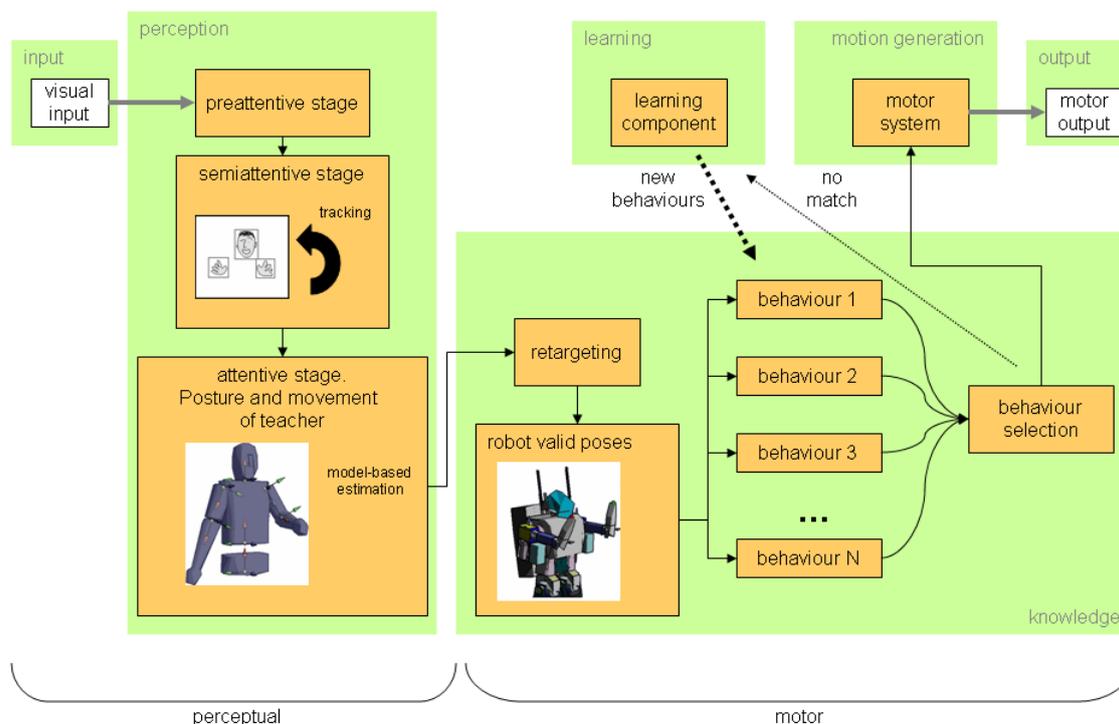


Figure 2.7: Architecture of the RLbI system proposed in Bandera et al. (2007).

## 2.5 First approach to the considered RLbI system. Discussion

Inspired by some of the previously detailed architectures, and taking into account some of the described concepts about social learning and RLbI, a first RLbI architecture for a social robot was implemented for this thesis (Bandera et al., 2007). The main objective set for this architecture was the integration of a system that could go through all the processes involved in RLbI, from perception to action. Thus, it could serve as a prototype which advantages could be emphasized, and which drawbacks could be corrected, in successive revisions.

The architecture is focused in gesture perception, representation and recognition, thus it is constrained to visual input, following the proposals of (Schaal, 1999) or (Demiris and Hayes, 2002). The design of this early architecture was driven by the requirements of RLbI scenarios. Thus, the human wears no specific markers nor color patches, the robot uses only stereo vision to perceive its environment, and this environment itself is dynamic and unpredictable. Fig. 2.7 depicts this first proposed architecture.

As depicted, the perceptual inputs for the architecture are the images obtained by a pair of stereo cameras. These cameras are useful to model human perceptual capabilities, and provide a considerable amount of information about the environment. Most humanoid robots are equipped with such a perceptual system. Thus, they seem a reasonable perceptual input for the architecture. The main requisite, for the used stereo pair, should be to be able to perceive human gestures performed at typical social interaction distances, between 1.5 and 2 meters, depending on the culture and type of interaction. This means that the baseline of the stereo system should not be very small. Average human baseline, about 10-12 centimeters, may be an adequate lower limit for this parameter.

The images obtained from stereo cameras contain a huge amount of information, as commented before. It may be very difficult to achieve *on-line* response if all these data have to be processed. Biological entities filter perceived information by attention (Bandura, 1969). Mohammad and Nishida (2009) give a key role to this filtering process, that becomes the main element of their architecture. In this proposal, stereo data are also filtered in the perception component, but using a simpler attention mechanism, that is also more independent respect to the rest of elements of the architecture. The mechanism is composed by three stages that follow the model of Treisman and Gelade (1980). The first of these stages -the preattentive stage- is a task independent element that detects objects of interest in the images. Different criteria are used to select these objects: color, proximity to the user or contrast respect to near regions in the image. The preattentive stage is followed by a semiattentive stage, that tracks objects of interest *on-line*, and performs 'inhibition of return'. This mechanism avoids detecting tracked objects as new objects of interest in the preattentive stage. These two elements of the architecture can be compared to the 'features and spatial map extraction' module of Breazeal et al. (2004), or the 'spatial information' and 'object recognition' modules in the architecture of Schaal (1999).

The two previous stages are followed by an attentive stage in which an Human Motion Capture (HMC) behaviour is implemented. Other behaviours, such as landmark detection or face recognition, may also be implemented in this stage if adequate objects of interest are located and tracked in the previous stages. Similar elements to this stage can be found in the architectures proposed by (Schaal, 1999) or (Demiris and Hayes, 2002). The HMC behaviour implemented in this architecture uses a model-based method to extract human pose from face and hands perceived positions. These positions may be considered one of the main features in social gestures (Breazeal et al., 2003). They are obtained by considering the three largest perceived skin color regions. It is also necessary to impose that, in the first frames of the gesture

demonstration, the left hand is located anywhere in the left part of the body, and the right hand is located in the right part of the body. Once detected, the tracked positions of these three body items are feed to the attentive stage, that uses an IK algorithm, reinforced with an alternative pose search algorithm, to estimate human pose. These algorithms are explained in chapter 3.

Once the motion of the performer has been extracted, it is necessary to extract gesture data from this motion. In this proposal, the motion is first encoded, or retargeted, in the robot motion space. This approach follows the ideas of [Lopes and Santos-Victor \(2005\)](#), that propose a representation that unifies perception and action. The same concept can be implicitly found in the proposals of [Schaal \(1999\)](#) or [Demiris and Hayes \(2002\)](#), where the absence of a specific retargeting module implies a strong similarity between the human and the robot, that eases the use of a common representation encapsulating perceived human motion and executed robot motion. It is difficult, however, to find robots that meet this requirement. In practice, in the experiments performed to test this first proposal, there was a considerable limitation in the amount of gestures that could be reproduced by the used robot, due to the constrained reachable space of its arms ([Bandera et al., 2007](#)). Then, as both action and knowledge were performed in the robot motor space, this limitation in the reachable space was extended to the representation stage. Thus, hand movements that were correctly tracked could not be adequately described in the knowledge component of the architecture, simply because the kinematics system of the particular robot could not adopt those positions. This proposal limited the gesture repertoire that can identify the robot not by its perceptual capabilities, but by its motor abilities. In conclusion, the robot was not able to recognize the gestures that it was not able to imitate correctly. This is not a desirable situation for a social robot that, regardless its kinematics resemblance to people, should provide natural and intuitive interaction channels to human users.

In order to address the previous issue, it is important to consider that biological observers perform imitation and social learning from demonstrators of the same species ([Mosterín, 2005](#)), or who are perceived as belonging to the same species. [Meltzoff and Moore \(1989\)](#) demonstrated that small children may find difficulties in learning from imitation when the demonstrator was not a human teacher, but a machine. It may be reasonable to suppose that the opposite situation will find similar difficulties. Thus, using the robot motion space to represent perceived human movements is a strategy that should be restricted to situations in which the bodies of the human and the robot are similar enough. Otherwise gesture recognition can be constrained by the motor limitations of the robot, not by its perceptual abilities.

The following component of the architecture is the knowledge component. It includes the units of social knowledge, related to the memes defined by ([Dawkins, 1976](#)). These units

of knowledge receive different names in previous architectures: 'movement primitives' (Schaal, 1999), 'gestures' (Mülich et al., 2009) or 'behaviours' (Demiris and Hayes, 2002). As depicted in Fig. 2.7, this first approach followed the last nomenclature, although it may lead to confusion with the 'behaviours' implemented in the attentive stage.

The selection of the behaviour that matches perceived one is achieved using DTW, thus temporal influence is not considered as in (Mülich et al., 2009). If no stored behaviour is close enough to the perceived one, a learning component is invoked to update knowledge database. As depicted in Fig. 2.7, selected behaviours are directly translated to motor systems as they have been represented in the motion space of the robot.

Finally, it is important to consider that this proposal served as a starting point to the research presented in this thesis. It presents some issues that have been highlighted and should be corrected. This task is eased by the modularity of the architecture, that allows changing different elements of the architecture by updated or new modules without modifying the rest of the system. Besides, the relative small number of connections between elements allows to move, discard or divide elements more easily than in other more densely connected architectures (Mohammad and Nishida, 2009).

## 2.6 Proposed RLbI system

The RLbI architecture presented in this thesis is depicted in Fig. 2.8. It is based on the architecture described in the previous section, although it presents some important changes that are detailed in the description of each module, provided below.

- **Input.** This thesis presents a RLbI architecture that is strictly based on vision. Thus, the only input to the system are the images perceived by a pair of stereo cameras. This stereo pair is mounted in the head of the social robot and, as detailed in chapter 3, its baseline is set to an adequate value, close to human average eye-to-eye distance.
- **Perception.** The perception system is based on the three stages depicted in Fig. 2.7. The elements, however, are slightly different. Thus, the preattentive stage has been replaced by a more specific 'feature detection' module. Instead of extracting generic objects of interest from input images, the feature detection element is focused on the perception of the human performer, thus it firstly looks for a close human face in the perceived images. Once the face is detected, the person silhouette is obtained from disparity map, and the hands are located as skin color regions in certain parts of the silhouette. The tracking system executes

the same task than the semiattentive stage. Finally the last stage, that implements the HMC system itself, has been extended by considering torso pose estimation, not only arms pose extraction. Chapter 3 deeply explains this HMC element.

It may be argued that these modifications produce a less generic system. However, this system meets better the requirements of the proposed RLbI system. Besides, it can easily be extended to include new elements, that address different tasks or more complex perceptual abilities.

- **Knowledge and learning.** As detailed above, encoding the perceived gestures in the robot motion space may impose additional constraints to the set of gestures the robot can perceive. This limitation is more important as the differences between the robot and the human grow. The knowledge component of the proposed architecture uses instead the human motion space to encode perceived gestures. Thus, even if the robot body is very different to the human one, the robot is able to perceive, understand and learn human gestures. The base of this proposal can be expressed as 'use a human model to represent human perceived motion'. In this approach a human model is, in fact, replacing the robot itself, and assuming the role of the imitator in the inner knowledge representation of the robot, as detailed in chapter 3. While the subjects and details are of course different, it is possible to set a certain relation between this strategy and the method of Stanislavsky, widely used by theater actors (Stanislavsky, 1936).

The second important characteristic of the knowledge component is the existence of a direct imitation path from perception to action. This connection is inspired by the 'active imitation' approach detailed by (Demiris and Hayes, 2002). In this case, however, the motor commands are replaced by the joint angles of the virtual human model that lays inside the knowledge representation. The active imitation element allows to perform 'true imitation' (Thorpe, 1963) of new gestures, even if they are not present in the knowledge database, as they may be directly retargeted from the motion encoded in this module. This provides immediate feedback to the human user and, in general, eases learning processes as active imitation element becomes an useful interface for the learning component.

On the other hand, if a certain gesture is recognized, it is not necessary to use the contents of the 'active imitation' module. The robot can use instead its stored representation for that gesture. The approaches proposed in this thesis to represent gestures using reduced sets of dominant points, and to recognize gestures combining distances based in local and global features, are detailed in chapter 4.

Finally, Fig. 2.8 shows that memory units are simply called 'gestures' in this architec-

ture. Gestures are here understood as movements that express or help express thought or emphasize speech. As detailed in further chapters, in this thesis gestures are divided into static gestures and dynamic gestures. Static gestures are simple poses. Dynamic gestures are movements limited by two static poses. This classification is related to the experiments of [Smyth and Pendleton \(1990\)](#), and inspires the use of different retargeting algorithms.

- **Motion generation and output.** The retargeted motion is not directly feed to the robot motors. A virtual model of the social robot is used before to check that the resulting poses are valid. The robot model adopts desired poses using the same algorithms than the human model. Thus, it is able to look for alternative body configurations that lay inside the reachable workspace if required. Once a valid pose has been computed for the robot, it is finally sent to its motor system.

The last advantage of this RLbI architecture is that, being the retargeting element located after gesture recognition and learning systems, it does not need to be executed if the robot is not going to perform imitation. This increases the efficiency, and reduces the response time, of the proposed architecture.

## 2.7 Map of the thesis

Fig. 2.9 shows the RLbI architecture used in this thesis, and previously depicted in Fig. 2.8. Fig. 2.9 identifies the chapters of the thesis that describe each of the parts of the proposed architecture.

The feature extraction and tracking elements are presented in chapter 3. Some of the algorithms used to perform these tasks and, more precisely, the entire tracking algorithm, are not contributions of this thesis but previous works, and thus they are explained in the appendices.

Once features are extracted from the input stereo images, they are used to obtain the upper-body pose of the human performer. The model-based HMC system implemented in this thesis to achieve this translation, from image features to human pose, is detailed in chapter 3. On the other hand, it is important to consider that the same algorithms used to pose the human model, in the perception component, are also used to pose the virtual model of the social robot, in the motion generation component. This decision allowed to implement a generic model-based pose estimator that deals with kinematics structures that differ to a certain degree. But it would be complex to detail some of the characteristics of these algorithms without considering that they are used not only to track human motion, but to make a virtual robot imitate it. Thus,

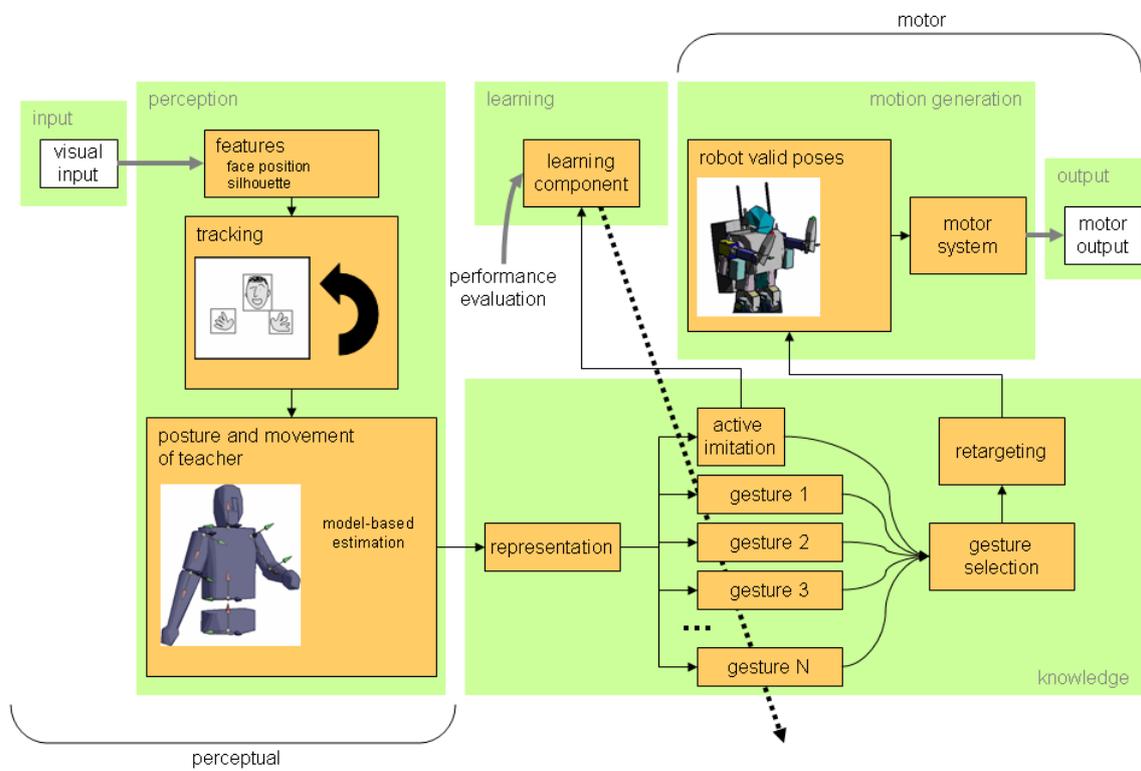


Figure 2.8: System architecture.

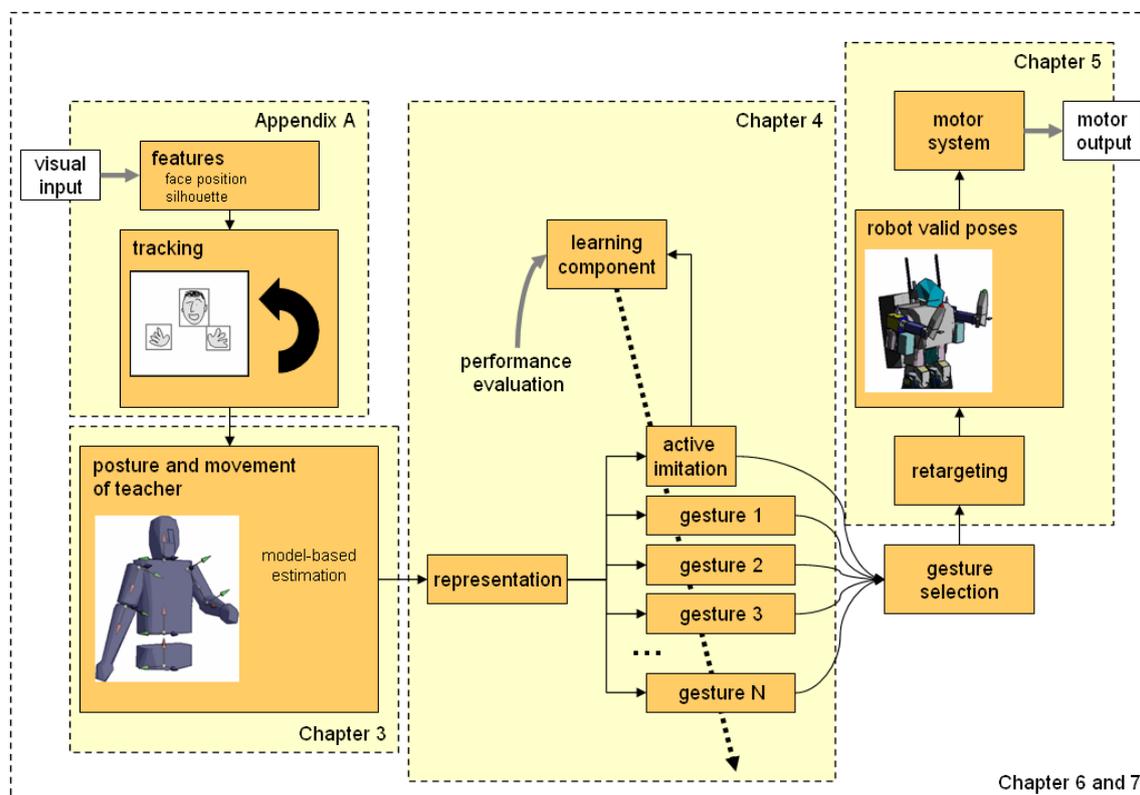


Figure 2.9: System architecture, showing the chapters that describe each part.

robot models are also used in chapter 3 to illustrate some of the steps of the pose estimator. This chapter ends with a quantitative evaluation of the HMC.

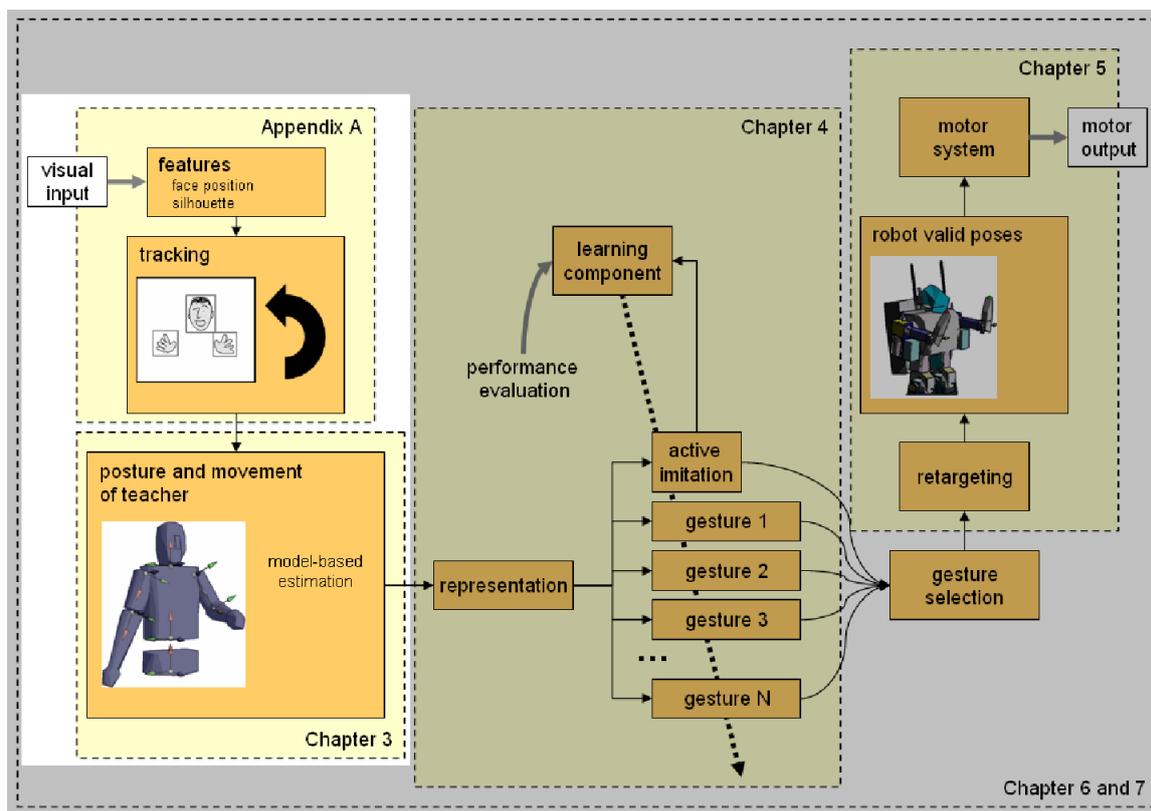
The next step in the proposed RLbI architecture is to segment the perceived motion in discrete gestures, that are encoded in an efficient representation. Then, gesture recognition and learning elements are executed. These processes achieved in the knowledge and learning components are described, and evaluated, in chapter 4. However, some final considerations about the learning system are presented in chapter 6.

The social robot does not only perceive and recognize gestures. It should also be able to imitate them. Imitation involves translation of the perceived gestures from human to robot motion space, a process that is deeply explained in chapter 5. Then, translated poses are checked to ensure they are safe and coherent for the robot, using the same pose generation algorithm employed in the HMC system.

Results obtained when the complete architecture, depicted in Fig. 2.8, is employed, are presented in chapter 6. Some final topics are discussed in chapter 7.

# Chapter 3

## Human motion perception



### 3.1 Outline of the chapter

It is necessary for the perception component of the proposed RLbI architecture to extract and track *on-line* important features from input images. However, an useful RLbI system requires additional perceptual cues that should be considered at a higher level perception stage. As detailed in chapter 2, in the proposed RLbI not only this higher level perception stage is oriented

to capture human motion, but there is also a certain degree of particularization in the previous steps of the perception component. As discussed there, this does not imply losing generalization capabilities, but focusing on the task to be performed.

In order to learn from a human demonstrator, it is necessary for the social robot to be able to detect human interaction partners. These partners should be tracked independently, regardless of environmental influences, such as additional people entering or remaining in the robot's field of view. Besides, while some surveillance applications only require the perceptual system to inform about the presence of a human, in RLBI scenarios the person should not only be detected, but his/her pose has to be extracted from the different features that are perceived by the stereo vision system. In this chapter, a novel HMC system that is able to meet these requirements is presented and evaluated. This system can be described as a perceptual stage that uses the information provided by the previous elements ('feature extraction' and 'tracking') to achieve higher perceptual capabilities. It has to deal with severe specifications, imposed by the RLBI scenarios defined in chapter 1. Some of these specifications are reminded below:

- *Non-invasiveness.* The human is not required to wear markers, color patches nor specific garments.
- *Untrained users.* The HMC system should be able to adapt to different human interaction partners. No specific training or initialization phase for each user are allowed. On the other hand, the users may not be familiar with the system.
- *Interaction with unpredicted users.* The system should impose only soft initialization constraints, if any.
- *Real environments.* Occlusions, variable light conditions, distracting objects, multiple people in the field of view or dynamic changes in the scenario should be considered. While both indoor and outdoor environments are complex, the last ones are usually much more challenging, dangerous and unpredictable. As commented in chapter 1, in this thesis we are considering indoor scenarios only.
- *Limited a priori knowledge.* The social robot should be able to learn a wide variety of social gestures, that are not restricted nor known *a priori*.
- *On-line response.* The robot should work at human interaction rates.

Visual perception, besides, may offer incorrect or noisy data due to calibration errors, lens distortion or disparity limited resolution. Thus, these data should be controlled and filtered

to avoid the social robot trying to imitate incorrect movements. In order to help in this process, the proposed vision-based HMC system employs an internal model of the perceived human that avoids incorrect poses. Model-based approaches are also more suitable to meet the previously mentioned fast response times (Agarwal and Triggs, 2006), specially when they are not based on probabilistic solutions. The proposed approach uses an analytic IK algorithm instead. This allows to imitate perceived motion *on-line*. The proposed HMC system is designed to capture upper-body motion, that composes the vast majority of social gestures. Thus, the model includes only the upper part of the human body.

The chapter is divided in the following sections:

- Section 3.2 firstly describes different HMC systems that are used in RLbI scenarios. Then, the proposed HMC system is introduced and compared against these approaches.
- The first step to capture the motion of a person is to locate him/her. Section 3.3 details how the proposed system detects the face of a human that is standing in front of the social robot. Once located, the face 3D position is used as a reference to extract the person silhouette from the perceived disparity image. This process is detailed in section 3.4.
- Section 3.5 describes the HMC system. The section starts detailing the geometric model that is used to represent the perceived human. Then, the algorithm that flexes and rotates the torso of the human model to match the perceived one is described. The hands motion is one of the main features in social gestures. The main part of the HMC system generates an estimation of the pose of the arms. This algorithm firstly uses an IK algorithm to produce a set of arm joint angles from the desired hand -or end-effector- 3D position. The subsequently pose is analyzed in order to check whether it is a valid pose or not. Invalid poses are avoided and replaced by valid alternative poses if possible. If not, a motion is generated in any case that tries to minimize the distance from perceived motion to imitated motion.

It is important to consider in this section that the algorithms used to pose human model, in the HMC process, are also used to pose robot model after perceived motion has been translated to robot motion space. Thus, these algorithms have been designed to deal with different models, being this generic approach responsible of some of their characteristics, as it is explained in section 3.5.

- The following sections focus on the evaluation of the HMC system. Before, it is necessary to detail the stereo vision system that has been employed to execute the experiments.

Section 3.6 describes the stereoscopic vision system proposed to capture upper-body human movements.

- Section 3.7 details the experiments used to evaluate the vision-based HMC system. Different options have been considered to perform this evaluation process. It has been finally decided to capture the same motion using both the evaluated system and a HMC system based on optical markers, and detailed in this section. The pose data provided by this system is used as a ground-truth against which the results of the evaluated vision-based HMC system are compared.
- Due to the specific conditions in which these tests had to be performed, data obtained from the marker-based system had to be processed in order to obtain valid ground-truth. Thus, section 3.7.2 describes how ground-truth is extracted from captured data, and then compares the results provided by the vision-based HMC system against this ground-truth.
- Finally, section 3.8 concludes the chapter discussing the results of the evaluation process, and suggesting directions of research that could be addressed if more precise motion capture was required.

## 3.2 Overview of the human motion capture system

As commented before, in this work it is assumed that imitation and learning by imitation is achieved by the robot itself, i.e. without employing external sensors or marks. Thus, invasive items are not used to obtain information about the demonstrator's behaviour. The proposed approach is exclusively based on the information obtained from the stereo vision system of the robot imitator. Thus, it is related to other experiments, e.g. the mimicking experiments shown by [Sauser and Billard \(2005\)](#) or the RLbI system proposed by [Asfour et al. \(2006a\)](#). However, in this thesis no external color marks nor specific garments are employed.

Vision-based HMC systems include very different approaches to extract motion information from images. In the last decade some authors have proposed different classifications of these methods. One of the classic references for this research field is the survey of [Gavrila \(1999\)](#), that classifies the vision-based HMC systems in: i) 3D approaches; ii) 2D approaches with explicit shape models; and iii) 2D approaches without explicit shape models. There is a certain overlap between these classes, as the author reminds. As all 3D approaches mentioned by [Gavrila \(1999\)](#) use models, this classification implies an effective differentiation between HMC methods that use models, and HMC methods that do not use them. This same classification is present in the

survey of [Aggarwal and Cai \(1999\)](#) who distinguish between model based and non-model based methods, depending on whether a priori shape model is used or not. This taxonomy is ambiguous as some described non-model based methods have some implicit information about the human body and construct a dynamic model using this information as a starting point ([Niyogi and Adelson, 1994](#)). This information could be considered as a model itself.

More recently, [Moeslund et al. \(2006\)](#) focus on describing the overall structure of a HMC system and its components, and do not try to classify methods in a global taxonomy. They follow the structure of a previous survey ([Moeslund and Granum, 2001](#)) and divide the HMC process into four steps: initialization, tracking, pose estimation and recognition. The different HMC systems are then analyzed considering how they resolve these different sub-problems. Thus, it is in the context of pose estimation where they include their analysis of human model usage in HMC systems, describing *model-free* approaches, approaches that benefit from an *indirect* use of a model and approaches that are based on a *direct* use of a model. [Agarwal and Triggs \(2006\)](#) adopt a very similar approach and divide the solutions to the problem of estimating and tracking the configuration of a complex, articulated object in two main categories: *learning-based* approaches, that rely on probabilistic and search methods to infer human pose from image cues, and *model-based* approaches that use a model of the perceived human to help in extracting his/her pose. This thesis adopts these classifications and divides HMC systems in model-free approaches and model-based approaches.

Model-free approaches directly map visual perception to pose space. This is a powerful tool that is useful to extract human pose from complex input data, such as cluttered natural scenes that are perceived from a single view. However, these approaches are time consuming due to the involved search and matching process. Their results are also restricted to the poses demonstrated in the training phase, and extension to wider vocabularies of poses may introduce ambiguities in the mapping. Finally, they also have to deal with the problem of local minima or silhouette ambiguities ([Mori and Malik, 2002](#); [Agarwal and Triggs, 2006](#)). Recent methods based on probabilistic assembly of body parts have achieved promising results in overcoming these limitations, but they are still bounded by training requirements ([Demirdjian et al., 2005](#)), and are in general more suitable to surveillance or people detection applications than for the *on-line* response requirements of RLBI scenarios ([Moeslund et al., 2006](#)).

Model-based approaches rely on the use of a human model to help in the human pose detection and tracking processes. Traditionally model-based approaches required controlled environments and/or visual marks located in the person body to achieve HMC ([Moeslund and Granum, 2001](#)). On the other hand, they are able to offer faster results as they do not need

to execute extensive search algorithms, they are more robust against ambiguities and incorrect poses as the model helps avoiding them, and they are not required to execute complex training phases as information about human kinematics and dynamics can be stored in the model itself (Safonova et al., 2003; Agarwal and Triggs, 2006; Moeslund et al., 2006). In the last years different approaches have appeared that benefit from these advantages while they are able to be used in real dynamic environments. Stereo vision systems have received a particular attention in the fields of Human Robot Interaction (HRI) and social robotics. While monocular perception have to use complex computation, such as probabilistic matching or optical flow to extract human 3D pose (Kulic et al., 2009), stereo vision-based systems benefit from disparity information to obtain 3D data from the scene. Thus, Hecht et al. (2009) propose the use of a flexible model, a short efficient training phase and several particle filters to track full-body human pose in real indoor scenarios at 10 frames per second, using stereo vision. Other approaches detect key body parts, such as face and hands, to infer human pose from limited and/or noisy perceived data and few or none initialization requirements (Kojo et al., 2006; Azad et al., 2007a; Fontmarty et al., 2007; Bandera et al., 2008b). These approaches are able to offer *on-line* response, but, as pointed out by Hecht et al. (2009), the use of a short stereo baseline provides only one perspective of the performed motion to these systems, that are, then, sensitive to occlusions. Besides, the positions of certain body parts, e.g. the elbows or shoulders, are not directly perceived, but have to be estimated, as no marks are located on the human body. This introduces an estimation error in the HMC process that may affect the RLbI process, and thus should be carefully considered.

This thesis proposes the use of a model-based Human Motion Imitation (HMI) system to infer the upper-body pose of a human performer and translate it to the robot. The performer wears no markers, color patches nor specific garments. The input data is constrained to color images and disparity information provided by a pair of stereo cameras which baseline is limited, as they are mounted on the head of a social robot. The HMC subsystem obtains the upper-body pose from key body parts (detected face and hands) positions, and estimated torso pose. Thus, it is affected by previously mentioned issues, i.e. occlusions and usage of estimations. As it is intended to be used in RLbI scenarios, noisy perceived data, interferences, dynamic lighting and variable structural conditions are also present. Different strategies that are deeply explained below are followed to minimize the effects of these issues. E.g., a human model is used to help dealing with noisy or incorrect data, perceived motion is filtered to recover from punctual hand occlusions, the system restarts the HMC subsystem once a face occlusion has been detected, disparity silhouette is processed to increase robustness of estimated torso pose, etc. A quantitative evaluation of the results achieved by the proposed HMC system is further provided in this chapter.

Fig. 3.1 shows the flow diagram of the HMC system implemented in this thesis. As depicted, it is composed by a set of modules, that are described below:

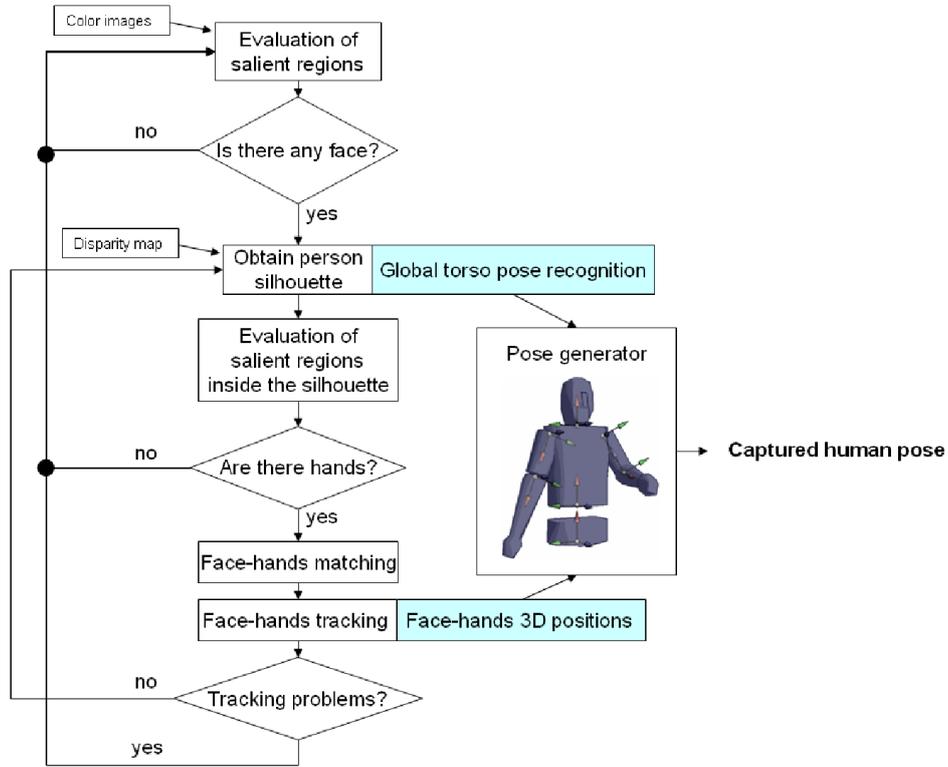


Figure 3.1: Flow diagram of the Human Motion Capture (HMC) system.

- Evaluation of salient regions. This module is more related to the feature extraction and tracking stages of the perception component, in which the HMC system is integrated. Fig. 3.1 shows how the HMC system firstly looks for a face in these regions, basically using the face detector proposed by [Viola and Jones \(2001\)](#).
- Obtain person silhouette. Once a face has been perceived, it is used as the starting point from which the whole silhouette of the performer is extracted. The process to obtain this silhouette is detailed in section 3.4, and it is based in the use of the stereo disparity map. The flexion and rotation angles of the torso are extracted from this silhouette using analytic relations based on anthropometry as described in section 3.5. The use of analytic methods instead of search or probabilistic algorithms allows for a fast response.
- Evaluation of salient regions inside the silhouette. After extracting the silhouette of the performer his/her head and hands are identified as the three biggest skin colour regions inside the silhouette. This strategy reduces noise and confusion produced by other skin

color regions, such as additional people or wood surfaces, appearing in the field of view of the cameras.

- **Face-hands tracking.** This process uses the Bounded Irregular Pyramid (BIP)-based tracking method detailed in appendix D to track the image regions corresponding to face and hands. The inverse of the projection matrix associated to the employed stereo system allows computing the 3D positions of the tracked body parts.
- **Pose generator.** Hands and face 3D positions, and torso orientation, do not define a pose. However, they may be used to compute a complete set of joint angles for the human upper-body. The proposed method relies on an IK algorithm to extract arm poses from end-effector positions. The analytic nature of this algorithm allows it to offer *on-line* results. However, incorrect poses may appear due to tracking errors, disparity noise or occlusions. These incorrect poses are not only detected, but also avoided using a novel alternative pose search algorithm detailed in section 3.5 and based also in analytic computation.

As depicted in Fig. 3.1, the HMC subsystem returns to the face detection phase if tracking problems arise during the perception of the performed motion. These problems involve losing the tracked regions or receiving incoherent data for a certain amount of frames. On the other hand, when different tracked regions overlap (e.g. a hand touching the face), the imitation system also returns to face detection phase once it detects the overlapping has finished, and thus it is again possible to track human head and hands. Returning to face detection implies that the motion is not captured for some frames after the overlapping. Finally, it is important to consider that in this last case the imitation system selects as tracked face the one that is closest to the last position detected for the performer’s face. This increases the possibilities of tracking the correct person after the overlapping.

### 3.3 Face detection

The first task the robot has to address to achieve gesture recognition is the detection of a human that pretends to start a demonstration. In the considered RLbI scenarios these demonstrations are executed in the context of a social interaction between the human and the robot, in which demonstrator and imitator face each other. The proposed system considers that the first step to locate a human standing in front of the robot is to locate his/her face. Appendix B details the method, proposed by Viola and Jones (2004), used in this thesis to perform this operation.

The results described in appendix B for the selected face detector show that it may produce too many incorrect results. Even when not built classifiers, but the ones provided by OpenCV computer vision library are used, the percentages of false negatives and false positives remain too high. It is important to avoid the social robot trying to imitate an incorrectly detected person, and thus the robustness of this system should be improved.

The analysis of the incorrect frames show that erroneous frames are usually isolated. Thus, to increase the robustness of the detector, the human imitation system considers that a detected face corresponds to a valid human performer only when it has been detected in  $k$  consecutive frames. Performed tests show that, in the proposed RLBI scenarios, if  $k = 5$  the percentage of false positives and negatives is constrained to less than 3%. Once a face has been definitively detected, its bounding box and 2D centroid, in image coordinates, are stored to be used in further steps.

Fig. 3.2 shows the face detection system applied to an example sequence in which a person tries to start an interaction process. It can be seen that false positives (Fig. 3.2.c and Fig. 3.2.g) and false negatives (Fig. 3.2.d) commonly occur. Besides, not only the face of the performer, but also faces of other people in the field of view of the cameras may be detected, even when they are not facing the robot (Figs. 3.2.f and 3.2.h). As depicted, despite these problems, if only faces that are detected in  $k = 5$  consecutive frames are selected, the system is able to correctly perceive performer in most situations.

The last consideration for the face detector is the possibility of detecting more than one face for  $k$  consecutive frames. If this occurs, stereo information is used to determine which of these faces is closest to the vision system. The closest face will be selected as the face of the human to track. An exception to this rule is the case in which the tracked face overlaps with one of the tracked hands during the performance. After overlapping it may be difficult to decide which of the skin color object regions is the face. Thus, in this case the face detection module is executed again, when the overlapping finishes. The face that is closest to the last valid tracked face (before overlapping) is selected as the face of the performer.

### 3.4 Silhouette extraction

Once a face region is obtained, the silhouette of the corresponding human is extracted from input stereo images. Extracting the silhouette allows the system to restrict the search areas in the image. Besides, from this silhouette it is possible to produce an estimation of torso pose as

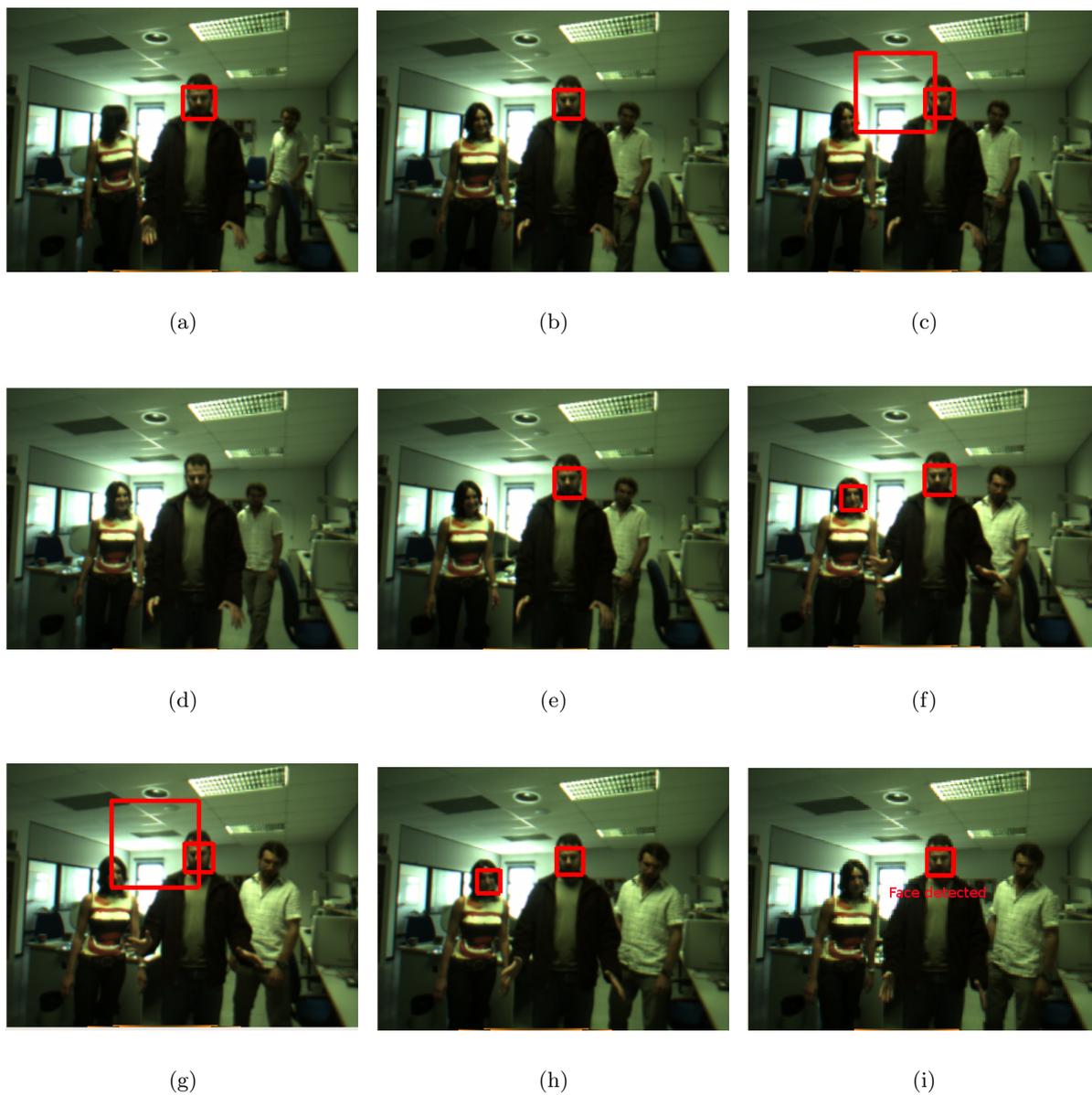


Figure 3.2: Face detection in uncontrolled indoor environments ( $k = 5$ ).

it will be further detailed.

The first step towards extracting the silhouette from perceived images is to threshold the disparity map. Only pixels which disparities are close to the disparity value associated to the detected face should be included in the silhouette. In order to define appropriate thresholds for this operation, a simple option is to consider that the maximum distance from the detected head to one hand of the same person is determined by the length of a stretched arm,  $L(\text{arm})$ . This solution can be improved considering the limitations of the arm-reachable workspace of a healthy person (Klopčar et al., 2007), that is more constrained when the arm moves backward.

Following these considerations, we use Eq. 3.1 to threshold disparity maps, where  $S$  is the set of points that conforms the silhouette. Thus, depth values  $d(i, j)$  that does not meet Eq. 3.1 are discarded. A very similar depth filtering process is applied in (Kojo et al., 2006).

$$\begin{aligned} \text{if } U_d - 25 < d(i, j) < U_d + 80, & \quad d(i, j) \in S \\ \text{else,} & \quad d(i, j) \notin S \end{aligned} \quad (3.1)$$

$U_d$ , in Eq. 3.1, is the mean depth value of the face in centimeters. It can be seen that upper and lower thresholds have been set to fixed, non-restrictive values. While this approach has demonstrated to offer adequate results for all the performed experiments, if the arm length of the performer were known these thresholds could be adapted to each particular case.

Fig. 3.3 shows three different input images from the left camera<sup>1</sup>. The corresponding disparity maps are shown in Fig. 3.4. The results of applying the threshold operation represented by Eq. 3.1 to these input images are depicted in Fig. 3.5. It can be seen that not only the desired silhouette, but also more image regions meet the previous condition and thus appear in this filtered image. The usual solution to this problem is to use connected components, to extract the silhouette as the set of points that are connected to the detected face. Fig. 3.6 depicts the silhouettes obtained when this method is applied.

The silhouettes shown in Fig. 3.6 are rough and noisy, as the disparity information is affected by shadows, plain areas that present no significative textures, etc. These issues can lead to critical errors if only these silhouettes are used. Fig. 3.7.a shows one of these situations, in which the shadows in the right elbow make the right hand lose connectivity with the rest of the body. Fig. 3.7.b depicts another example, in which the silhouette presents important noisy holes that difficult extracting the torso pose. These problems appear frequently, and they prevent from directly use these filtered disparity maps.

---

<sup>1</sup>The image obtained from the left camera is usually the reference in stereopsis. In fact, the origin of coordinates is usually located at the base of the left camera

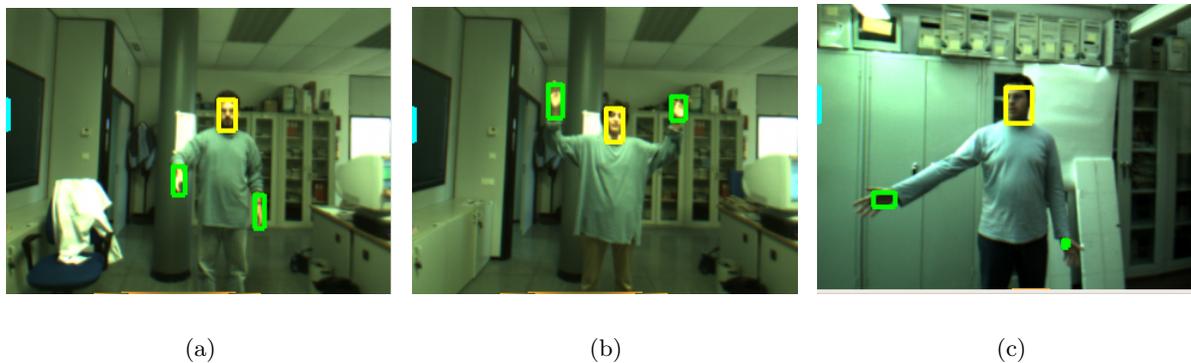


Figure 3.3: Three examples of input images from the left camera.

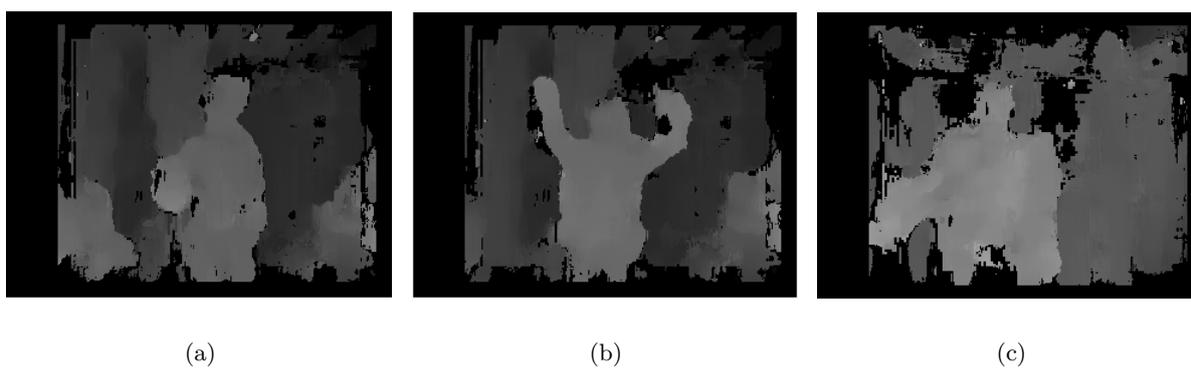


Figure 3.4: Disparity images corresponding to left input images shown in Fig. 3.3.

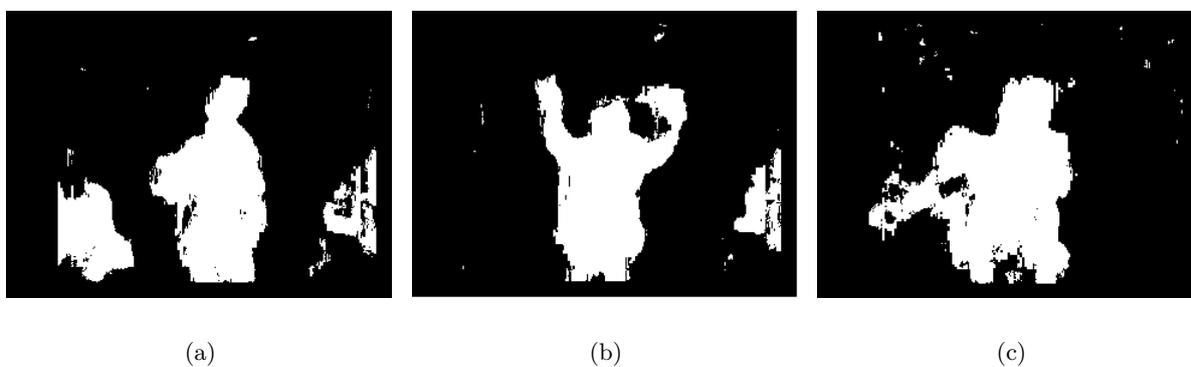


Figure 3.5: Results obtained when Eq. 3.1 is applied to disparity maps depicted in Fig. 3.4.

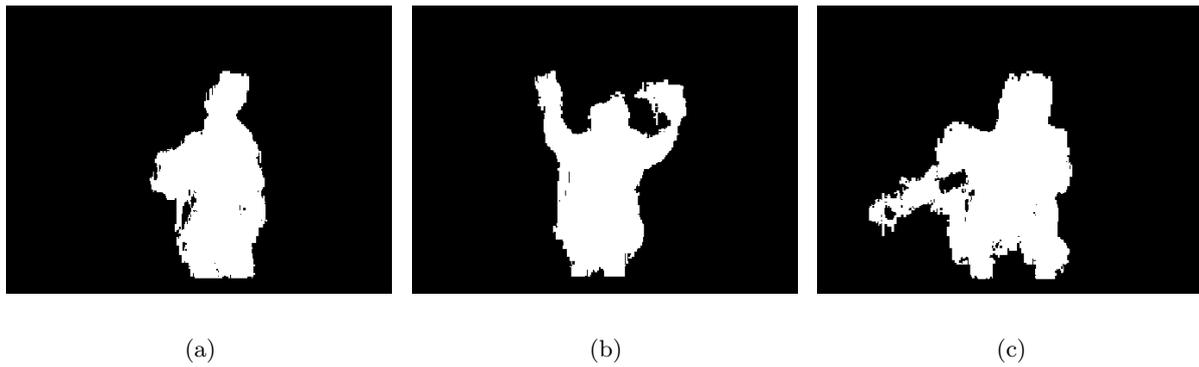


Figure 3.6: Silhouettes extracted applying connected components to maps in Fig. 3.5.

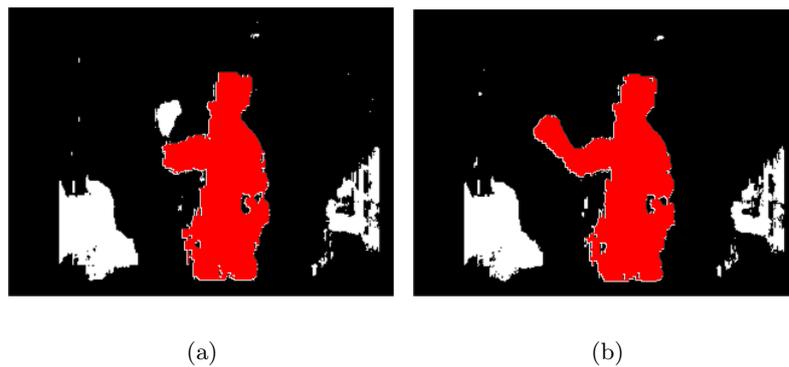


Figure 3.7: Two examples of filtered disparity maps. The silhouettes extracted if only connected components are applied are marked as red regions.

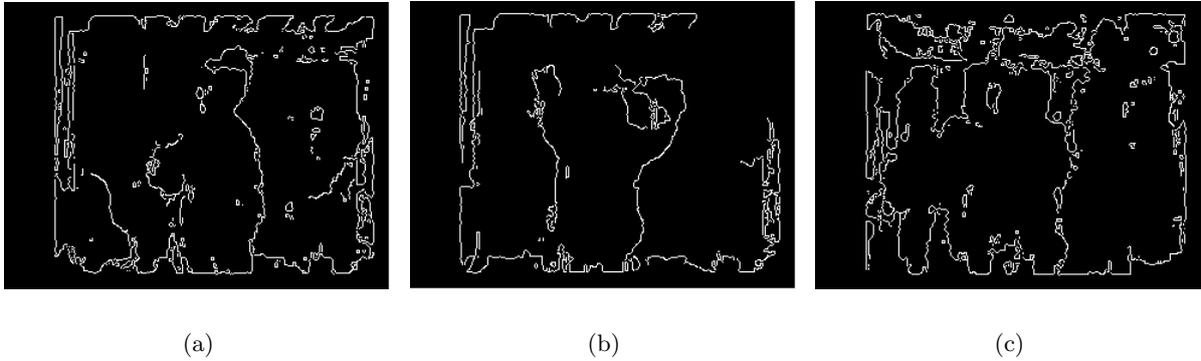


Figure 3.8: Canny borders extracted from the disparity maps depicted in Fig. 3.4.

In order to generate better silhouettes without dramatically increasing the processing time, some additional steps are applied to the filtered images in Fig. 3.5. Some researchers propose to use colour or disparity border information to locate the human silhouette and infer torso pose (Agarwal and Triggs, 2006; Kehl and Gool, 2006; Asfour et al., 2006a). These approaches usually require the person to wear specific colour garments or work in certain controlled environments. If these requirements are not met colour borders should be considered as an unreliable information source. On the other hand, disparity maps are less sensitive to lighting variations. As long as the performer does not interact with objects, disparity maps are also less affected than colour images by incorrect merging of non-interesting regions to the desired silhouette. Thus, in the particular RLbI scenarios addressed in this thesis, the results obtained when the border detection algorithm is run over disparity maps are usually better, but they are still noisy. It would be difficult to use these borders by themselves to extract human silhouette. But they may be useful, on the other hand, to complement the filtered disparity silhouettes, reducing the influence of noisy border pixels and holes.

The well-known Canny edge detector [Canny \(1983\)](#) is employed to extract edges from the disparity maps (Fig. 3.4). The details of the particular implementation of the Canny detector used in this thesis are explained in appendix C. Fig. 3.8 shows the borders obtained when this method is applied to the disparity maps shown in Fig. 3.4. It can be seen that most silhouette borders are correctly extracted, but too many additional, non-silhouette borders are also located. Fig. 3.8.c represents a situation in which extracted borders are particularly noisy.

Only silhouette borders should be considered to reinforce disparity information. It may be difficult to extract these borders from the results of the Canny detector. But it is possible to restrict the valid borders to the vicinity of the detected human. In order to obtain the image pixels that constitutes this vicinity, for each frame the thresholded region that contains the

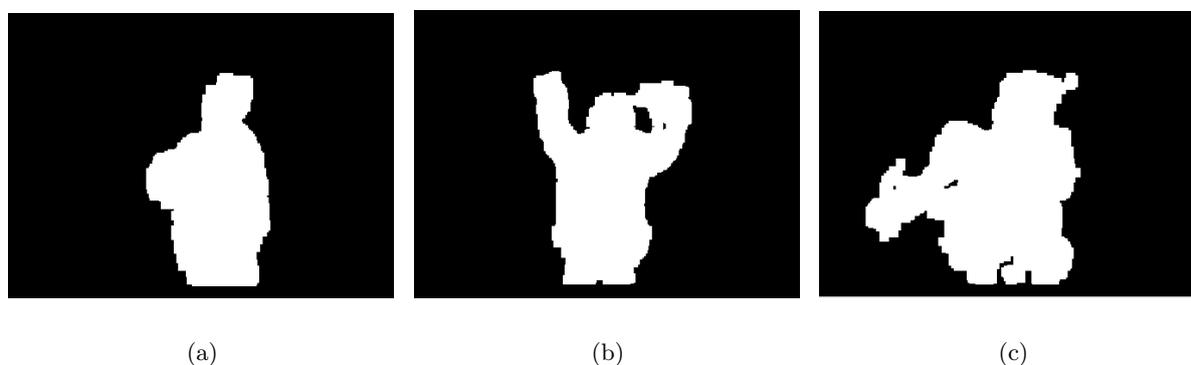


Figure 3.9: Regions in which silhouette borders of people depicted in Fig. 3.3 are searched.

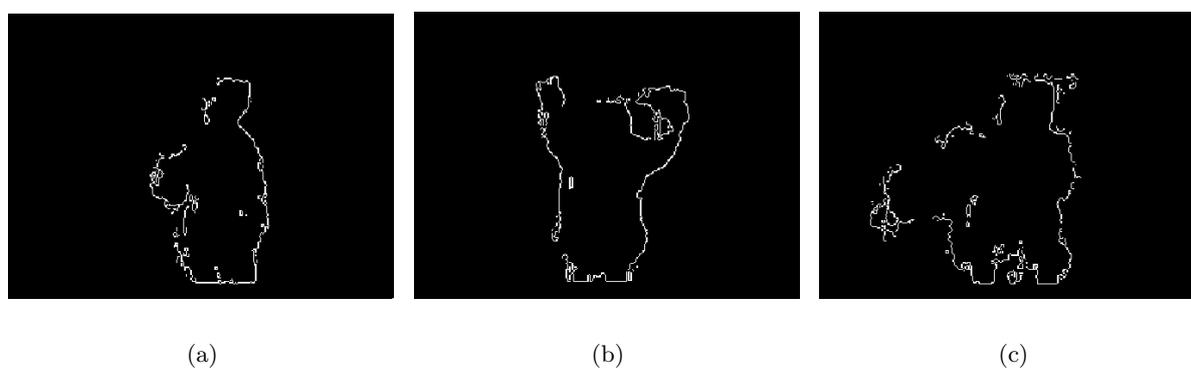


Figure 3.10: Canny borders located inside the search regions depicted in Fig. 3.9.

human face (Fig. 3.6) is firstly affected by two consecutive dilations, as depicted in Fig. 3.9. Then, all borders outside this region are discarded. The remaining borders, shown in Fig. 3.10, contain all silhouette borders. Noisy borders inside the silhouette are still present. However, these borders will not affect the results, as it is explained below.

Selected Canny borders (Fig. 3.10) are merged to silhouettes obtained when connected components were applied to filtered disparity maps (Fig. 3.6). This merging operation is performed by computing the logical OR between pixels in both images. Incorrect silhouette borders located inside the silhouette will either have no effect, or fill noisy holes in the filtered disparity maps. Once the merging step is complete, a *closing* (erode-dilate) morphological operation is applied to fill noisy holes that may remain inside the silhouette and smooth silhouette borders.

The final results for the sample images used throughout this section are depicted in Fig. 3.12. Some noisy pixels are still present at the borders, and some holes were not completely removed in the closing step. But the obtained silhouettes improve the results obtained when

only border or disparity information is used. They are also precise enough as to be used to offer an estimation of the torso pose, as it will be explained in further sections. Finally, it may be argued that a simple dilation of the disparity silhouette could have offered very similar results to those obtained by the proposed method. This is essentially true for many situations. In fact, in the first steps of this research dilated silhouettes were used to restrict the search area for the human face and hands (Bandera et al., 2008a). However, simple dilation presents some disadvantages that should be taken into account:

- Canny edges refine silhouette borders. Silhouettes extracted using only dilated disparity maps present rougher borders. Thus, it may be more difficult to use them to infer torso pose.
- If dilation is applied before constraining the silhouette to connected components, undesired regions of the image could be merged to the performer's silhouette. But if dilation is applied after connected components, it does not correct issues as the one depicted in Fig. 3.7.a.
- The dilation process required to fill noisy holes inside the silhouette tend to fill also holes that should not have been modified (e.g. if one arm lays near the torso it should not be merged to it). The comparison between figures 3.6.b and 3.12.b shows that the hole between the head and the left arm is larger when the proposed method is applied, thus the obtained silhouette will be closer to the real one.
- The proposed method firstly merges borders and disparity silhouette. Then it erodes and dilates the resulting binary images. This allows filtering small border errors. On the other hand, if a dilation is directly applied over the disparity silhouette, some of these small errors are magnified. The left part of the head of the silhouette depicted in Fig. 3.6.c is an example of these problems.

Fig. 3.11 depicts the silhouettes obtained when the proposed method is applied to the same input data that were used to obtain the maps in Fig. 3.7. The comparison between these two figures provides a good example of the advantages of the proposed method. These advantages, and its small computational cost, motivated the inclusion of this method in the proposed robot perceptual system.

Once the silhouette has been obtained, the human head and hands are determined as the three largest skin coloured regions located inside of it (Breazeal et al., 2003), as depicted in Fig. 3.13. Face has been previously detected using the method detailed in section 3.3. Thus,



Figure 3.11: Final results obtained by the proposed method for the disparity maps depicted in Fig. 3.7.

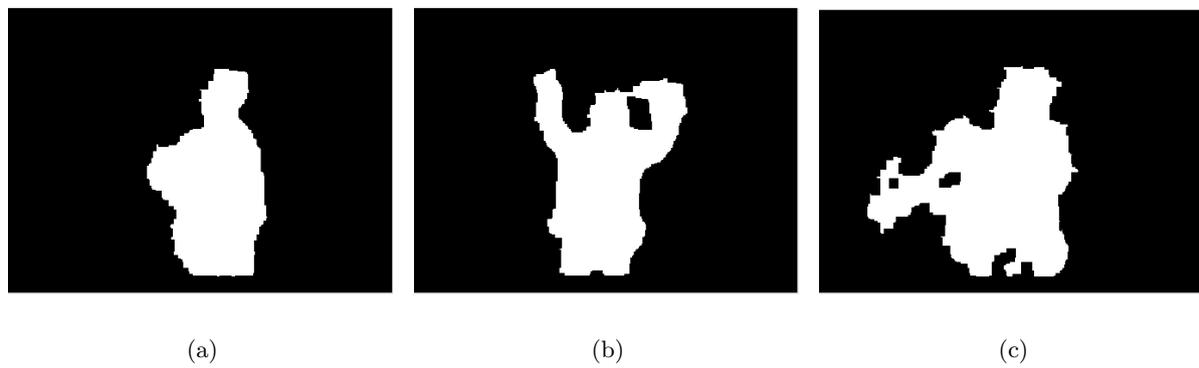


Figure 3.12: Silhouettes corresponding to input images from the left camera depicted in Fig. 3.3.

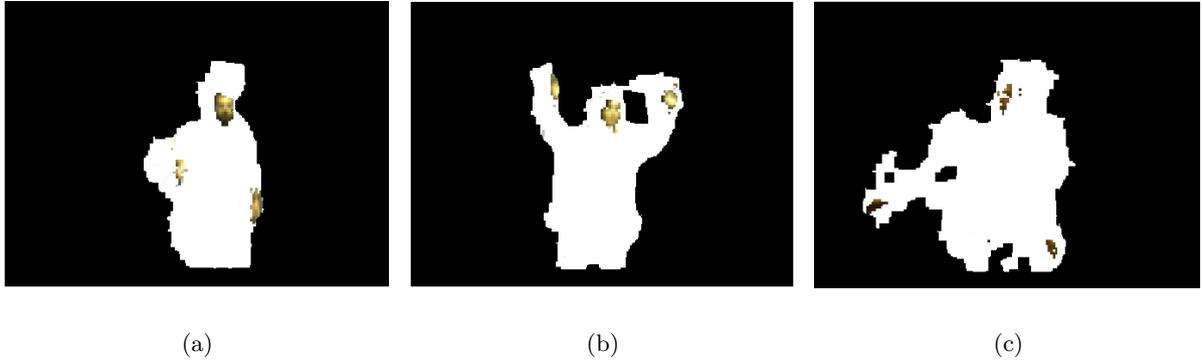


Figure 3.13: Silhouettes corresponding to input images from the left camera depicted in Fig. 3.3, containing skin color regions associated to human head and hands.

the bounding boxes of the skin colour regions of interest are compared against the face bounding box, using Eq. 3.2, where  $p_k(i, j)$ ,  $q_k(i, j)$  are the  $k$  2D vertices of each of the compared bounding boxes. The skin colour region associated to a higher  $S_{bb}$  value is labeled as the face region. It can be seen that the face detector is not executed during the tracking process. Instead, the face is tracked as a labeled skin color region, a procedure that is faster than face detection.

$$S_{bb} = \frac{4}{\sum_{k=0}^3 (\|p_k(i, j) - q_k(i, j)\|_2)} \quad (3.2)$$

In order to distinguish left and right hands, the remaining two largest skin colour regions located inside the silhouette are initially labeled considering that in the initial frames the left hand is located in the left part of the body, and the right hand is located in the right part of the body. This is a soft initialization constraint, although it has to be taken into account. It must be noted that this process is run only as an initialization step, i.e. to search for a human demonstrator. Once the demonstrator has been found and these three skin colour regions have been labeled, face detection algorithm needs only to be executed again if the face region is lost. This increases the efficiency and frame rate of the system.

### 3.5 Pose estimation

Human pose is estimated using perceived silhouette and tracked 3D centroids of head and hands. These data can be considered the main elements of social gestures (Breazeal et al., 2003), apart from face expression and finger movements, that lie beyond the scope of this thesis. As detailed above, these silhouettes and tracked centroids may be noisy due to tracking errors, partial occlusions, disparity noise or calibration errors. Thus, in order to constraint the movements of

tracked items to valid person poses, the HMC system employs an internal model of the human. This model will also provide the sets of joint angles that correspond to perceived human motion, and that will later be used by the retargeting module to help translating human motion to robot motor commands.

In order to create an useful human model, proportions should be carefully considered. The main issue is that people have very different body shapes, sizes and proportions depending on genre, age, lifestyle, etc. It is not possible to create a model able to adjust to any performer without executing an initialization phase (Moeslund et al., 2006), but this process may be difficult to achieve in the considered RLBI scenarios. On the other hand, throughout history many artists and researchers have tried to define what should be considered a standard or average human body. While these efforts have finally become a scientific discipline, the anthropometry, they obviously fail offering complete responses. However, some results derived from these studies may be useful to set average human adult proportions. The use of a generic model that can fit a high percentage of the possible performers is an interesting choice for the considered RLBI scenarios. Thus, anthropometry is subsequently revised in the context of this thesis.

### 3.5.1 Anthropometry

Anthropometry (literally "measurement of humans") is the science that refers to the measurement of the human body. Since ancient times there has been a wide interest in knowing the distribution of mass in human body, and its anatomic proportions. This interest motivated the apparition of different canons and rules that represent what should be considered an average human body. Thus, in 3000 BC Egyptians used as standard the sole. At the end of the Ptolemaic dynasty this standard was changed and the length of the middle finger was adopted as the new measure unit. The average human body was supposed to measure 19 of these units. Polykleitos, in the ancient Greek, used the hand palm for the same purposes. The Roman architect Vitruvius, in the first century, postulated that the human height equals the distance between fingertips if both arms are stretched. This idea inspired Leonardo da Vinci to paint his famous "Vitruvian Man", depicted at Fig. 3.14.

Since XIX century, anthropometry has received a growing interest from researchers and developers related to criminology, clothing and industrial design, ergonomics and architecture (Peasant, 1986). The importance of anthropometry in ergonomics is reflected by the apparition of ISO 13407:1999 norms *Human-centred design processes for interactive systems*, as well as the associated norm, the ISO TR 18329, *Human-centred lifecycle process descriptions* (Earthy et al.,

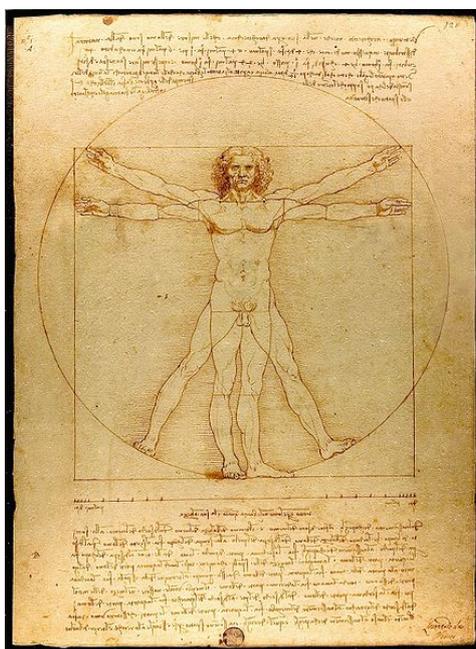


Figure 3.14: Vitruvian Man, drawing by Leonardo da Vinci (around 1487).

2001).

In this thesis we adopt the anthropometric tables proposed by [Contini \(1972\)](#). These researchers obtained the volume of each body part using the Principle of Archimedes. They submerged sequentially different body segments and supposed elliptical sections to infer anthropometric data. Lengths of different body links were also obtained using specific estimation measurements for living subjects ([Contini, 1972](#)). While different tables were obtained in this work for both male and female subjects, in the proposed RLbI scenarios the robot has no knowledge about the person gender. Thus, we adopt intermediate, non-restrictive values to model the human. These values, normalized for a human height  $H = 1$ , are depicted in [Fig. 3.15](#).

### 3.5.2 Human geometric model

As our RLbI system is restricted to upper-body movements, the used human geometric model contains parts that represent hips, head, torso, arms and forearms of the human to be tracked. These body parts are represented as fixed meshes of triangles that adjust to human proportions and are used to compute collisions. These meshes have been modeled using low-polygon modeling to achieve higher frame rates, a technique borrowed from computer game developers ([Walker and Walker, 2001](#)). [Table 3.1](#) details the amount of polygons used to model each body part.

[Fig. 3.16](#) shows the upper-body human model obtained from these meshes. It can be

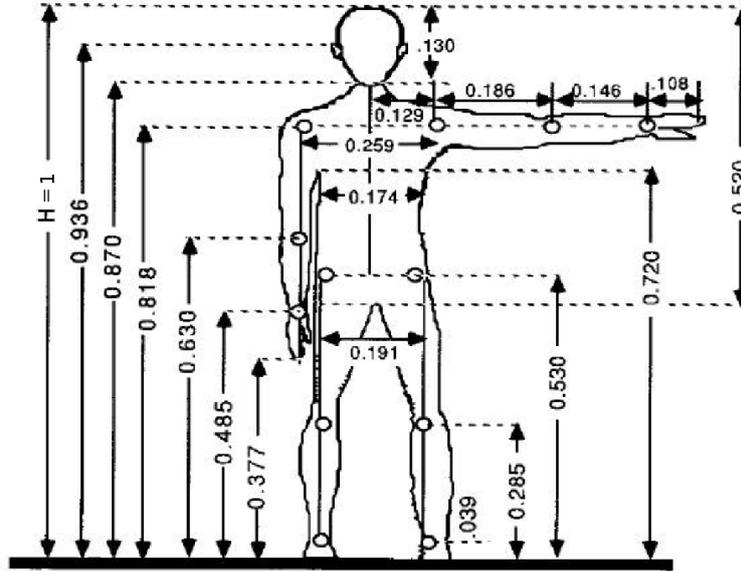


Figure 3.15: Anthropometric values used in this thesis.

Table 3.1: Number of triangles used to model each body part.

Body part	Triangles used
Head	116
Left arm	32
Left forearm	80
Right arm	32
Right forearm	80
Torso	108
Hips	60
<b>Complete upper-body</b>	<b>508</b>

seen that model proportions have been set to average human values, using the anthropometric tables depicted in Fig. 3.15. As face expression or precise finger movements are not considered in this thesis, this model is sufficient, while it allows fast computation<sup>2</sup>.

The meshes that represent each body part are rigidly attached to different coordinate frames. The set of coordinate frames and attached meshes is organized hierarchically in a structure called a scene graph (Martz, 2007). A scene graph is a hierarchical tree structure used in computer graphics to organize spatial data for efficient rendering. Scene graphs are a generic concept that can include very different sets of nodes, such as objects, body parts, light sources,

<sup>2</sup>More realistic models, composed by more triangles and including flexible meshes, are currently being developed in our research group. These models may represent poses more precisely. They would also allow to represent gestures involving finger movements or face expressions.

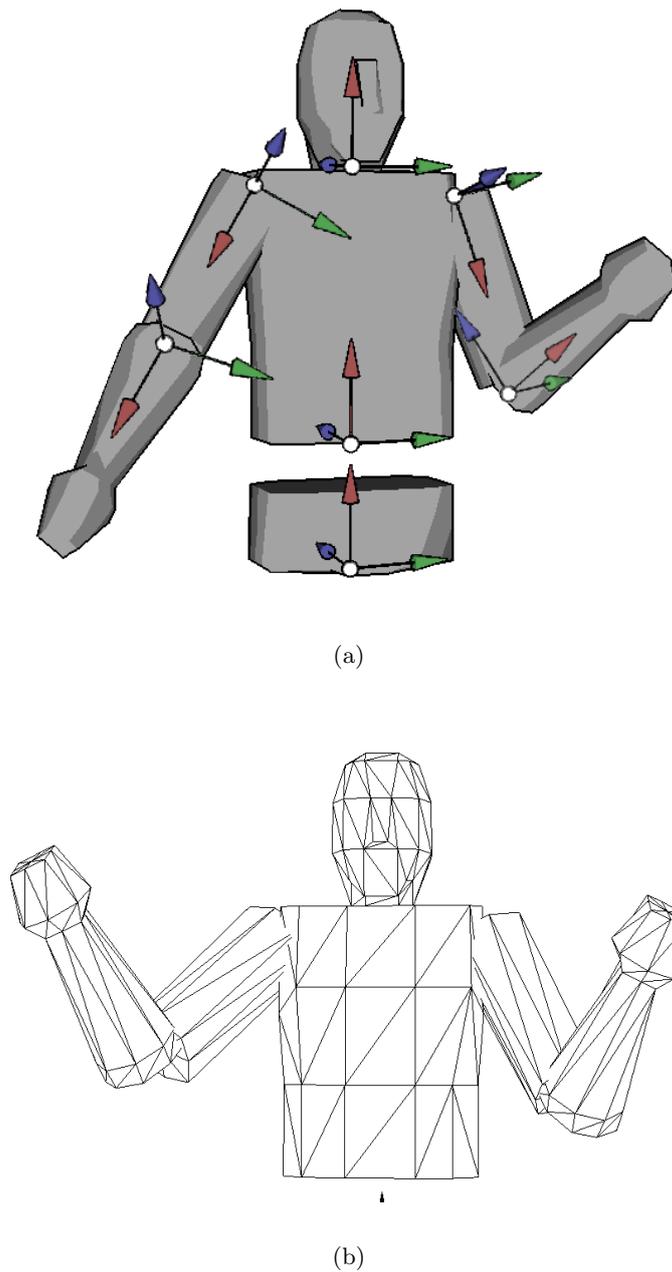


Figure 3.16: (a) Illustration of the human upper-body kinematic model; and (b) human upper-body kinematic model showing the triangles used to model each mesh.

materials or terrains. Each of these nodes in the scene graph can have zero or more children. The transformation defined for a parent root also affects its children, but not viceversa, e.g. if a parent node is translated to a different 3D position, all its children are also equally translated. Thus, scene graphs ease establishing dependencies between different elements in the model, and computing geometric transformations that propagates through these dependencies. It is this characteristic what makes scene graphs become an interesting tool to build the proposed human model.

Fig. 3.17 depicts the scene graph used to model the human upper-body. The root of the tree is the coordinate frame attached to the hips<sup>3</sup>, that represents the global translation and orientation of the model. Subsequent vertices in the tree represent the three-dimensional rigid transformations between a vertex and its parent. Triangle meshes are included as children of the coordinate frames to which they are attached.

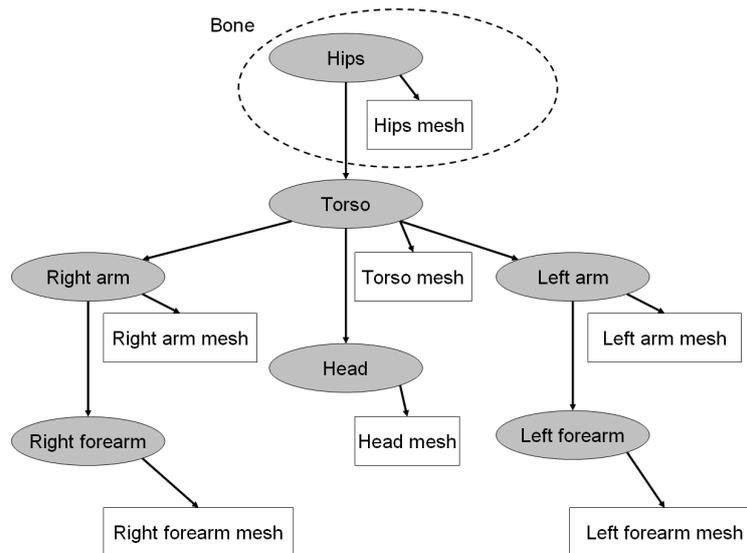


Figure 3.17: Scene graph used by the upper-body human model. The skeleton is composed by the ellipsoidal grey nodes.

It can be seen that, if meshes are eliminated from this representation, information about the kinematic relations between different body parts is still stored in the scene graph. These simplified scene graphs, that only model a certain kinematic system, are normally called skeletons or kinematic chains. The skeleton of the proposed human model can be seen in Fig. 3.18, where connections between different vertices of the kinematic chain are represented as line segments. On the other hand, each node of these skeletons, together with its corresponding body part

<sup>3</sup>Hips are a common reference point in motion generation in general, and upper-body motion generation in particular.

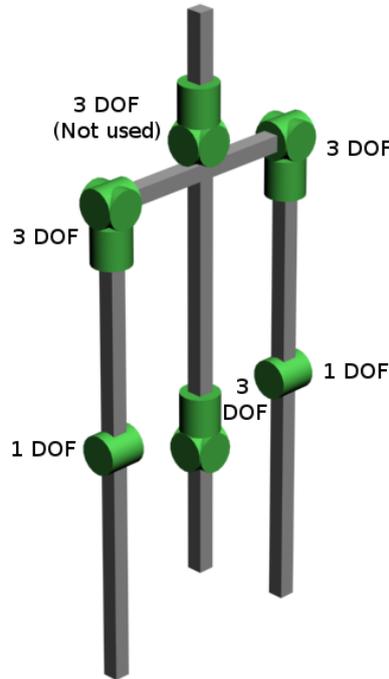


Figure 3.18: Skeleton and DOFs for the upper torso of the human model.

attached is called a bone (Fig. 3.17). Each bone is allowed to rotate –but not translate– with respect to its parent around one or more axes, or Degrees of Freedom (DOFs). Thus, at a particular time instant  $t$ , the pose of the skeleton can be described by  $\Phi^{(t)} = (\vec{\Theta}^{(t)}, \vec{s}^{(t)}, \vec{\theta}^{(t)})$  where  $\vec{\Theta}^{(t)}$  and  $\vec{s}^{(t)}$  are the global orientation and translation of the root node, and  $\vec{\theta}^{(t)}$  is the set of relative rotations between successive children.  $\vec{\theta}^{(t)}$  can also be understood as the set of joint angles rotated around each of the DOFs of the model. For upper-body motion tracking, it is assumed that only  $\vec{\theta}^{(t)}$  needs to be updated.

The model we are using to represent the human contains 11 DOFs that are distributed as depicted in Fig. 3.18. Neck DOFs have not been considered in this thesis, that focuses more on arm motion and general torso pose, although head motion will definitely be addressed as future work. For each time instant  $t$ , the set of joint angles can be decomposed as  $\vec{\theta}^{(t)} = (\vec{\theta}^b(t), \vec{\theta}^l(t), \vec{\theta}^r(t)) = (\theta_0(t), \theta_1(t), \dots, \theta_{10}(t))$ , where  $\vec{\theta}^b(t)$  corresponds to the three joint angles located in the torso and  $\vec{\theta}^l(t)$ ,  $\vec{\theta}^r(t)$  are the sets of joint angles located in the left and right arms, respectively. We propose to extract these joint angles by applying two steps that are detailed below. The first of these steps extracts the three DOFs of the torso, while the second estimates the pose of the left and right arms, once the pose of the torso has been estimated.

### 3.5.3 Estimation of torso pose

Different authors have addressed the problem of estimating the torso pose of a tracked human. [Calinon and Billard \(2005\)](#) use color patches to obtain the 3D position of the torso using a stereo vision system. [Asfour et al. \(2006a\)](#) impose the human performer to wear a red shirt in order to fit a basic human model to the detected edge silhouette. [Kojo et al. \(2006\)](#) rely on the extracted disparity silhouette to search for the global pose in a proto-symbol space. Other approaches ([Ardizzone et al., 2000](#); [Krüger et al., 2005](#); [Agarwal and Triggs, 2006](#); [Lee and Cohen, 2006](#)) do not impose specific color patches or garments to the user. Instead, they rely on contours. These methods are very sensitive to noise and occlusions, and require controlled environments. In order to provide a more robust method able to work in real environments, or even in crowded scenes, [Wu and Nevatia \(2005\)](#) propose to represent each human as an assembly of different body parts that are located using certain edge features. This method, however, is only able to work in static images and provides rough results that are more oriented to surveillance applications. More recently, [Hecht et al. \(2009\)](#) propose a markerless full body motion tracking system based on particle filters. This system uses stereo vision only to detect face 3D position, and manages to run at 10 Hz on a standard PC. A quantitative evaluation of pose errors is not provided. Besides, in its current stage the system faces problems dealing with occlusions, requires a certain initialization phase and has to rely on background subtraction. In any case, the use of flexible models and multiple particle filters represents an interesting approach to the addressed problem.

In any case, if no specific color patches or garments are used and the robot works in real environments, estimation of torso pose becomes a complex task. In these real situations, it is not possible to use color segmentation nor background subtraction techniques. The interaction with unexpected users makes the use of trained models difficult, as few or none prior knowledge and/or learning phases are available before interaction starts. There are, however, certain properties of the HRI scenarios that can simplify the torso pose estimation problem. In this thesis a novel approach to extract the torso pose from stereo images is proposed. In order to simplify collision and IK computations, and achieve faster system responses, no shoulder complex nor flexible spine are considered. Thus, the torso is modeled as a rigid link. The proposed method to obtain the rotation and flexion angles of the torso is based on the use of anthropometric relations applied over disparity silhouettes, and it is detailed below.

Before detailing the proposed approach, it may be interesting to mention that a different torso pose estimation method was developed and tested in the context of this thesis. This

previous method relied on the use of the Earth Mover’s Distance (EMD) algorithm to compute distances between silhouettes represented as disparity blobs. When compared against the approach proposed in this thesis, this method, detailed in [Bandera et al. \(2008b\)](#), presents some drawbacks: (i) it requires a training process; (ii) EMD can become a time-consuming algorithm if many blobs are employed to set the pose; and (iii) it is less robust against disparity noise and occlusions.

### 3.5.3.1 Torso pose estimation using anthropometric relations

Section 3.3 described how a bounding box containing the face of the human performer is extracted from stereo images using the face detector proposed by [Viola and Jones \(2001\)](#). The torso pose estimation method proposed in this thesis uses this information to delimit search regions for the torso and the shoulders.

Firstly, the height of the person is obtained. This process starts by computing the disparity value of the human face,  $d_{face}$ , as the average value of the 3x3 central region of the face bounding box. The use of nine pixels, instead of only the central one, to extract this information makes the result more robust against disparity noise. Then, the inverse of the projection matrix of the stereo cameras is used to obtain the 3D point,  $\overrightarrow{P_{face}}$ , associated to  $[(i_{face}, j_{face}), d_{face}]$ , being  $(i_{face}, j_{face})$  the central point of the face bounding box.

The vertical coordinate of  $\overrightarrow{P_{face}}$  contains information about human height. However, the origin for the coordinate system used by the cameras is usually located in the base of the left camera. It is necessary, then, to sum the height of the cameras to the previous coordinate in order to obtain human face height,  $h_{face}$ . According to anthropometric tables, the human total height  $H$  can then be obtained as  $H = h_{face}/0.936$ .

Before inferring torso pose, it is necessary to establish the correspondence between centimeters and image pixels. This relation depends not only on the distance between the cameras and the object, but also on the optical parameters of the cameras. However, if the object is centered in the image and cameras are correctly calibrated, the correspondence between image pixels and physical distances can be approximated to a linear relation, as far as disparity remains constant. Thus, for disparity values close to  $d_{face}$ , Eq. 3.3 can be applied to obtain the distance in pixels,  $(X_{pixels})$  corresponding to any distance in centimeters,  $(X_{cm})$ :

$$X_{pixels} = \frac{X_{cm} \cdot H}{0.1 \cdot H} \cdot h_{bbox} = 10 \cdot X_{cm} \cdot h_{bbox} \quad (3.3)$$

Table 3.2: Anthropometric values used to constraint the torso and shoulder search regions.

Body segment	Value (cm)	Value (pixels)
Neck base - Waist	$0.288 \cdot H$	$2.88 \cdot h_{bbox}$
Left shoulder - Right shoulder	$0.259 \cdot H$	$2.59 \cdot h_{bbox}$
Left hip - Right hip	$0.191 \cdot H$	$1.91 \cdot h_{bbox}$
Face center - Neck base	$0.102 \cdot H$	$1.02 \cdot h_{bbox}$
Torso search region width	$0.389 \cdot H$	$3.89 \cdot h_{bbox}$

being  $h_{bbox}$  the bounding box height in pixels.

If human torso is not excessively flexed forward or backwards, then its average size, in pixels, can be approximated using Eq. 3.3 over anthropometric values shown in Fig. 3.15. In other words, once  $h_{bbox}$  has been obtained, anthropometric measure  $X_i$ , in pixels, can be computed as  $10 \cdot X'_i \cdot h_{bbox}$ , being  $X'_i$  the anthropometric measure, in centimeters. Table 3.2 depicts the anthropometric values used in this approach. These values define search regions. The used anthropometric tables are valid for most adult people, being them male or female (Contini, 1972). The system would need different tables to perceive, for example, small children movements. Other authors adapt anthropometric values to different people by executing a certain initialization phase (Azad et al., 2007b). However, unsupervised initialization may be difficult to achieve in real uncontrolled environments, in which noisy data and occlusions are present. On the other hand, manually performed initialization is not adequate for a system to be integrated in autonomous social robots. Thus, we finally decided to rely on the tables given by Contini (1972), that fit most users.

The approach finally considers that the performer is standing still in front of the robot in the initial frames, and that the waist is fixed (i.e. the performer is not walking around while gesturing to the robot). Given the previous considerations, the torso search region is computed considering the anthropometric relations in Table 3.2 and the torso pose in previous frames.

Lateral flexion (left / right) of the human torso ( $\alpha$ ) can be computed applying trigonometric relations to Fig. 3.19, where displayed segments have the dimensions depicted in Table 3.2.

Forward/backwards flexion is computed using a very similar criterion. Thus, the forward/backwards flexion angle  $\beta$ , depicted in Fig. 3.20 is computed using Eq. 3.4, where  $z_{face}^j$  is the distance from the cameras to the face for frame  $j$ .

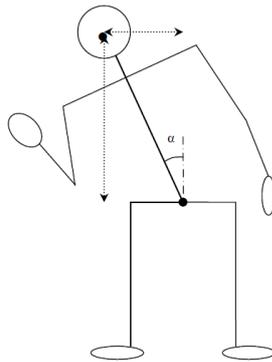


Figure 3.19: Lateral flexion of human torso.

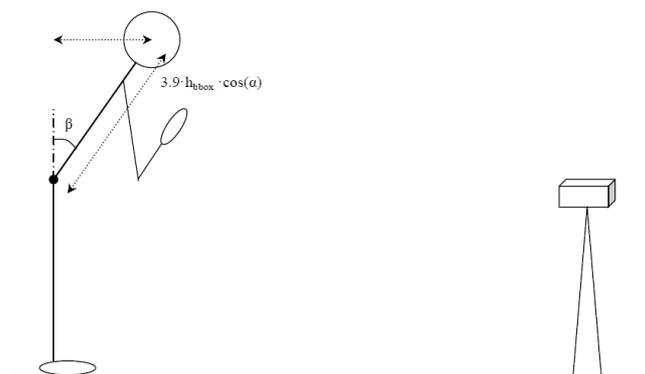


Figure 3.20: Forward/backwards flexion of human torso.

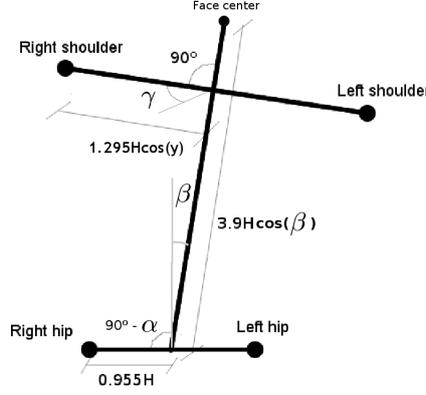


Figure 3.21: Geometric system showing the center points of the shoulder search regions.

$$\beta = \arcsin \left( \frac{z_{face}^0 - z_{face}^i}{3.9 \cdot h_{bbox} \cdot \cos \alpha} \right) \quad (3.4)$$

Once torso flexion has been obtained, the central point of the hips is searched in an oversized area. The location of this area is determined by using the face position detected for the current frame, and the obtained  $\alpha$  and  $\beta$  flexion angles. The width of the search region equals  $m \cdot w_h$ , being  $w_h$  the width of the hips provided by anthropometric tables. This oversized region is used to include torso lateral flexion.  $m$  is used to set the maximum allowed lateral flexion, and it is set to 1.5 for all depicted experiments.

In order to constraint the search region for the torso, it is necessary to estimate shoulder positions. These data are also required to compute torso rotations. However, it may be difficult to obtain shoulder positions directly from input color or disparity images. Thus, we propose the use of anthropometric tables, information about torso flexion and previous rotations not to compute the 3D positions of shoulders, but to estimate the areas of the disparity silhouette in which the shoulders are most probably located.

The proposed method applies trigonometric relations to the geometric system depicted in Fig. 3.21 to extract Equations 3.5, 3.6, 3.7 and 3.8. These equations are used to obtain the image pixels where right and left shoulders are most probably located. These pixels are labeled  $(i_{rightsh}, j_{rightsh})$  and  $(i_{leftsh}, j_{leftsh})$ .  $\alpha$  and  $\beta$  are the torso flexion angles, and  $\gamma$  is the torso rotation angle of the previous frame.

$$i_{rightsh} = \begin{cases} i_{face} + 1.02 \cdot h_{bbox} \cdot \cos \beta & \alpha = 0.0 \\ i_{face} + (1.02 + 1.29 \cdot \cos \gamma \cdot \tan \alpha) \cdot h_{bbox} \cdot \cos \alpha \cdot \cos \beta & \alpha > 0.0 \\ i_{face} + (1.02 - 1.29 \cdot \cos \gamma \cdot \tan |\alpha|) \cdot h_{bbox} \cdot \cos \alpha \cdot \cos \beta & \alpha < 0.0 \end{cases} \quad (3.5)$$

$$\dot{j}_{rightsh} = \begin{cases} j_{face} - 1.29 \cdot \cos \gamma \cdot h_{bbox} & \alpha = 0.0 \\ j_{face} + (1.02 \cdot \cos \beta \cdot h_{bbox} \cdot \tan \alpha - 1.29 \cdot \cos \gamma) \cdot h_{bbox} \cdot \cos \alpha & \alpha > 0.0 \\ j_{face} - (1.02 \cdot \cos \beta \cdot h_{bbox} \cdot \tan |\alpha| + 1.29 \cdot \cos \gamma) \cdot h_{bbox} \cdot \cos \alpha & \alpha < 0.0 \end{cases} \quad (3.6)$$

$$i_{leftsh} = \begin{cases} i_{face} + 1.02 \cdot h_{bbox} \cdot \cos \beta & \alpha = 0.0 \\ i_{face} + (1.02 \cdot \cos \alpha - 1.29 \cdot \cos \gamma \cdot \sin \alpha) \cdot h_{bbox} \cdot \cos \beta & \alpha > 0.0 \\ i_{face} + (1.02 + 1.29 \cdot \cos \gamma \cdot \tan |\alpha|) \cdot h_{bbox} \cdot \cos \alpha \cdot \cos \beta & \alpha < 0.0 \end{cases} \quad (3.7)$$

$$\dot{j}_{leftsh} = \begin{cases} j_{face} + 1.29 \cdot \cos \gamma \cdot h_{bbox} & \alpha = 0.0 \\ j_{face} + (1.02 \cdot \cos \beta \cdot h_{bbox} \cdot \tan \alpha + 1.29 \cdot \cos \gamma) \cdot h_{bbox} \cdot \cos \alpha & \alpha > 0.0 \\ j_{face} - (1.02 \cdot \cos \beta \cdot h_{bbox} \cdot \tan |\alpha| - 1.29 \cdot \cos \gamma) \cdot h_{bbox} \cdot \cos \alpha & \alpha < 0.0 \end{cases} \quad (3.8)$$

Once shoulders and hips most probably locations have been located, they conform the vertices of a trapezoidal search region which contains the torso of the performer. The points that conform the medium axis of this torso are estimated using the following procedure:

- For each row in the torso search region, the silhouette pixels are grouped into segments composed by connected pixels.
- The longest segment in the row is selected as the torso segment.
- The medium point in the torso segment is marked as a point of the medium axis.
- Once all the previous medium points have been extracted, the central limit theorem is applied to model the distribution of these points as a gaussian.  $\mu$  is the mean value of this distribution, and  $\sigma$  its variance. All points which distance to adjacent points is over  $\mu + 2 \cdot \sigma$  are then discarded as outliers (we are assuming that less than 4% of obtained points are outliers).

This procedure reduces the influence of arms or other non-torso objects appearing in the torso search region. Once these points have been extracted, the projection of the torso medium axis is computed as the result of performing a 2D linear interpolation over all of them (Fig. 3.22). Then, the depth information associated to the points in this line is used to compute the 3D position of torso medium axis.

The 3D positions of the right and left shoulders,  $\overrightarrow{P_{rightsh}}$  and  $\overrightarrow{P_{leftsh}}$ , are extracted by averaging the values in the vicinity of  $(i_{rightsh}, j_{rightsh})$  and  $(i_{leftsh}, j_{leftsh})$ , respectively. In order to avoid the influence of outliers, only the points which Euclidean distance to previous

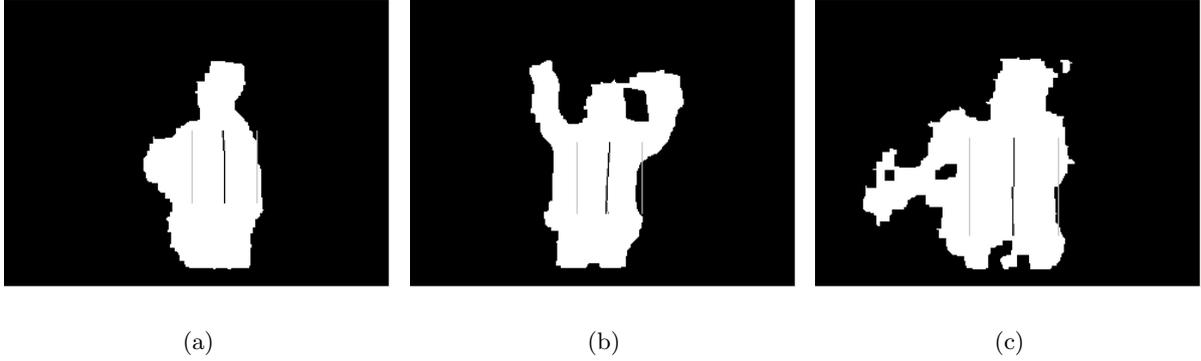


Figure 3.22: Torso medium axes extracted from the silhouette images depicted in Fig. 3.12.

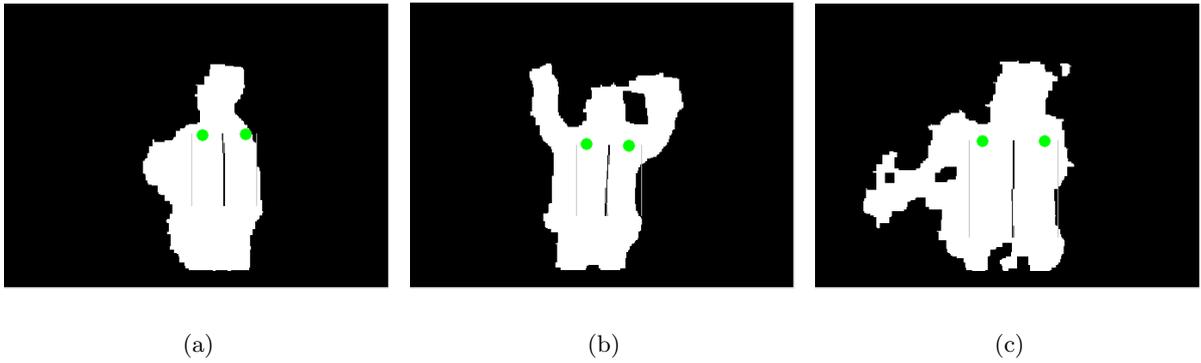


Figure 3.23: Estimated torso medium axis and shoulder locations for the performers depicted in Fig. 3.3.

shoulder position is under a certain threshold are considered to compute this average 3D value. Once obtained, these positions are used to obtain torso rotation  $\gamma$  for the current frame, using Eq. 3.9. Fig. 3.23 shows the shoulder search regions for some example silhouettes.

$$\gamma = \arccos \left( \frac{\left( (\overrightarrow{P_{leftsh}} - \overrightarrow{P_{rightsh}}) \cdot \overrightarrow{x}, (\overrightarrow{P_{leftsh}} - \overrightarrow{P_{rightsh}}) \cdot \overrightarrow{y}, 0.0 \cdot \overrightarrow{z} \right) \cdot (1.0, 0.0, 0.0) \right)}{\|(\overrightarrow{P_{leftsh}} - \overrightarrow{P_{rightsh}}) \cdot \overrightarrow{x}, (\overrightarrow{P_{leftsh}} - \overrightarrow{P_{rightsh}}) \cdot \overrightarrow{y}, 0.0 \cdot \overrightarrow{z}\|} \right) \quad (3.9)$$

### 3.5.4 Estimation of arms pose

Once the torso pose has been estimated, the tracked hand positions are used to generate valid arm configurations. An IK algorithm modified to avoid incorrect poses is used to achieve this task. Before applying this pose algorithm, perceived hand trajectories are smoothed to reduce the effects of disparity noise and outliers.

Once hand trajectories have been filtered, the method uses these trajectories plus torso

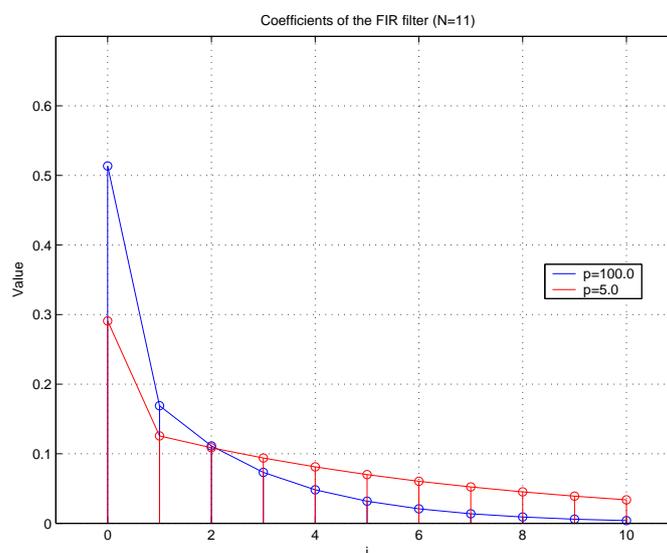


Figure 3.24: Coefficients of the FIR filter used to smooth one of the coordinates of a hand trajectory.

pose to compute a set of joint angles using a constrained model-based IK algorithm. The analytic nature of this approach allows it to offer the required joint angles in real time. The use of a model-based analytic method also avoids local minima, a common problem of optimization and probabilistic approaches (Moeslund et al., 2006).

### 3.5.4.1 Hand trajectories preprocessing

The hand trajectories are smoothed, for each frame, by applying a Finite Impulse Response (FIR) filter to each particular X,Y,Z hand coordinate. This filter, detailed in appendix E, uses different parameters for the depth coordinate, that accumulates a higher amount of noise due to disparity errors. Fig. 3.24 depicts two different impulse responses for this FIR filter. The blue coefficients correspond to faster responses used when the amount of noisy is low. Red coefficients produce a slower response, more adequate to filter depth coordinates.

### 3.5.4.2 Inverse kinematics algorithm

As shown in Fig. 3.16, each arm of the human model is modelled with a two-bone kinematic chain. These bones are articulated in the shoulder and in the elbow. For the human model, two Degrees of Freedom (DOFs) are located in each shoulder, and two DOFs are located in each elbow. As commented before, the model of the robot that is used in the motion generation component of the proposed architecture uses also the algorithms detailed in this section. These

robot models may differ in the location of the DOFs, respect to the human model.

The pose of the arms is computed (for the human or robot model) using an analytic IK algorithm that is detailed in appendix F. This algorithm considers three DOFs in the shoulder, and one DOF in the elbow, to compute shoulder and elbow rotation matrices. But only three equations, obtained from the three coordinates of the detected arm, are used to infer the values of these four DOFs. The problem is, then, unconstrained, and the elbow position is not determined. The algorithm detailed in appendix F tends to follow a smooth trajectory for the elbow, but physical constraints are not considered.

Thus, the pose generated by the IK algorithm must be analyzed in order to determine if it respects model joint limits and does not produce a collision between different links. If the obtained pose is not valid, it should be necessary to search for an alternative pose that respects both the end-effector position and the model constraints. Finally, it is also necessary to analyze the rotation matrices, obtained by the IK algorithm, to extract the joint angles corresponding to the particular used model. These processes are detailed below.

### 3.5.4.3 Detection of incorrect poses

People are not able to flex their articulations to all possible angles. There are certain joint limits that depend on the person's anatomy, age, lifestyle, etc. The kinematics model includes joint limits. It is possible to tune these limits depending on the particular performer. However, when it is not possible to access this information, it is common to set these limits to less restrictive values, thus the model can successfully track any performer while avoiding odd poses.

Probabilistic approaches such as the Markov chain Monte Carlo method used by [Lee and Cohen \(2006\)](#), or the particle filter proposed by [Hecht et al. \(2009\)](#), select a suitable pose by evaluating different possibilities for each frame. These approaches use joint limits information to reduce the amount of possibilities to be checked, as all impossible poses are discarded. Analytic methods, such as the proposed one, should also consider these limits in order to detect perception errors, help in tracking the motion and avoid incorrect learning. Besides, these analytic methods usually have to estimate the positions of certain body parts that are not directly perceived. This estimation process may also benefit from considering joint limits, to avoid odd poses.

Detection of joint limits, however, does not eliminate all incorrect poses from the repertoire of the model. There are situations in which no angle limit is exceeded, but still the performed pose is not valid, due to collisions between arms or between the arms and the torso.

While joint limits are easily added to most kinematics models, collision detection systems are more complex to implement, unless simple primitive shapes are used to build the model (Safonova et al., 2003). If a more accurate model is used, it may be complex to find a collision detection system that runs *on-line*.

On the other hand, as commented in chapter 2, in this thesis the same algorithms that are used to pose the human model are also used, in the motion generation component, to make the robot model imitate perceived movements. The only imposed requirement is that robot arms have to be modelled using the same Degrees of Freedom (DOFs) that are used to model the human ones.

The application of the previously described IK algorithms to robot models reveals some important issues of these methods, and requires to increase their robustness. Thus, the detection of limits violation and collisions is merely used to correct tracking errors and produce a more natural motion in the human model. However, when this same IK algorithm is used to set the robot poses, detection of limits violation and collisions become a crucial part of the motion generation, as an incorrect pose could damage the real robot if it is not previously detected and avoided.

- *Joint limits detection.* Given the updated shoulder and elbow rotation matrices, it is necessary to extract joint angles from these matrices that match the real DOFs of the model.

This process is made by applying a parameterization change to rotation matrices. There is a direct correspondence between Denavith-Hartenberg (DH) (Craig, 1986) parameters and model joint angles, so the local axes referred angles are converted to DH parameters. The shoulder conversion can be done applying the following parameterization to the rotation matrix  ${}^w_1R$ :

$${}^w_1R = \begin{pmatrix} c\theta_2c\theta_3 & -c\theta_2s\theta_3 & s\theta_2 \\ s\theta_1s\theta_2c\theta_3 + c\theta_1s\theta_3 & -s\theta_1s\theta_2s\theta_3 + c\theta_1c\theta_3 & -s\theta_1c\theta_2 \\ -c\theta_1s\theta_2c\theta_3 + s\theta_1s\theta_3 & c\theta_1s\theta_2s\theta_3 + s\theta_1c\theta_3 & c\theta_1c\theta_2 \end{pmatrix} \quad (3.10)$$

where  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  are the real DOFs of the model arm,  $c\theta_i$  is  $\cos\theta_i$  and  $s\theta_i$  is  $\sin\theta_i$ .

The elbow angle is directly obtained: as there is only one Degree Of Freedom (DOF) in the elbow, the local rotation angle is equal to  $\theta_4$  in the model.

The used human model, and some robot models, distribute the DOFs in the following way: two of them are located in the shoulder, and two in the elbow. But the used inverse

kinematics method provides a solution in which the shoulder contains three DOFs. It is required to translate this kinematics system to the previous ones. While this may be a complex operation in general terms, given the chosen parameterization an easy solution can be provided. The third DOF of the shoulder is made to correspond with the rotation along the segment axis, so that it can be directly mapped to the first elbow DOF, as Fig. 3.25 depicts. It must be taken into account that while our human model arm meshes has cylindrical symmetry along the segment axis, some robot arms may not meet this requirement. In these cases the consequences of this DOF change should be considered before applying it. Fig. 3.26 depicts an example of one of these situations, in which the right arm of a NOMADA virtual robot (see chapter 6) has been modified to lose its symmetry.

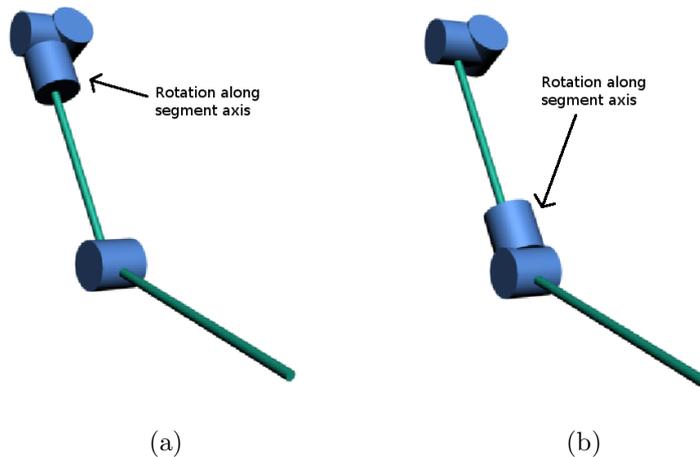


Figure 3.25: Movement of the torsion DOF along segment axis. Both configurations are equivalent as long as arm segment has cylindrical symmetry.

Once the model DOFs are computed, the system can directly check if any of them lies beyond its limits. For the robot models, joint limits are extracted from the robot specifications. For the human model, joint limits have been set to the values provided by [Maestri \(1996\)](#) for virtual character animation.

- *Collision detection.* Collision detection algorithms have to deal with certain issues, that are listed below ([Gottschalk et al., 1996](#)):
  - Models composed by hundreds, or thousands, of polygons.
  - Input models are usually collections of polygons with no topology information. These unstructured representations, usually named 'polygon soups', may present cracks, T-joints or complex, non-manifold, geometry.

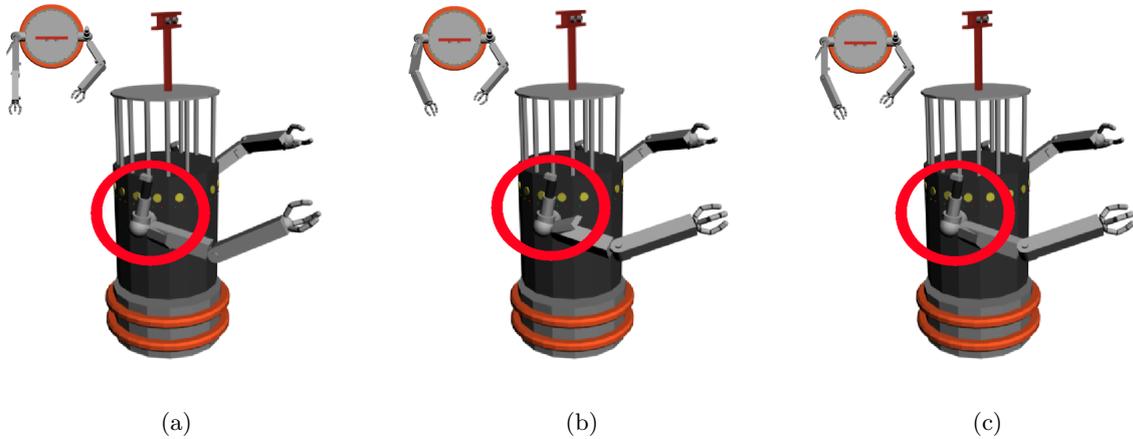


Figure 3.26: Consequences of the movement of the torsion DOF along segment axis if the arm is not symmetric: (a) Initial pose; (b) Right arm rotation (torsion DOF located in the shoulder); and (c) Right arm rotation (torsion DOF located in the elbow).

- Models should be able to come in close proximity of each other and can have multiple contacts.
- The contact between models should be known accurately, up to the resolution of the model and machine precision.

Different authors have proposed to use collision detection algorithms based on simple geometric primitives to detect collisions between different body parts, both in the model used to track perceived motion, and in the robot itself (Safonova et al., 2003). These algorithms provide rough, but fast results. They also imply additional motion constraints, as the used primitives (e.g. cylinders) are usually larger than the body segments they model. In practice, the use of simple geometric primitives to model the agents, and compute collisions, difficulties perceiving precise movements and produces unnecessary reductions in the reachable workspace.

Our method uses RAPID (Gottschalk et al., 1996) as the base of the collision detection module. RAPID is a library that computes collisions between meshes composed by triangles. This library is based in the use of tight-fitting oriented bounding box trees (OBBTrees) to model each of the two objects involved in a certain collision detection process. Collision checks are performed by projecting the oriented bounding boxes (OBBs) over a certain axis. The problem is then simplified as depicted in Fig. 3.27, where the OBBs are disjoint if  $T \cdot L > (r_a + r_b)$ . Gottschalk et al. (1996) demonstrate that only 15 different axes for each pair of OBBs have to be considered in order to find the separating axis. Thus, collision checks are quickly and efficiently computed.

In this thesis, a triangle mesh is attached to each model link. The model has also information about the pairs of meshes between which collisions should be checked (e.g. it is usually not necessary to check collisions between the foot and the head). For each frame, collisions are checked between listed pairs of meshes.

Table 3.3 shows the collision pairs defined for the human model depicted in Fig. 3.16. The system is able to check all these pairs for each frame at about 100 frames per second on a standard PC. On the other hand, it can be seen how collisions between meshes that are linked together in the kinematics chain are not included in this list. This avoids detection of spurious, undesired collisions produced by rigid meshes intersecting each other near joint locations. But some collisions that should have been detected are also masked (e.g. the forearm colliding against the torso). The use of realistic flexible meshes may reduce these issues and allow for more complete collision checks.

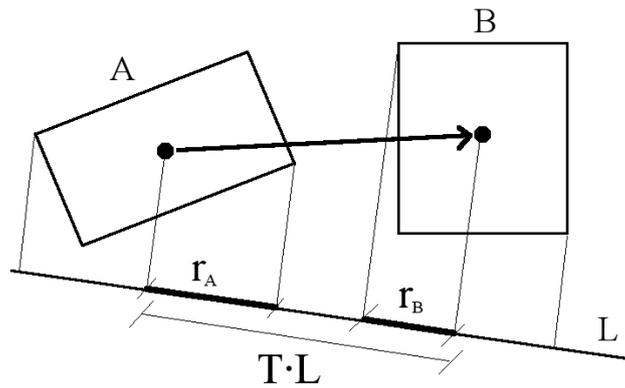


Figure 3.27: Collision check using projections of the OBBs over a separating axis.

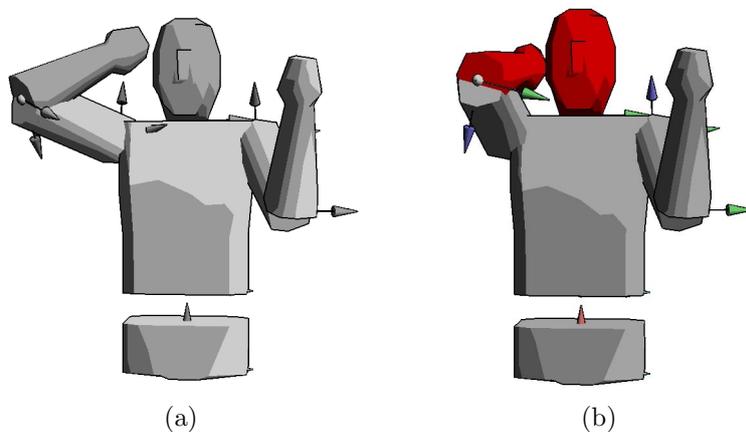


Figure 3.28: RAPID collision detection: (a) Valid pose. (b) Collision.

Table 3.3: Collision pairs defined for the human model depicted in Fig. 3.16.

Collision pairs	
Left forearm	Hips
Left forearm	Torso
Left forearm	Head
Left forearm	Right arm
Left forearm	Right forearm
Right forearm	Hips
Right forearm	Torso
Right forearm	Head
Right forearm	Left arm
Left arm	Hips
Left arm	Head
Right arm	Hips
Right arm	Head

#### 3.5.4.4 Avoidance of incorrect poses

Once the system detects an incorrect position, due to joint limit violation or collision, there are different options it can follow:

- Return to the last valid pose. This is the easiest solution and requires no additional computation, although it produces the most inaccurate results. It is only adequate for systems in which the perception error is very small, and the model is able to perform perceived pose for most frames. On the other hand, if perception errors and noise are present, this strategy tends to produce rough and staggered movements.
- Search for a different valid pose. There are model-based HMC systems in which the perceived pose is computed independently for each frame (Wu and Nevatia, 2005; Lee and Cohen, 2006). These systems, however, may find difficulties to work *on-line*. In order to achieve faster responses, model-based HMC systems usually obtain each pose as a certain modification with respect to previous poses. While these approaches are more efficient and allow faster computation, they may drive the model to poses from which it is not able to track the movements of the performer. Besides, perception errors can deviate the trajectories to unreachable positions. In these situations the model would remain bogged until a valid, reachable pose is perceived.
- Search for an alternative pose. If certain perceived positions are not reachable, the model should firstly try to find a different valid pose, as above. But if this process does not

offer positive results, then the model could at least update some of its joint angles to approach these perceived positions. This approach reduces the probabilities of bogging the model in a certain pose. It also improves HMC results, as the distance from the perceived position to the one reached by the model is reduced. Probabilistic methods, that are based on evaluating different options and selecting the most suitable one, usually follow this strategy (Hecht et al., 2009).

In this thesis, we propose an analytic HMC system that also follows the last strategy. Thus, if a certain arm pose provided by the IK algorithm is not valid, the steps detailed below are executed for this arm:

1. The system looks for alternative poses (i.e. different arm configurations). Imitation of end-effector positions requires to place hands in certain coordinates, but the elbow is free to move in the circle presented in Fig. F.1 (appendix F). Thus, alternative poses will preserve hand positions, but will move the elbow in this circle.
2. The motion of the arm should be as smooth as possible. Thus, alternatives should be more densely searched near the current elbow location. This is implemented by exponentially distributing the alternatives around the initial incorrect elbow position, as shown below:

$$\begin{aligned} \theta_{2i} &= \frac{1}{100 \frac{(n-i)}{n}} \cdot \theta_{max} \\ \theta_{2i+1} &= -\theta_{2i} \end{aligned} \quad i = 0, 1, 2, \dots, (n-1) \quad (3.11)$$

where  $\theta_{2i}$  and  $\theta_{2i+1}$  correspond to two symmetric alternatives on the elbow circle with respect to the current pose, and  $n = \frac{N}{2}$ , being  $N$  the number of alternative poses checked when current pose is erroneous.  $\theta_{max}$  sets the maximum angle that the elbow will rotate in the circle in which alternative poses are searched.

Fig. 3.29<sup>4</sup> depicts the alternatives for two different poses, when  $N = 40$  and  $\theta_{max} = 3 \cdot \pi/4$ . The human model has been replaced by a stick figure to better show the results. As required, alternative poses are placed on the elbow circle (Fig. F.1, appendix F) and are more densely distributed near the current elbow position. Examples shown in Figs. 3.30 and 3.31 use the same  $\theta_{max}$  value than in Fig. 3.29, but  $N$  has been set to 20 and 10, respectively.

---

<sup>4</sup>All figures showing different  $N$  and  $\theta_{max}$  values consider the same elbow flexion angle. The point of view, however, has been changed in certain figures to better show the alternatives.

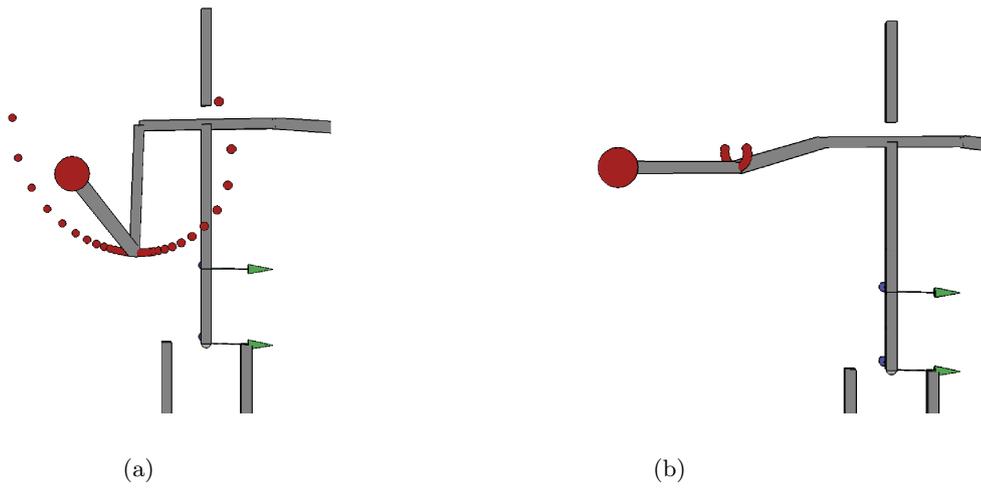


Figure 3.29: Alternative elbow locations ( $N = 40$ ,  $\theta_{max} = 3\pi/4$ ) for two different hand positions.

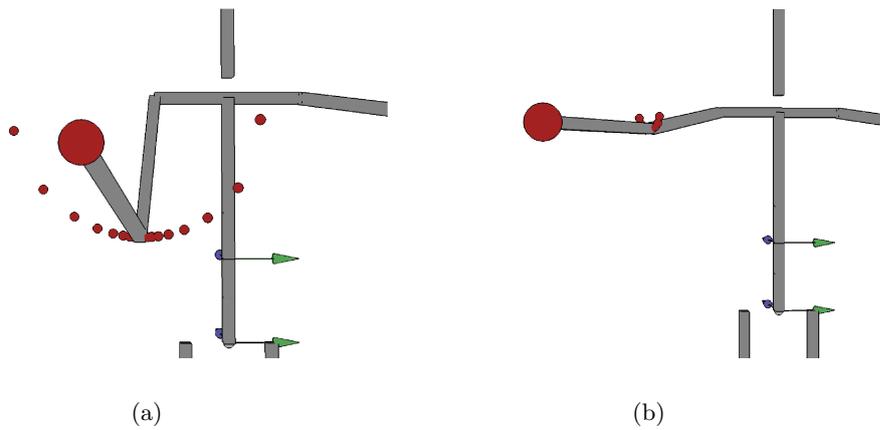


Figure 3.30: Alternative elbow locations ( $N = 20$ ,  $\theta_{max} = 3\pi/4$ ) for two different hand positions.

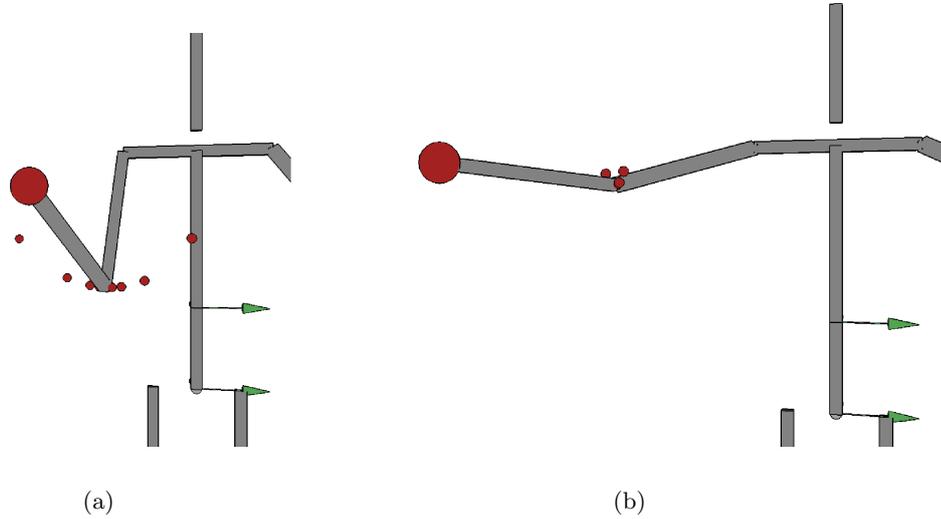


Figure 3.31: Alternative elbow locations ( $N = 10$ ,  $\theta_{max} = 3\pi/4$ ) for two different hand positions.

3. The system chooses the nearest valid alternative. As depicted in Figs. 3.29, 3.30 and 3.31, for a certain  $\theta_{max}$  alternatives are distributed in the same arc regardless of the value of  $N$ . However, it can be seen that the use of larger  $N$  values produces more densely distributed alternatives. Results obtained for different  $N$  values are very similar if low values for the elbow flexion angle  $\theta_4$  are considered. But as the values of  $\theta_4$  corresponding to an invalid pose grows, the use of densely distributed alternatives produces smoother results. Higher  $N$  values may also require more time if all alternatives have to be evaluated. However, this is not a common case, as the algorithm usually finds a valid alternative that is located near the invalid pose.

A different option to obtain a dense alternative distribution near the current -invalid- pose without increasing  $N$  is to reduce the value of  $\theta_{max}$ . As commented above, most alternatives are found near the invalid pose, thus it is usually not necessary to check alternatives that locate the elbow far from the current pose. Besides, significant elbow rotations between two consecutive frames produce non smooth motion, thus they should be avoided. The use of small  $\theta_{max}$  values appears as an interesting option to the increment of  $N$ . Figs. 3.32 and 3.33 shows examples obtained for  $N = 20$ , when  $\theta_{max}$  is set to  $\pi/2$  and  $\pi/8$ , respectively. Fig. 3.34 show the alternative distribution ( $N = 30$ ,  $\theta_{max} = \pi/8$ ) that we have adopted for most experiments.

4. If there is no valid alternative, the four DOF of the arm are independently considered. Angles  $\theta_i$  that lie beyond the limits are restored to the last valid value, but the rest of

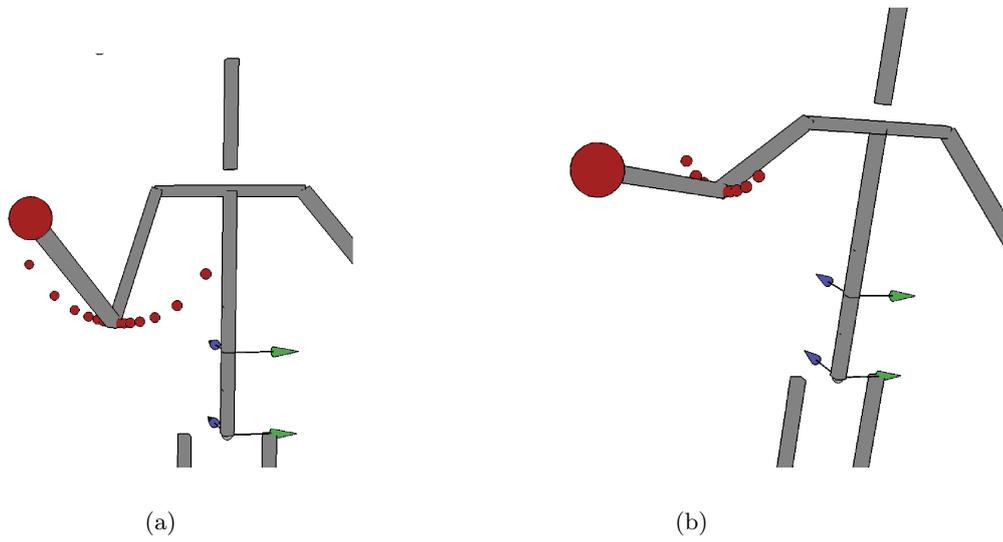


Figure 3.32: Alternative elbow locations ( $N = 20$ ,  $\theta_{max} = \pi/2$ ) for two different hand positions.

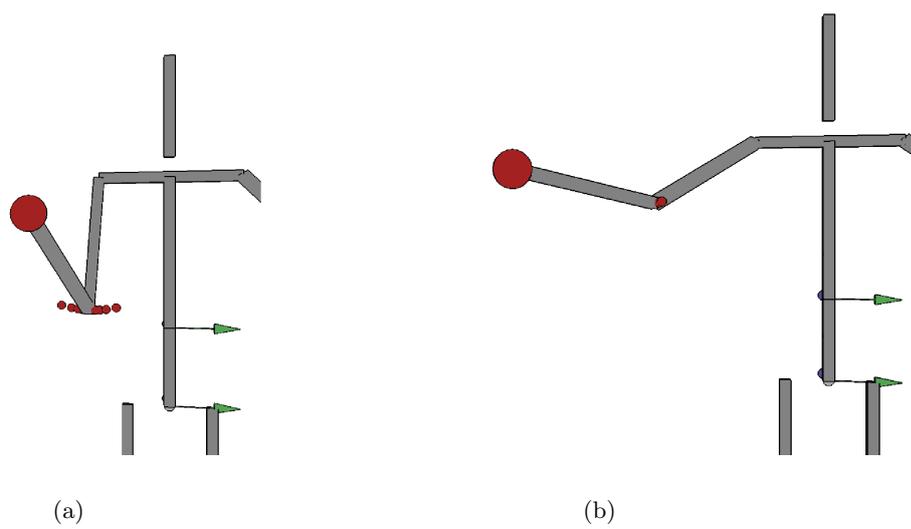


Figure 3.33: Alternative elbow locations ( $N = 20$ ,  $\theta_{max} = \pi/8$ ) for two different hand positions.

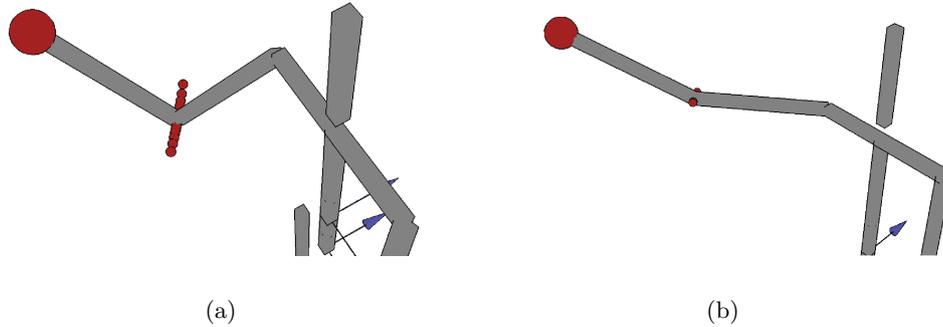


Figure 3.34: Alternative elbow locations ( $N = 30$ ,  $\theta_{max} = \pi/8$ ) for two different hand positions. These are the parameters selected for most of the experiments.

angles are updated to the new perceived value. This allows the arm to approach the perceived pose even when it is not able to reach it.

Fig. 3.35 shows a virtual imitator trying to perform a perceived right hand movement that lies beyond its reachable-workspace. The perceived hand positions are marked using red spheres. In this example, if the imitator does not find valid poses that reach these perceived hand positions, it remains static until a valid pose (Fig. 3.35.f) is perceived.

Fig. 3.36 shows the same situation depicted in Fig. 3.35, but now the imitator always updates its valid right arm joint angles, regardless the perceived hand position is reachable or not. As depicted, this approach obtains the same results than the previous one for the reachable hand positions. But it also allows to reduce errors (Euclidean distance from the perceived position to the performed position) and preserves more characteristics of the motion when perceived positions are not reachable. In the example given in Fig. 3.36, the imitator manages to imitate the swinging motion of the right arm, even when only the first and last perceived hand positions were reachable.

The speed of the process depends on the number of alternatives it needs to check. A system using a correct number of alternatives should produce smooth movements and work *on-line* even in the case in which all of them need to be checked. As commented above, for most of the experiments detailed in this thesis  $N$  is set to 30, and  $\theta_{max} = \pi/8$ . These values are able to update the pose at more than 100 frames per second, when the algorithm runs in a standard PC.

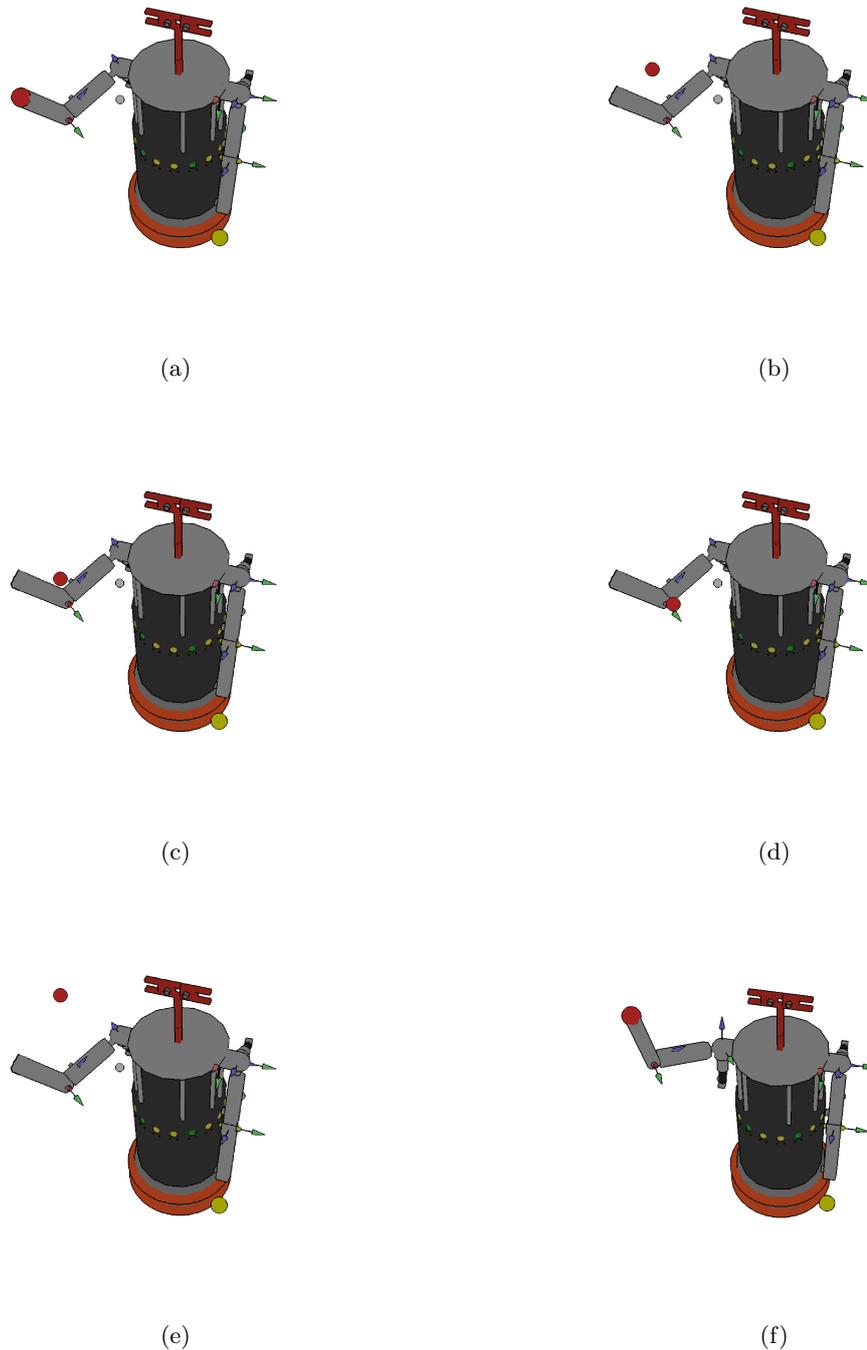


Figure 3.35: Example of a virtual model tracking a perceived right arm movement (red spheres) that lies beyond its limits. Valid joint angles are not updated. As depicted, right arm motion stops until a valid pose is provided.



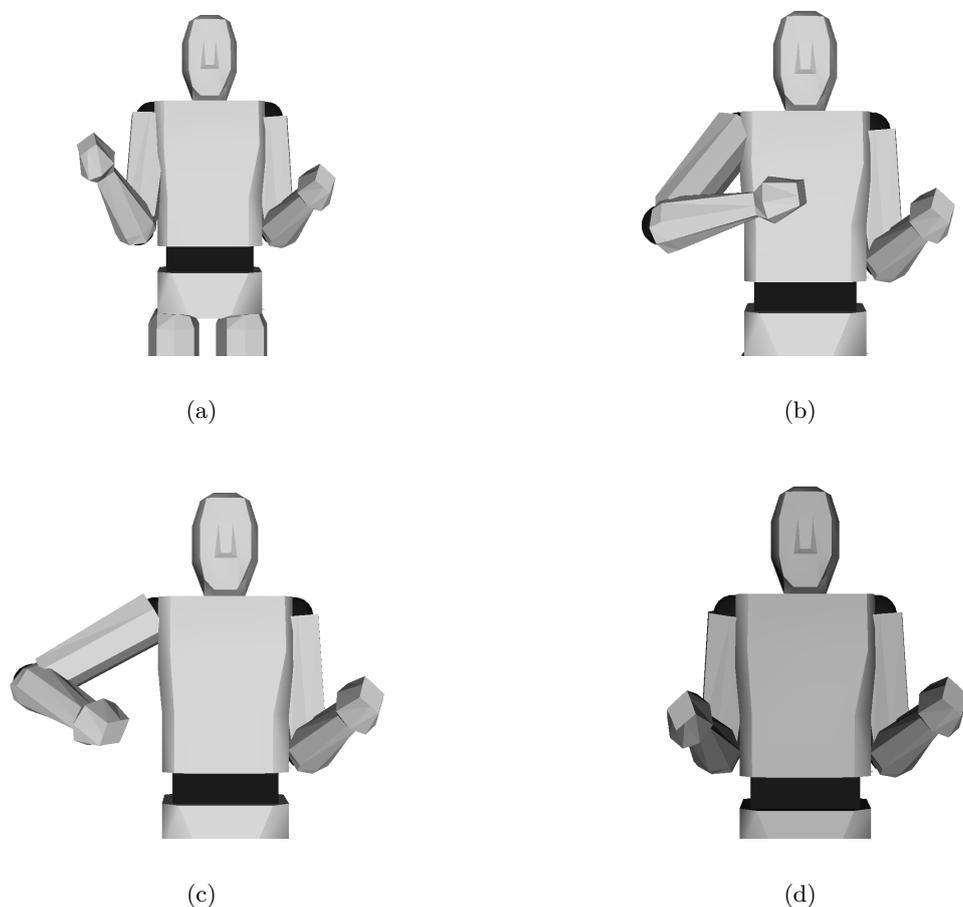


Figure 3.37: Update algorithm to obtain more natural and efficient poses: (a) Initial human model pose; (b) the right arm moves to the left. Alternative elbow positions are adopted to follow the movement; (c) the right arm moves to the right. The elbow is located in the valid pose nearest to the previous one; and (d) the system looks for alternatives that locate the elbow in a lower vertical positions to obtain a more natural and efficient pose.

The alternative evaluation module has been also used when the system is in a valid pose: in these cases, the two nearest alternatives to current pose are checked. If one of them locates the elbow in a lower vertical position, and does not produce limits violation nor collisions, then the elbow is moved there (Fig. 3.37). This procedure increases stability. It also reduces the forces that have to be applied to the joints to maintain that pose. While this only makes the human model move more naturally, as people usually adopt energy saving poses, for the models used to check robot motion the efficiency in the performed movement is an important issue regarding autonomy and motor lifetime.

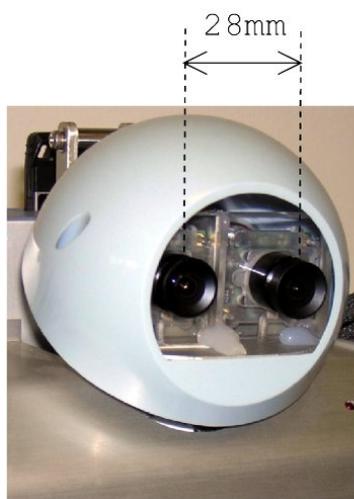


Figure 3.38: Pair of stereo cameras mounted on the HOAP-1 robot.

### 3.6 Stereo cameras mounted on the proposed RLbI system

The first experiments performed for this thesis used a Fujitsu HOAP-1 humanoid robot, in which head a pair of stereo cameras was mounted (Bandera et al., 2006). Following the design mounted by Fujitsu in other HOAP-1 robots, the two cameras were located inside the plastic carcass provided as HOAP-1 head (Fig. 3.38). This limited the baseline to only 28 mm. The image size was limited to 128x128. This reduced resolution eased working *on-line*, as less pixels per frame had to be processed. These cameras, then, were affected by small baseline and small image sizes, and thus low pixel resolution. These issues limited their capability to perceive depth information. Recognition of gestures could only be achieved if the performer stayed very close to the HOAP-1, at a distance below one meter. This was not only a very short distance for most social interactions, but it also avoided the execution of wide gestures, as they easily exceeded image borders.

As chapter 6 details, the HOAP-1 proved not to be an adequate platform to be used as a social robot, thus development started for a new platform. The current head mounted in this platform is a pair of stereo cameras, referenced STH-DCSG-VAR-C, and provided by Videre Design. These are low power consumption Complementary Metal Oxide Semiconductor (CMOS) cameras, that capture Video Graphics Array (VGA) stereo images at a maximum resolution of 640x480. Stereo images are provided at a frame rate of 20 Hz. The cameras are equipped with a global shutter system, that allows all the pixels in both cameras to be captured simultaneously, a key feature for stereo calculations. Other characteristics of these devices are listed in Table 3.4. As depicted, different lenses could be mounted on the cameras, although in this thesis the



Figure 3.39: STH-DCSG-VAR-C cameras, Videre Design.

provided CSL-2.8-1/3 lenses were maintained.

The cameras can be accessed from a PC through an IEEE 1394 digital interface, allowing the user to modify their exposure, gain, resolution, etc... It is also possible to modify the baseline by moving one of the camera supports along the provided rail. In our case, the baseline was set to 10.7 cm. This allows the cameras to capture the movements performed by the human demonstrator at a distance from 1.5 to 2.0 meters. This is an usual distance in human social interaction scenarios, and thus it has been the distance at which the experiments performed in this thesis have been conducted.

## 3.7 Evaluation of the HMC system

This section describes the experiments conducted to test the HMC system implemented in this thesis, and analyzes the obtained quantitative results.

### 3.7.1 Experimental setup

While the features of the previously described Videre stereo cameras can be addressed to obtain average and maximum errors for a perfectly calibrated system working in good environmental conditions, this information is only valid to characterize the error associated to a certain static

Table 3.4: Main specifications of the STH-DCSG-VAR-C stereo cameras.

Device	CMOS 1/3" (Micron MT9V022)
Image resolution	640x480, 320x240
Image format	8 bit, monochrome or Bayer color
Frame rate	3.75, 7.5, 15, 30 Hz 3.125, 6.25, 12.5, 25 Hz 30 Hz. max. with a resolution of 640x480
Image format	8 bit, monochrome or Bayer color
Exposure	One line to the complete frame
Gain	0-12 dB
Exposure	One line to the complete frame
Sensitivity	4.8 V/lux-sec. (monochrome)
S/N	>> 60 dB, without gain
Power consumption	< 2 W.
Synchronization	Extern: 60 $\mu$ s.
Lens focal distance	< 2.8 mm. (optional), 4.0 mm. (mounted), 8.0 mm. (optional)
Lens dimensions	3.81 cm x 2.54 cm x 6.604 cm
Weight	106.19 grams each lens module 444.85 grams the cameras and the rail 536.69 grams is the typical weight of the complete system
Baseline	5-20 cm.

marker perceived using the stereo cameras. However, the proposed vision-based HMC system considers uncontrolled environments, partial information, moving tracked items, noisy disparity, pose estimation, etc. The errors in such a system should be evaluated against a reliable ground-truth in order to determine its precision and characterize its errors.

There are different options to obtain this ground-truth. Some of them are listed below:

- Perform a certain, measured pose, and compare the adopted joint angles and those provided by the vision-based HMC system. The main drawbacks this solution faces are the complexity of obtaining precise human pose measurements and that it is only valid for static poses.
- Use a database of upper-body gestures, captured using stereo vision, that provides a pose ground-truth for each recorded frame. If the stereo images are feed as inputs for the proposed HMC system, the comparison between the resulting pose and the provided ground-truth can be useful to characterize the system errors. After search for useful alternatives, no gesture database has been found that is suitable to evaluate the proposed HMC system.

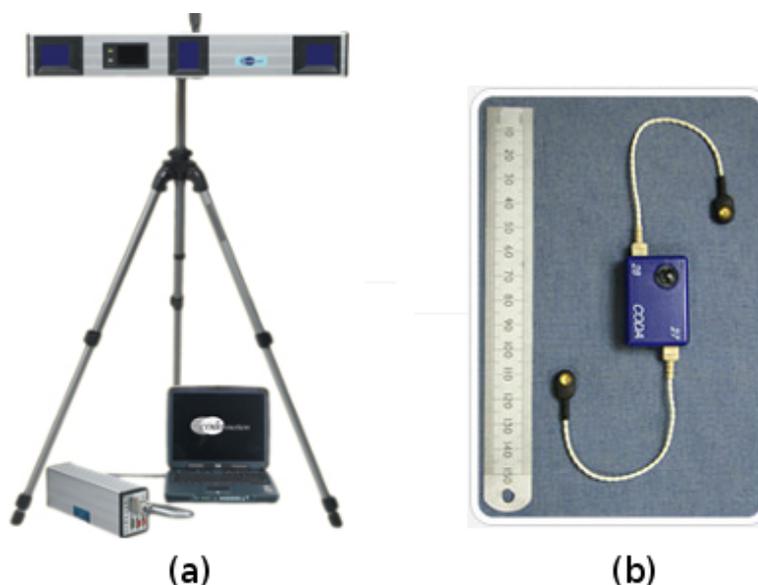


Figure 3.40: Codamotion CX1 motion capture system: a) CX1 camera unit; and b) Codamotion infra-red markers and one drive box used to power them.

- Use both proposed vision-based system and a different motion capture system to capture the upper-body gestures. The second system must be precise enough as to allow its results be used as ground-truth. Mechanical systems meet these requirements, but they tend to be cumbersome, and require a complex calibration process. Capture systems based on markers are also subject to complex initialization processes and requires to wear specific suits and/or markers. These systems are expensive to buy or hire. On the other hand, this strategy allows to freely tune the parameters of the tests in order to correctly characterize system errors.

In this thesis, the ground-truth has been obtained by choosing the third option. A Codamotion CX1 motion capture system based on active markers<sup>5</sup> was used to obtain this ground-truth. This motion capture system is based on infra-red optical markers (Fig. 3.40.b) that are attached to the body. Infrared light emitted by the markers is captured by one or more CX1 camera units (Fig. 3.40.a). The resolution of the system depends on the visibility of the marker and the angle between the marker and the camera unit. Normal incidence, optimal visibility and correct placement conditions (the distance from the CX1 unit to the performer should be about 2 meters) guarantee a position error below 1.5 mm.

Each CX1 camera unit has a viewing angle of about 80 degrees, and thus one single unit should be able to capture all markers attached to the performer if the optimum distance

<sup>5</sup><http://www.codamotion.com/>

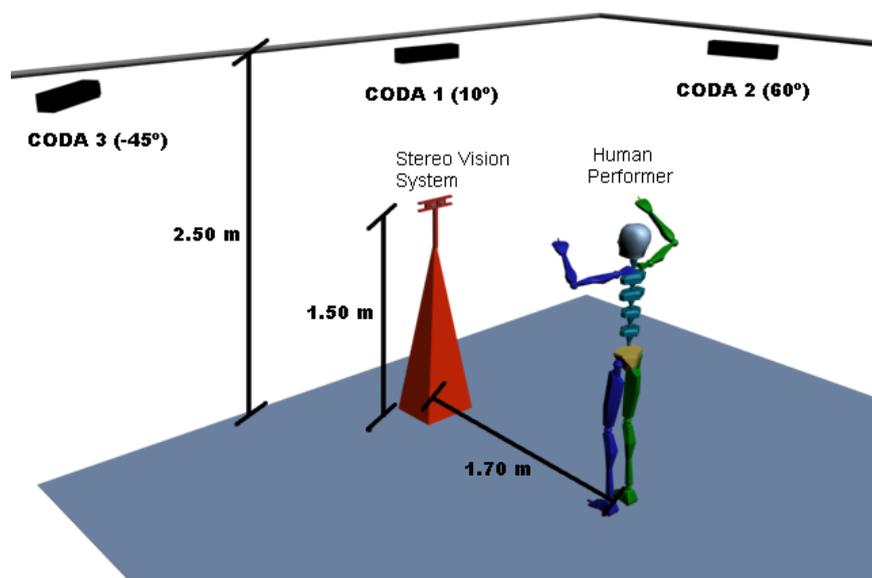


Figure 3.41: Experimental setup used to evaluate the vision-based HMC system.

-2 meters- is considered. However, two issues prevent us from using one single CX1 unit: i) occlusions; and ii) not normal incidence. While the first problem involves losing markers in certain frames, the second represents a more complex situation in which the marker is located, but its position error may be very high. In order to avoid these issues three CX1 camera units have been finally used, as depicted in Fig. 3.41. This setup minimizes marker occlusions and reduces the average position error to 5 mm. for the performed gestures.

Once the CX1 units were installed, the lighting conditions had to be tuned as the Codamotion system is interfered by light sources, while the vision-based HMC requires light to perceive human motion. The first experiments showed that, as expected, it was not possible to use the proposed vision-based HMC system in the low lighting conditions required to use the Codamotion CX1 system. Finally, an intermediate solution was adopted and only few low intensity light sources were used. As the environment was controlled for these tests, it was possible to relax skin color detection thresholds in order to achieve skin color detection and tracking under these conditions (Fig. 3.42).

The location of the markers attached to the human was selected in order to provide robust upper-body motion data. No information about finger movements, face expressions or other detailed motion is required. Fig. 3.43.b shows the position of the thirteen markers that were finally used. These positions are also listed below:



Figure 3.42: a) Left frame of a sequence captured under normal indoor lighting conditions; and b) Left frame captured during system evaluation.

- One marker attached to the waist to provide a reference for the rest of the markers.
- Two markers around the head, to capture head movements.
- One marker on each shoulder to provide information about torso bending and rotation.
- Two markers at each elbow.
- Two markers at each wrist.

As depicted, elbows and wrists are equipped with two markers, as these body parts are more sensitive to occlusions. In order to be able to compare real and perceived pose, the 3D human model used to track upper-body motion was also equipped with virtual markers attached to the same body locations that real markers (Fig. 3.44.a). Then, once the offset between the real and virtual markers is manually corrected, errors can be measured by computing the distance between real and virtual markers, as Fig. 3.44.b depicts.

Thus, the performer's motion was recorded using both three CODA units at a sampling rate of 25 Hz, and the proposed vision-based HMC system, that processes about 15 frames per second. As both systems were not synchronized, the results obtained from the stereo vision system were interpolated to offer the estimated pose corresponding to each recorded CODA sample. The waist marker was used as a reference to locate the waist of the human virtual model, thus real and virtual markers can be compared as depicted in Fig. 3.44.b.

### 3.7.2 Error measurement

The system was set up following the previously detailed steps. Then, different people were told to wear the upper-body markers and perform different movements while both CODA motion

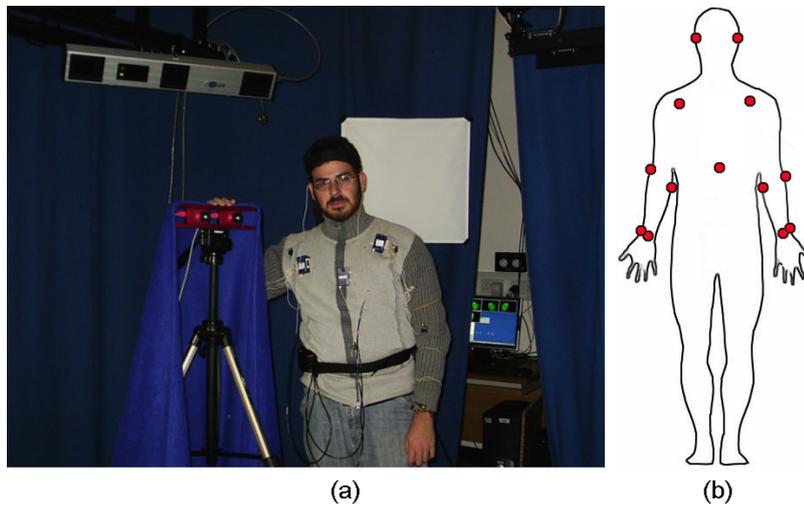


Figure 3.43: a) the stereo vision system STH-DCSG-VARX from Videre Design and the CODA motion capture system at the Centre for Vision, Speech and Signal Processing, at the University of Surrey; and b) distribution of markers.

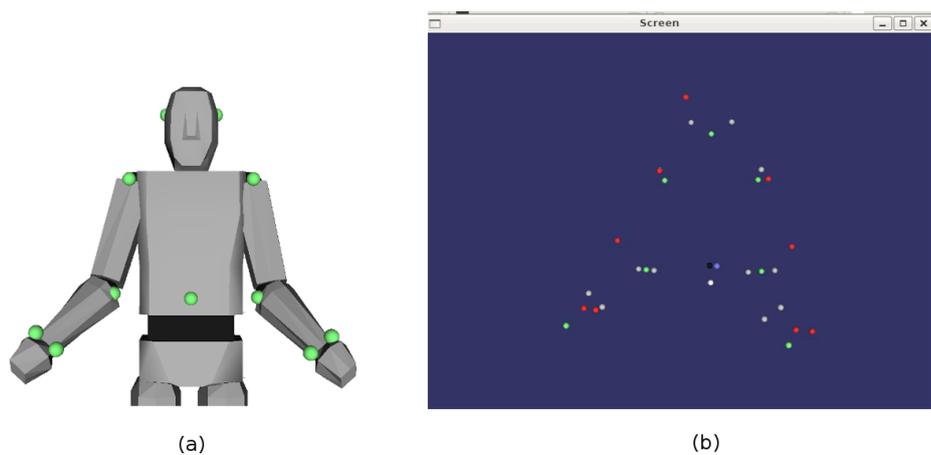


Figure 3.44: a) Upper-body human model with attached virtual markers (green spheres); and b) real (red spheres) and virtual (grey spheres) markers for a perceived frame.

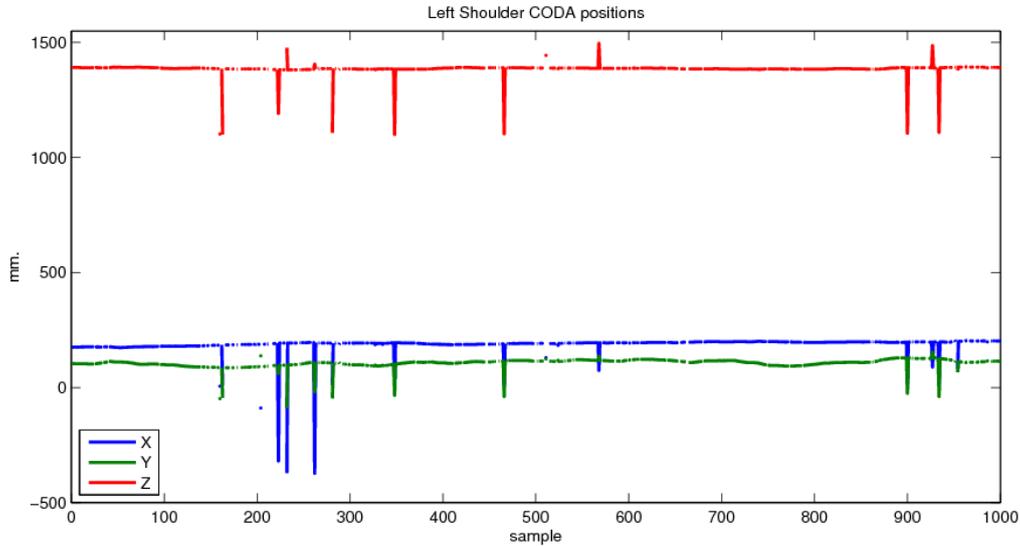


Figure 3.45: Positions of the CODA marker located in the left shoulder during a test movement (handshake) lasting 40 seconds.

capture system and the proposed vision-based HMC system recorded his/her movements. These performed movements did not need to be meaningful, but they spread over all the field of view of the stereo cameras in order to obtain complete information about the errors, depending on distance and relative position of the tracked items.

### 3.7.2.1 Ground-truth extraction

Figures 3.45 and 3.47 show the XYZ positions of two CODA markers captured at a sampling rate of 25 Hz. during one of the tests. In this case, the performer was told to slowly move her right hand forward and wave it up and down mimicking a 'shake hands' gesture. The XYZ positions depicted in Fig. 3.45 corresponds to the CODA marker located on the left shoulder, while Fig. 3.47 shows the positions of one of the CODA markers located at the right wrist. The general characteristics of the movement can be clearly perceived in Figs. 3.45 and 3.47: the left shoulder is nearly static while the hand marker firstly moves forward (the Y component plotted in green decreases in frames 300 to 400) and then it waves up and down (oscillations in Z component, plotted in red, from frame 500 to 750 approximately). It can be seen that these oscillations are also projected to the other components). After this wave movement, the right hand returns backward (frames 750 to 900). The end of the sequence registers a new right hand movement that corresponds to the next performed gesture.

The main concern about these data captured by the Codamotion system is the high

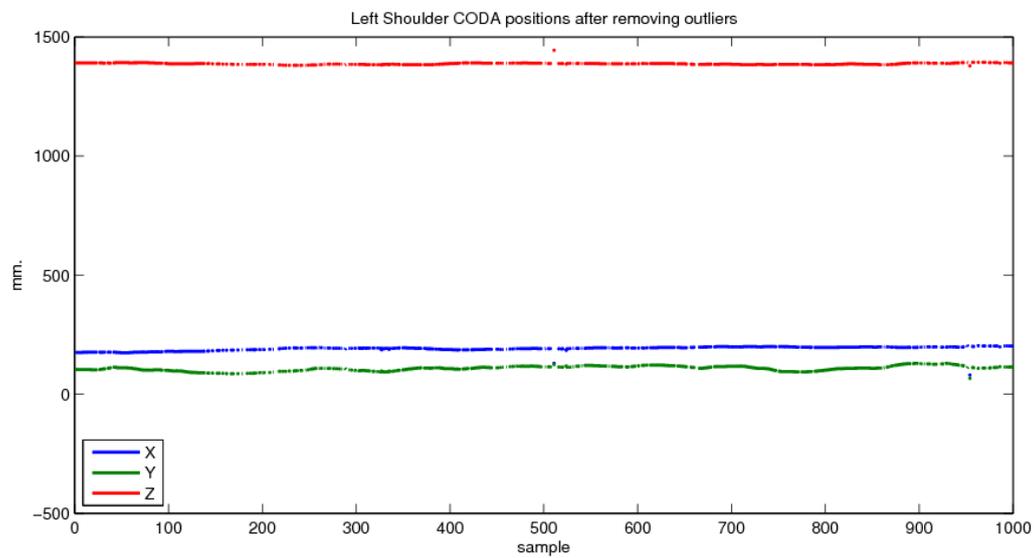


Figure 3.46: Results obtained after applying outlier removal to the movement shown in Fig. 3.45.

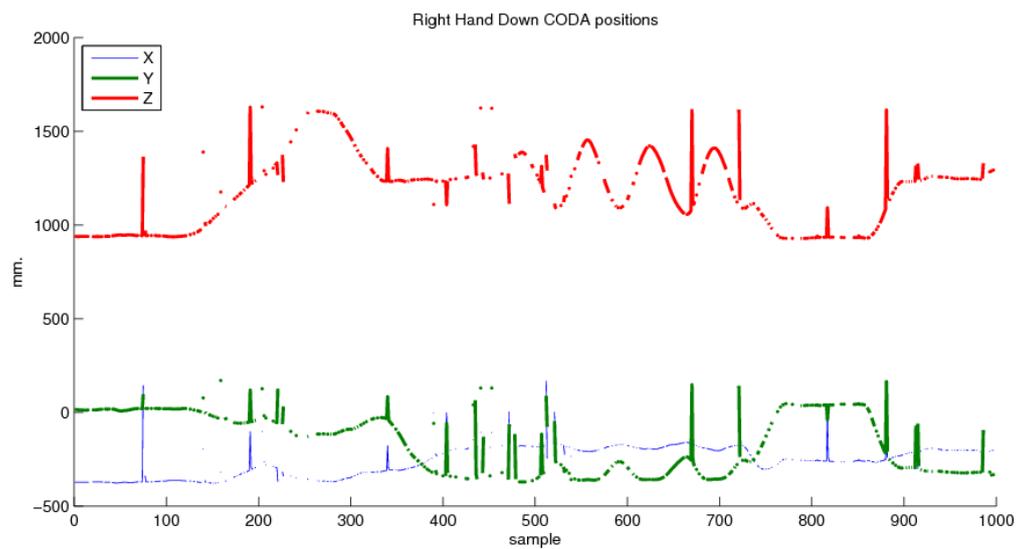


Figure 3.47: Positions of one of the CODA markers located in the right wrist for the movement depicted in 3.45.

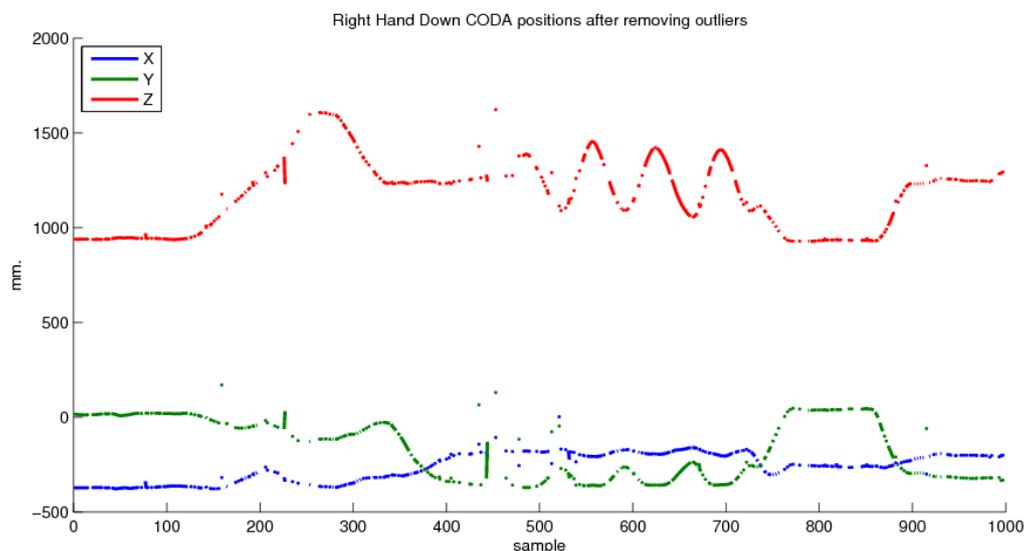


Figure 3.48: Results obtained after applying outlier removal to the movement shown in Fig. 3.47.

percentage of outliers. As commented above, it was necessary to use a soft light in order to allow the stereo system capture the motion. This light, however, introduces interferences and reflections in the infrared signals captured by the CX1 CODA units, drastically increasing the amount of outliers.

Limited visibility is the other issue that affects the captured motion. CODA markers, specially those attached to the elbows or the wrists, are frequently occluded. There are different types of occlusion, that should be addressed in different ways:

- Isolated occlusions do not represent a problem for this evaluation framework. These samples can just be ignored.
- Prolonged occlusions that are produced by an object or body part moving -or staying- between the marker and the CX1 unit can also be easily handled. As before, occluded samples are simply not used in the evaluation process.
- Finally, prolonged occlusions that are related to 'borderline' situations represent a different issue that can induce important errors. These occlusions affect markers that are nearly completely occluded, or that are nearly parallel to one of the CX1 units, and not visible to the rest.

In these cases, the affected marker is usually not perceived, or the uncertainty in the measure is so high that the Codamotion software neglects the capture. Again, in these

frames the occluded marker can just be ignored. However, noisy captures and, specially, reflections, can also offer incorrect lectures for these markers in certain frames, specially in the lighting conditions in which these tests had to be performed. These incorrect captures represent outliers for which detection becomes a more complex task as there are few, if any, correct nearby captures that can help filtering them. Samples from 400 to 500 in Fig. 3.47 are a good example of one of these situations.

These outliers should be filtered before using the captured data as ground-truth to evaluate the system. There are different methods that can be applied to filter outliers, being one of the most used the RANSAC algorithm proposed by Fischler and Bolles (1981). This algorithm is an iterative method that estimates parameters of a mathematical model from a set of observed data that contains outliers. While RANSAC offers very good results if its parameters are correctly tuned, it requires to know the model of observed data.

It is difficult to find a model that fits the complete XYZ trajectories captured by the CX1 units for a certain gesture. Thus, these trajectories should be divided in smaller segments, that can be modelled using simpler curves. RANSAC algorithm could then be applied to these segments. This solution is valid if the outliers are isolated. However, as commented above and depicted in Fig. 3.47, captured positions of the CODA markers can accumulate many outliers in certain zones of the trajectory. RANSAC algorithm will not produce good results for these segments, in which the amount of outliers may even be superior to the amount of inliers. Besides, RANSAC requires the setting of problem-specific thresholds that should be tuned for each segment. These issues prevent us from using this algorithm. Other approaches, such as the Chauvenet's criterion (Taylor, 1997), do not require the use of a model, but they are designed to deal with isolated outliers and employ fixed thresholds, that may not be suitable in our case due to the nature of the captured trajectories, which amplitude can vary very much from one part of the gesture to another.

In order to filter not only isolated outliers but also the groups of outliers that appear in the perceived positions, an algorithm is proposed that uses information about previous samples in order to filter the current one. This algorithm processes XYZ trajectories and thus it has to be executed independently for each captured marker. The algorithm is based on the assumption that the distance between consecutive XYZ positions is constrained. The continuous human trajectories captured by the Codamotion system meet this condition.

The proposed algorithm is detailed below, where  $N$  is the number of trajectory positions,  $\overrightarrow{p_e(j)}$  is the  $j$ -th trajectory position,  $n$  is the number of previously stored positions,  $\overrightarrow{P(i)}$  is the

$i$ -th stored position and  $\sigma_o$  is the distance threshold.

- Initialization:  $\sigma_o$  is initialized to a minimum value,  $\sigma_{om}$ , and  $\overrightarrow{P(i)}$ ,  $i \in [1..n]$  are initialized to the first  $n$  XYZ trajectory positions. The algorithm considers that none of these first  $n$  positions is an outlier. Thus, this initialization phase required the human to supervise the first frames of the sequences.

After initialization, the following steps are executed for  $j = n + 1$  to  $j = N$ .

1. The average XYZ value of stored positions,  $\overrightarrow{p_m}$ , is computed as follows:

$$\overrightarrow{p_m} = \frac{\sum_{i=1}^n \overrightarrow{P(i)}}{n} \quad (3.12)$$

2. The evaluated sample  $\overrightarrow{p_e(j)}$  is marked as outlier if  $|\overrightarrow{p_e(j)} - \overrightarrow{p_m}| > \sigma_o$ .
3. If  $\overrightarrow{p_e(j)}$  is an outlier, then  $\sigma_o$  is updated as follows:

$$\sigma_o = \min(\sigma_o + \Delta_o, \sigma_{oM}) \quad (3.13)$$

where  $\Delta_o$  is a fixed value and  $\sigma_{oM}$  is the maximum value for the distance threshold.

4. If  $\overrightarrow{p_e(j)}$  is not an outlier, then the set of  $n$  previous positions is updated by discarding  $\overrightarrow{P(1)}$  and including  $\overrightarrow{p_e(j)}$  as  $\overrightarrow{P(n)}$ . The value of  $\sigma_o$  is also updated as follows:

$$\sigma_o = \max(\sigma_o - \Delta_o, \sigma_{om}) \quad (3.14)$$

It can be seen that this algorithm detects outliers as samples that deviate too much from the average sample value, as other approaches such as the classical Chauvenet or Peirce's criterion (Peirce, 1852). But the proposed algorithm differs from these approaches in that it does not use global measures, but it processes samples locally, by keeping a record of the most recent valid samples. Besides, it uses an adaptive distance threshold instead of a fixed deviation value. Figs. 3.46 and 3.48 show the results obtained after applying the previous algorithm to the trajectories depicted in Figs. 3.45 and 3.47, respectively. The parameters used for all sequences captured using the Codamotion system were the following:  $\sigma_{om} = 150mm.$ ,  $\sigma_{oM} = 250mm.$ ,  $\Delta_o = 50mm.$ ,  $n = 5$ . It can be seen that the use of these values allows removing most of the outliers in both trajectories, even when they are very different (the trajectory of the marker located in the left shoulder experiments nearly no movements while the marker located in the right hand moves widely).

### 3.7.2.2 Alignment and comparison of captured data against ground-truth

As detailed in section 3.7.1, the vision-based HMC system proposed in this thesis is evaluated by comparing its results against ground-truth values, that can be obtained by removing outliers from the trajectories captured by the Codamotion system.

However, before making this comparison it is necessary to align in time the sequences captured using the two different capture systems. In order to synchronize the two captured sequences, their starting points are aligned using a visual mark that appears when Codamotion CX1 system starts the capture process. The time value, between the frame in which this mark is present and the previous, that minimizes the error in the captured data, is selected as the moment in which Codamotion CX1 system starts to capture the motion.

On the other hand, while the Codamotion system captures data at a fixed sampling rate of 25 Hz, the proposed vision-based system provides captured pose at an average sampling rate of 15 Hz. This last value experiments considerable deviations depending on multiple factors such as the number of alternatives the HMC system has to evaluate for each frame, the number of pixels in the silhouette and/or the tracked skin color regions, the number of polygons of the virtual models, etc. It is not possible to find a direct match from samples captured using these two systems. Instead, the trajectories captured using the vision-based HMC system are linearly interpolated in order to generate intermediate samples with time stamps equal to those generated by the Codamotion system.

On the other hand, the vision-based HMC system imposes the human to be tracked to stand in front of the cameras, facing them, for the first captured frames. In order to obtain accurate measures, both cameras and human performer were aligned in the experimental scenario with floor marks that were perpendicular to the line from the cameras to the human waist. However, a perfect alignment between the human torso and the cameras was not possible. Both cameras and human torso experienced small deviations respect to the marks. These deviations produce offset errors (Fig. 3.49.a), that should be corrected before measuring capture errors. This correction can be achieved by performing a spatial alignment of the captured sequences. Thus, the virtual model that is the output of the vision-based HMC system is manually translated and rotated, so that its shoulders and waist match the positions captured by the Codamotion system in the first frames of each test sequence. The effects of this alignment process are shown in Fig. 3.49.b, where offset errors have been compensated.

Figs. 3.50 and 3.51 show the interpolated XYZ positions of the virtual markers attached

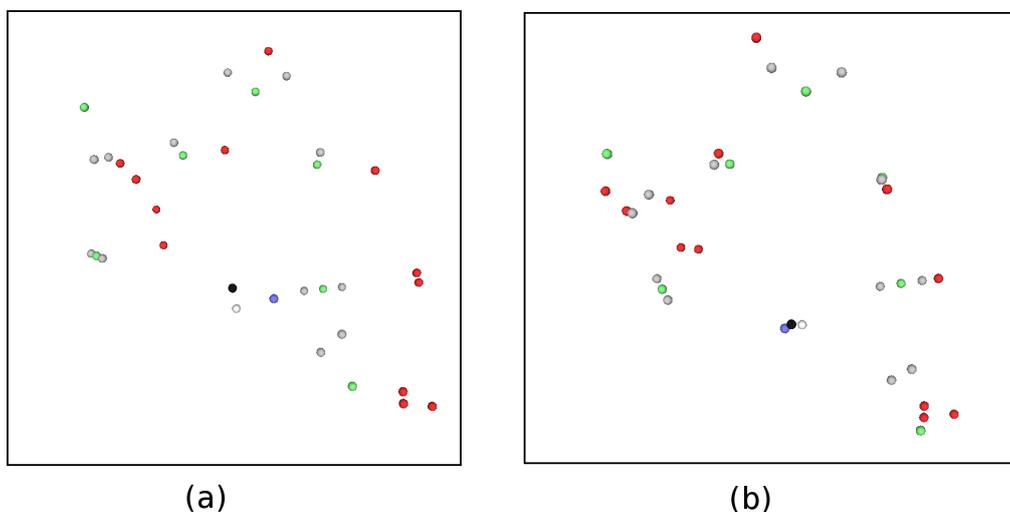


Figure 3.49: Comparison between real CODA markers (red dots) and virtual markers attached to the 3D human model (grey dots): a) before spatial alignment; and b) after spatial alignment.

to the left shoulder and right hand, respectively, for the movement in which the CX1 units recorded the data shown in Figs. 3.45 and 3.47. It can be seen that, after the temporal and spatial alignment processes, the XYZ trajectories of the virtual markers attached to the model can be compared against the ground-truth provided by the Codamotion system, depicted in Figs. 3.46 and 3.48. Then, error is measured by computing the Euclidean distance from the virtual markers attached to the 3D human model to the real CODA markers. Fig. 3.52 shows the mean errors and standard error deviations associated to each of the tracked markers. These values are averaged over the 5300 frames evaluated during the tests. As depicted, the elbows, which pose is estimated but not perceived by the stereo vision, accumulate a position error that is only slightly larger than the hands. This result shows the validity of the criteria used to pose elbows in the HMC module.

There are, however, other markers that are affected by considerable errors, i.e. the marker located in the left part of the head. While capture errors can be different depending on the evaluated marker, there is an additional factor that should be considered: the visibility of each CODA marker, that is affected by particular conditions, as the angle between the CODA marker and the CX1 units. As commented above (section 3.7.1) CODA errors are below 1.5 mm. but only in optimal conditions. Thus, markers that are not clearly visible can be affected by more significant errors. Fig. 3.53 show the visibility of each CODA marker during the tests. When the errors in Fig. 3.52 are considered together with these data, it can be concluded that visibility of CODA markers strongly affect the quality of the results. Thus, CODA markers that are hardly visible should not be included in the evaluation of capture errors, as these markers

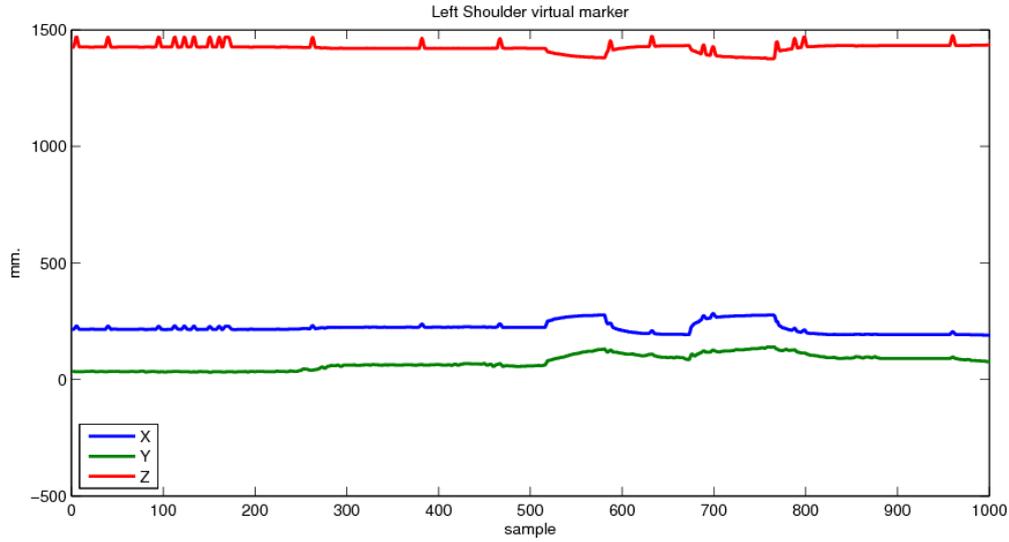


Figure 3.50: Interpolated positions of the virtual marker located in the left shoulder for the movement depicted in Fig. 3.45.

accumulate errors in the ground-truth provided by the Codamotion system. Table 3.5 shows the capture errors measured when markers for which visibility is under 40% for a certain test sequence are discarded. These values are adequate for learning the social gestures considered in this thesis.

Table 3.5: Tracking errors averaged over 5300 frames.

<b>Marker</b>	Left Shoulder	Left Elbow	Left Hand
<b>Mean Error (cm)</b>	5.74	12.53	11.51
<b>Standard Deviation (cm)</b>	3.13	6.06	6.55
<b>Marker</b>	Right Shoulder	Right Elbow	Right Hand
<b>Mean Error (cm)</b>	6.72	12.41	11.47
<b>Standard Deviation (cm)</b>	5.01	6.94	7.63
<b>Marker</b>	Left Head	Abdomen	Right Head
<b>Mean Error (cm)</b>	7.03	7.76	6.51
<b>Standard Deviation (cm)</b>	5.41	1.18	5.13

Finally, the experiments also show that the capture error is not equally distributed over the entire image, but it increases near image borders. This is illustrated in Fig. 3.54, which shows the differences between the positions estimated using both systems for two different items A and B. Item A is moving near the image center and item B is performing a similar movement near the border of the image. Although the calibration software provided by Videre design allows taking into account the radial and tangential distortions of the lens, obtained results show that

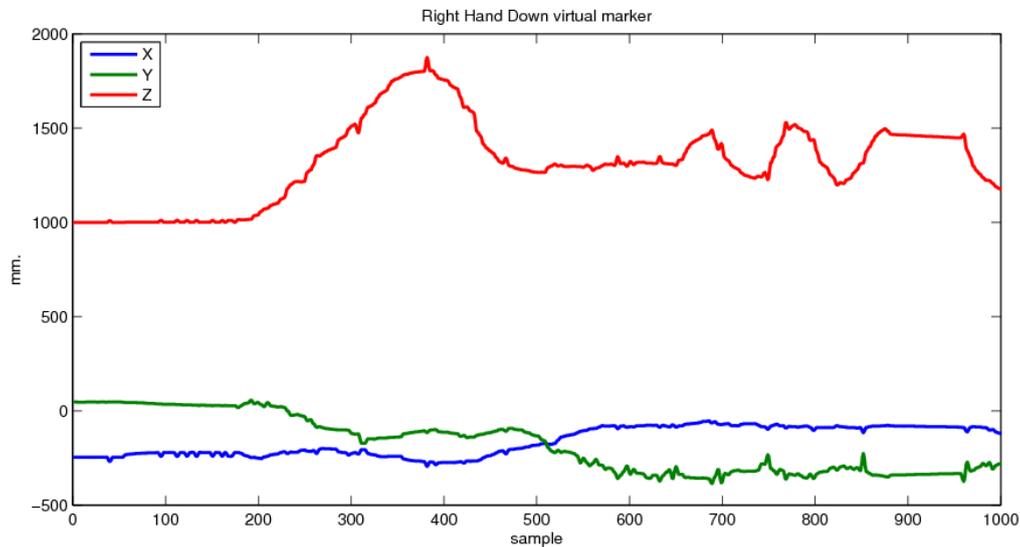


Figure 3.51: Interpolated positions of the virtual marker located in the right wrist for the movement depicted in Fig. 3.47.

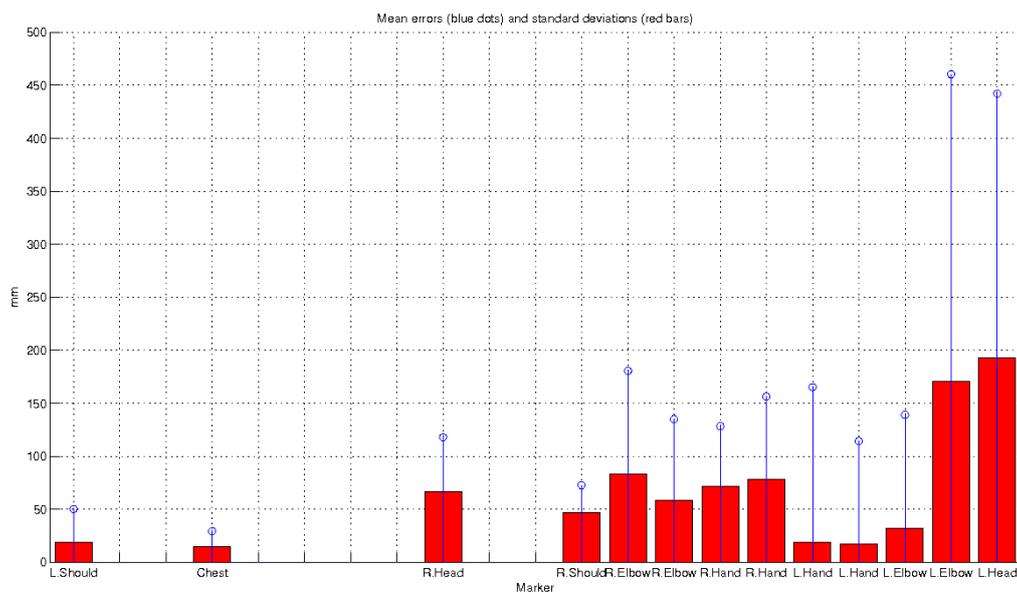


Figure 3.52: Comparison between Codamotion CX1 and the proposed vision-based HMC system. Mean errors and standard deviations associated to the different markers.

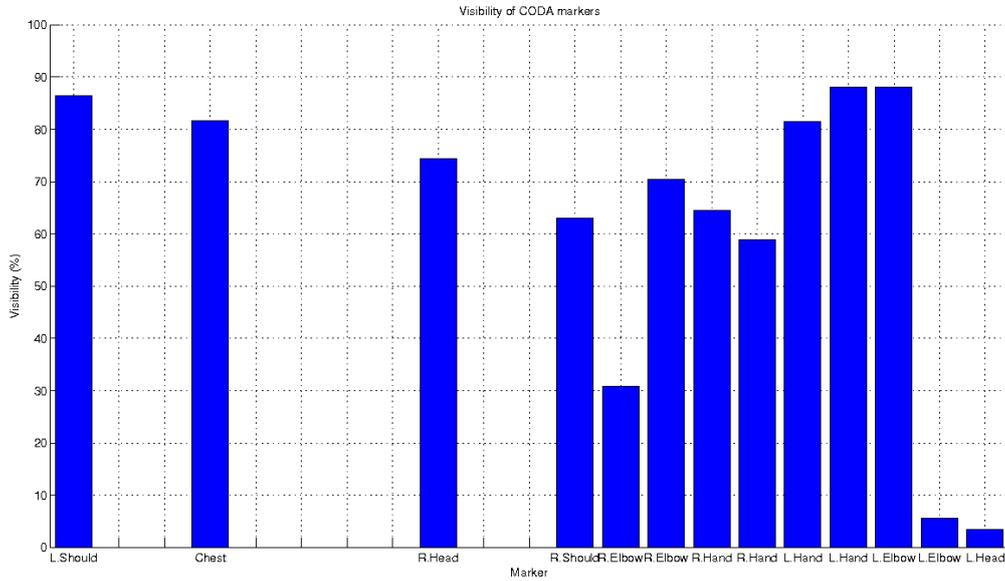


Figure 3.53: Percentage of visibility for the CODA markers during execution of test sequences.

this calibration is not exact and thus these distortions still affect to the correct estimation of the 3D position of perceived points. The pin-hole model used for the cameras and the perspective effects also introduce larger deviations near the image borders. All these factors make hand positions, that tend to approach image borders, to accumulate a higher error. It is significative that hand positions are not estimated by the proposed HMC method, but directly obtained from disparity information. As discussed in chapter 7, the main errors of the system are produced by the used stereo cameras. The HMC itself does not significantly increase these errors.

### 3.8 Conclusion

Through this chapter a performance evaluation of the proposed HMC system has been achieved. The errors that affect the captured motion have been isolated and described. In order to obtain these errors, the captured motion has been compared against ground-truth provided by a Codamotion CX1 HMC system. While this system is able to obtain very accurate XYZ positions for the markers attached to the human performer, the adverse lighting conditions that were required in order to capture the motion with both HMC systems produced an unusual high amount of outliers in the data collected using this Codamotion system. Together with the limited visibility that affected some of the markers, this factor forced to filter outliers, and even discard data, from the ground-truth before performing the capture error evaluation.

As the ground-truth is composed by the positions of the CODA markers, from which

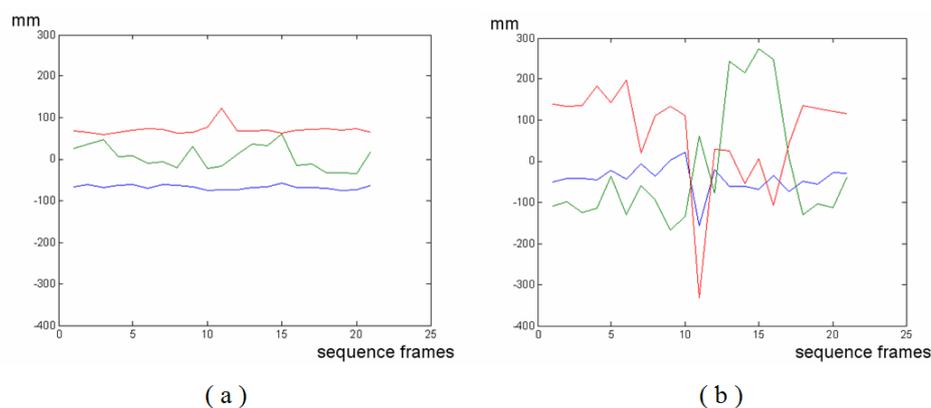


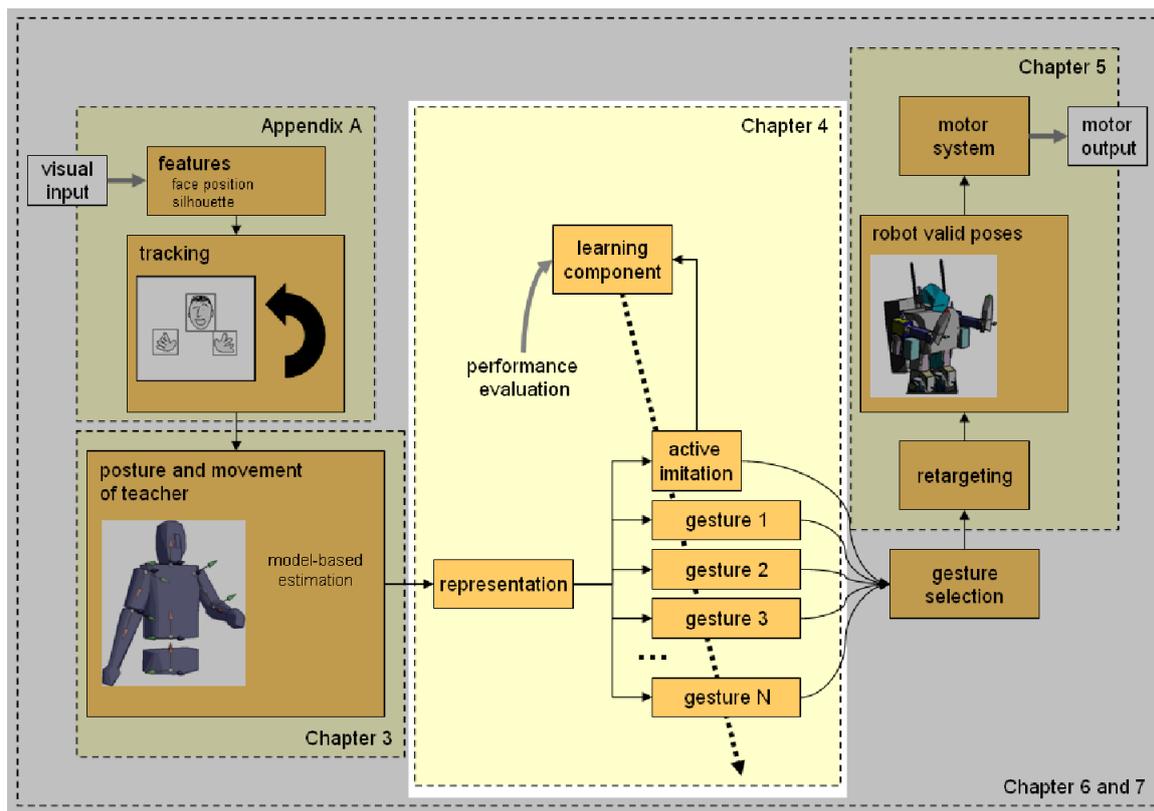
Figure 3.54: Comparison between Codamotion CX1 and the proposed vision-based HMC system. Differences between position estimated from both capture systems for a virtual marker located: a) near the image center; and b) near the image border (red, green and blue lines show the differences in the x, y and z coordinates, respectively).

it is possible to infer the pose of the performer, virtual markers were also attached to the 3D human model that performs the pose computed by the proposed vision-based HMC system. Thus, the error measurement can be achieved by computing the Euclidean distance from virtual markers to real ones. It must be noticed that, since the proposed system uses standard body proportions and only adjust the human model height to match the performer's height, virtual and real human bodies are different. Then, there is a component in the measured error that is related to the comparison between different bodies, and that could be eliminated by a more complex body initialization process.

Once the virtual markers have been set, their positions are aligned both in time and space with the positions of the CODA markers. Time alignment is achieved by linearly interpolating the virtual markers positions to obtain a measure that corresponds to the time stamp of each CODA sample. Spatial alignment is manually performed for the first frames of each test. Then, the rest of the samples are used to measure capture errors. The analysis of these measurements demonstrates that the proposed vision-based HMC system is able to provide a pose for the human performer that is accurate enough as to recognize social gestures. Capture errors are higher as the tracked item approach image borders, thus, in scenarios in which gestures were required to be captured with higher precision, the use of better cameras or lenses should be considered in order to improve capture results without modifying the proposed algorithms.

## Chapter 4

# Gesture representation, recognition and learning



### 4.1 Outline of the chapter

Previous chapters described the vision-based perceptual system and the model-based Human Motion Capture (HMC) system used in this thesis. These systems allow estimation of human

pose from limited perceived cues such as hands and head positions, and silhouette data. While the sequence of perceived poses may be directly imitated by the robot through the use of the retargeting module, a system that is able only to imitate perceived poses –a passive imitator, as defined by Demiris and Hayes (2002)– is far from providing the abilities required in Robot Learning by Imitation (RLbI) scenarios. Thus, a robot that is intended to relate to humans and learn from them must be able not only to imitate, but also to memorize, recognize and reproduce human gestures, as commented in chapter 1.

Many different systems have been proposed to provide a robot with these capabilities. However, they face major issues when applied to noisy, dynamic real scenarios in which the system has to deal *on-line* with limited perception, untrained human users, constrained initialization phases and few or none *a priori* training samples. Thus, a large amount of research is currently being performed in this particular topic. The approach proposed in this thesis to gesture representation, recognition and learning integrates already proposed ideas and novel contributions into a complete scheme that is detailed in this chapter, following these sections:

- Section 4.2 describes the current state of the art in gesture recognition and learning. Different approaches are presented, and some of their advantages and drawbacks are exposed.
- Section 4.4 completes the previous description by analyzing in detail one of the main issues a gesture recognition system must face: avoid the *curse of dimensionality* (Bellman, 1957).
- Section 4.5 presents an overview of the proposed approach, that is deeply explained in sections 4.6, 4.7 and 4.8.
- Finally, section 4.9 compares a traditional classification system, based on the combined use of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), with the proposed approach.

## 4.2 State of the art

As detailed in chapter 1, the evolution of the robotic agents towards *social robots*, that operate in real human environments and engage people and other robots in social interactions, demands the development of advanced perceptual systems that allow the robot to sense and interpret the same phenomena that humans observe (Dautenhahn and Nehaniv, 2002). In addition to verbal communication, facial expressions or context information, gestures and body language should be perceived, recognized and learned by a robot that interacts with humans. Calinon (2007)

points that such a robot, that is expected not only to interact, but also to physically cooperate with humans, must be predictable and behave human-likely regarding social interaction, gestures or learning behaviour. Thus, robots that are designed to participate in RLBI scenarios should be equipped with a gesture recognition system that takes into account not only the robot characteristics, but also the previous social considerations. This system should allow the robot to learn new gestures from observation, recognize them at human interaction rates and adapt them to new scenarios (Calinon, 2007).

### 4.2.1 Gesture representation

Given a perceived movement, it must firstly be represented in an adequate format to be analyzed, recognized and learned. Different symbolic representations have been proposed to help encoding behaviours in an efficient way to allow generalization (Demiris and Hayes, 2002; Nicolescu and Matarić, 2003; Breazeal et al., 2005; Saunders et al., 2006; Alissandrakis et al., 2007). Thus, inspired by the Active Intermodal Mapping (AIM) paradigm of Meltzoff and Moore (1989), Breazeal et al. (2005) propose to represent gestures into the robot *joint space* using a directed weighted graph, known as the *posegraph* (Downie, 2000). In this graph each node is an annotated configuration of the robot joints, or pose, and a gesture is understood as a certain path through the leaves of the graph. In Demiris and Hayes (2002), an alphabet of predefined poses is also used, and a gesture is defined as a transition between them. Alissandrakis et al. (2007) use a model-based approach in which knowledge is composed by states and actions. States contain positions and orientations corresponding to static poses, while actions represent transitions between states. A gesture is then represented and stored as a sequence of actions. Different subgoal granularity can be applied in order to obtain different symbolic representations. Saunders et al. (2006) rely on pre-defined behaviours to make a wheeled robot move through a maze, and use this symbolic representation to explore the scaffolding issue in the teaching process and organize tasks in hierarchical frameworks. Nicolescu and Matarić (2003) propose a high level network architecture composed by *abstract behaviours*. The links between behaviours represent precondition-postcondition dependencies, which can also have three different types: *permanent*, *enabling* and *ordering*. The activation of a certain behaviour depends both on its own preconditions and the postconditions of its relevant predecessors. Generalization is achieved at topological level, as links are incrementally updated by computing common subsequences and finding alternate paths to perform skills.

As stated by Schaal (1999), these symbolic representations ease interactive learning, but they rely on pre-determination of the observed cues and the efficiency of the segmentation

process, thus restricting the possible movement repertoire. Besides, it may be difficult to find the optimal granularity level to represent generic gestures. Encoding certain gestures using these symbols may also be difficult for these approaches. In this sense, Breazeal et al. (2005) propose to use different graphs for different sets of body parts (*organs*), although it may be difficult to extend this solution to systems involving many DOF (Schaal, 1999).

Another approaches to encode perceived gestures represent them as continuous streams of data. While learning becomes a more complex issue for these solutions, they are more suitable to generalize and refine stored gestures (Calinon, 2007). Thus, Ude et al. (2004) model captured body trajectories using B-spline wavelets to deal with uncertainty contained in different demonstrations performed both in *joint space* or in *task space*. This wavelet representation increases the efficiency of other spline based solutions (Gortler and Cohen, 1995), although it is still not able to offer reconstructed motion *on-line*. Other examples of *joint space* representations are proposed in Ude (1999) or Safonova et al. (2002). Ude (1999) uses a virtual model to help extracting a set of *twist coordinates* to represent perceived pose. Safonova et al. (2002) minimize an objective function to obtain for each frame a set of valid robot joint angles that correspond to captured motion. Ito et al. (2006) propose to use a Recurrent Neural Network (RNN) to learn the dynamics of the motion and switch between different behaviours depending on context information. They use *kinesthetic* learning to train the RNN and achieve some degree of generalization due to the characteristics of attractor dynamics that emerge from the RNN. Other researchers as Yamane and Nakamura (2003) propose the use of forces or torques to efficiently represent a manipulation skill. Yamane and Nakamura (2003) demonstrate the capability of their method to adapt to new environments and physical conditions. The main drawback of this approach is that parameters must be manually tuned for each task. Besides, force feedback is not always available.

An issue concerning representations based on joint angles, kinesthetic teaching or forces is that most robots are equipped with vision-based systems to perceive human gestures. Stereo vision systems are non-invasive, meet the requirements of RLbI scenarios, and are easily identifiable by people interacting with the robot. But only Cartesian 3D positions are provided by such a perception system, thus additional processing is required to construct previously mentioned representations from images. On the other hand, gesture recognition can be directly done on 3D Cartesian representations. Bandera et al. (2006) use this description into a learning system which allows the robot to recognize and learn dual-hand gestures. However, increasing the sampling ratio, the gesture length or the number of tracked body parts can lead to excessively large descriptors. The *curse of dimensionality* can be tackled by dimensionality reduction methods,

such as PCA (Jolliffe, 1986) or LWPR (Vijayakumar et al., 2005). Another option to reduce dimensionality is to define, and select for each gesture, a reduced set of features that describe the trajectories associated to different body parts. Previous works have addressed this problem of trajectory representation using *global trajectory features*, which are defined in relation to an external reference (Croitoru et al., 2005), or using *local trajectory features*, which are based on differential measures (Rodriguez et al., 2004). The main advantage of the global features is their robustness to outliers and noise. On the contrary, they face major difficulties in capturing fine details of trajectories (Alajlan et al., 2007). Local features are superior in discriminating fine details, but they are usually highly sensitive to outliers and noise.

It is important to consider a large body of work that focuses on finding the correct mapping between sensory information and motor outputs. These approaches consider this mapping as the most meaningful representation of a gesture. Some of these methods try to find a global non-linear function that models the complete gesture (Williams and Rasmussen, 1996; Smola and Scholkopf, 1998; Ghahramani and Beal, 2000). These approaches are oriented to process batch data, and are difficult to adapt to *on-line* applications. They also need to set *a priori* the right modelling biases (Vijayakumar et al., 2005). Other, more recent contributions (Lopes and Santos-Victor, 2005) begin to address these problems by using PCA to reduce input data dimensions and analytic perspective transformations and IK methods to map the perceived data into motor commands. On the other hand, methods which fit non-linear functions *locally*, using spatially localized models, are well suited for incremental fast learning, particularly in the framework of locally weighted learning (LWL) (Schaal and Atkeson, 1998). While LWL techniques may be very sensitive to the *curse of dimensionality*, recent works such as the LWPR method proposed by Vijayakumar et al. (2005) are addressing these problems by using local projections to alleviate the computational complexity, and allow fast response even in high dimensional spaces.

We finally mention the growing use of HMM as a solution to encode perceived gestures. Thus, from pioneering works such as the one of Yang et al. (1994), different authors have focused in these solutions, as HMMs ease generalization, parametric description, trajectory clustering and selection, and gesture recognition. Thus, Yang et al. (1997) used HMMs to encode the motion of a robot gripper by using either position or velocities. They considered an assembly task where the HMMs were able to generalize perceived skills over several examples. Ogawara et al. (2002) use HMMs to automatically cluster manipulation gestures demonstrated by the user, while Calinon and Billard (2005) or Asfour et al. (2006a) use HMMs as gesture descriptors that store all information needed not only to recognize gestures, but also to reproduce it. It is

important, however, to consider that HMM-based clustering is very sensitive to segmentation errors that can easily produce an inefficient or ill generalization from training data (Calinon, 2007).

### 4.2.2 Gesture recognition

Once the robot has correctly represented the perceived gesture, this encoded gesture should be compared with a set of memorized ones in order to recognize or learn it. This matching stage must take into account the unique characteristics of 3D trajectory data, such as different sampling rates, outliers, or different sequence lengths (Croitoru et al., 2005). Currently, Hidden Markov Model (HMM)s can be considered as the state-of-art modelling scheme used in gesture recognition. They provide a robust and accurate framework which has been employed in previous related works. Thus, Lee and Xu (1996) used HMMs to recognize hand gestures from the sign language alphabet. They use a data glove to provide accurate information about hand pose. Nam and Wohn (1996) proposed a HMM-based system to recognize hand posture and motion using vision. Wilson and Bobick (1999) use parametric HMMs for gesture recognition in *off-line*. The contributions of Park et al. (2005), Calinon and Billard (2005) or Asfour et al. (2006a) are good examples of the use of HMMs to recognize upper-body gestures and control a humanoid robot through vision or kinesthetic teaching. Kojo et al. (2006) choose an extension of HMMs, the Proto-symbol space proposed by Inamura et al. (2004), to represent gestures, measure the degree of similarity between them and detect unknown gestures. They apply their algorithm to recognize upper-body gestures using only face detection and stereo data. Inspired by (Ogawara et al., 2002), Aleotti and Caselli (2006) employ a simpler distance-based algorithm to cluster perceived gestures, but they also use a HMM in the recognition stage.

There are, however, several shortcomings which must be taken into account when using HMMs. Thus, the time complexity of training and inference algorithms limits the number of states that can be used in the model. This constraints the number, length and precision of the gestures that can be modeled using HMMs. Besides, these HMM-based approaches need a huge amount of training data to work well. In some scenarios, such as the ones considered in this thesis, it may be difficult to provide the system with many *a priori* samples of a gesture. To reduce these training requirements, Rajko et al. (2007) introduce the Semantic Network Model (SNM), an extension of *hierarchical* HMMs (HHMMs) that uses factorization of state transition probabilities to increase the efficiency of gesture recognition. SNM also includes the notion of *semantic states* to mark parts of the model that carry semantic meaning, and that are useful to locate the beginning and end of a certain gesture, or other non-hierarchical semantic

structures such as the mid-point of a gesture (Rajko et al., 2007). The results presented in Rajko et al. (2007) are promising, although this approach has still to be tested in scenarios involving complex observation vectors or *on-line* requirements.

Other approaches which are commonly used to match 3D trajectories use Dynamic Programming (DP) (Croitoru et al., 2005; Chen et al., 2005). These methods solve the problem by comparing the unclassified sequence of observations with a known sequence or training sample. In this context, Dynamic Time Warping (DTW) is able to accommodate elastic matching, and therefore, it has been suggested as a flexible mean to calculate distances between sequences of symbols that can have different lengths. Recent gesture recognition schemes have incorporated DTW to deal with time shifting (Bandera et al., 2006; Calinon, 2007; Mülihg et al., 2009). There are other Dynamic Programming (DP) techniques that share some of the benefits of DTW but presents different characteristics (Chen et al., 2005). One of the most recent ones is the Longest Common Subsequences (LCSS), that has been suggested as a more robust distance function, due to its ability to allow for mismatches and its ability to allow a more efficient approximate computation (Croitoru et al., 2005).

Finally, it is worthy to consider other classification algorithms that may be used to recognize different gestures. Thus, Ardizzone et al. (2000) use a set of Support Vector Machine (SVM)s to recognize simple upper-body poses (Ardizzone et al., 2000). Each of these SVM is trained for one of the particular gestures the authors use to command a mobile robot. Inside the PbD paradigm, SVM has also been used to recognize dynamic hand grasps through the use of a glove based input device (Zöllner et al., 2002). More classic classifiers, such as PCA or Linear Discriminant Analysis (LDA) (Fukunaga, 1990), can also be used to recognize performed gestures. When compared to other classification techniques, LDA allows a more efficient discrimination (Pang et al., 2005). Thus, although LDA faces difficulties in classifying complex, highly dimensional data, it may become a fast and simple method to classify reduced representations of trajectory data. The introduction of recent implementations of these algorithms, that allow for incremental learning (Pang et al., 2005; Ozawa et al., 2008), makes them suitable to be used in RLbI scenarios.

### 4.2.3 Gesture reconstruction and learning

The discovery of *mirror neurons* (Rizzolatti et al., 1996; Gallese and Goldman, 1998) in both monkey and human brains became a notable topic in brain sciences and provided a new perspective to study human behavioral learning. These neurons located in the F5 area of monkey

premotor cortex discharge both when the primate is perceiving a movement and when it is performing it. In humans, mirror neurons are located near Broca's area, which has a close relationship with language management. Rizzolatti et al. (1996), following the Mac Neilage's theory (MacNeilage, 1998), stated that this relation between F5 area and Broca's area can be explained by considering that speech has its origin not in primate vocal calls, but in the use of communicative gestures. Thus, the use of the same structures to both recognize and execute an action is in the basis of behaviour learning. This biological evidence agreed with the *Mimesis Theory* of Donald (1991), who said that symbol manipulation and communicative ability are founded upon behaviour imitation, which is integration of behaviour recognition and generation.

Inspired by this role of mirror neurons in behaviour learning, different authors in the field of robotics focused on finding structures that were able not only to recognize gestures, but also to reproduce them. HMMs appear as one of these structures. Thus, Brand and Hertzmann (2000) suggest to use HMMs to identify common elements in a motion and synthesize new motion by combining and blending them. The *Mimesis Model* of Inamura et al. (2004) allows to transfer motion from human to humanoid, and blend different motions, using a HMM-based representation space. In this space each symbol can be represented as a HMM, for which states are described by density functions modelling a set of observed features, such as trajectory key-points (Asfour et al., 2006a). Gestures are reconstructed from these symbols as generalized trajectories obtained by following these steps (Calinon and Billard, 2005): (i) generate multiple sequences of states using the transition probabilities of the HMM; (ii) obtain, for each sequence of states, the corresponding density functions, that can be used to generate multiple trajectories stochastically; (iii) retrieve a generalized trajectory by averaging a large amount of sequences. As pointed out by Calinon (2007), the main drawbacks of this approach are its computational complexity and that averaged trajectories are not smooth enough as to be used directly in a robot controller. Thus, this author proposes to use instead *Gaussian Mixture Models* to encode perceived motion, and *Gaussian Mixture Regression* to retrieve generalized trajectories from these models in a probabilistic and continuous form.

The Active Intermodal Mapping (AIM) hypothesis (Meltzoff and Moore, 1989) postulates that human infants use the demonstrator's states as against which to direct own body states, perceived proprioceptively. The AIM paradigm implies that there is a matching, or a common representation, for both perceived and executed actions, thus it also agrees with the Mimesis Theory. Inspired by these biological evidences, Demiris and Hayes (2002) or Demiris and Khadhouri (2005) have also proposed a model-based architecture in which different behaviours are matched in parallel against the perceived one and motion is generated from them

using forward models and combining the resulting outputs. They test this architecture over a dynamic simulator. [Breazeal et al. \(2005\)](#) use this same approach in a more complex scenario that involves a real robot and not simple one-to-one correspondences between tracked features and imitator's joints. These correspondences are learned through experience. As ([Breazeal et al., 2005](#)), [Lopes and Santos-Victor \(2005\)](#) map movements as joint motion, thus reproduction is directly achieved. However, they focus only on imitation of static postures and use a set of predefined gestures. The first architecture developed in the framework of this thesis, presented in section 2.5, followed these works and extended them to complex movements, where the temporal chaining of elementary postures are taken into account. Incremental learning were also considered as new gestures could be detected and incorporated to the system. As commented in chapter 2, this first architecture showed the strong limitations that physical differences, between the robot and the human, introduced in RLbI schemes in which perception and action were represented in the robot motion space. Mimesis Theory suggests the existence of such representations in humans, but as pointed out by [Meltzoff and Moore \(1989\)](#), or [Mosterín \(2005\)](#), a high degree of similarity between demonstrator and imitator -they should be individuals of the same kind- is required if this approach is considered. The architecture presented in this thesis has been designed as a generic system, able to be easily adapted to different social robots, that may be significantly different to the human performer. Thus, the inner representation of human gestures is translated to the human motion space, using a human model to achieve it.

Finally, [Mayer et al. \(2007\)](#) use principles known from fluid dynamics to reproduce a gesture at a trajectory level. While this approach allows for smooth and efficient generalized trajectories, its complexity makes it difficult to apply to on-line, interactive RLbI scenarios.

### 4.3 Gesture segmentation

Before considering the process of each particular gesture, it is important to remark that the human motion is perceived by the robot stereo cameras as a continuous sequence of frames. The first step towards recognizing gestures is to split this sequence into discrete gestures. [Kojo et al. \(2006\)](#) use a certain temporal window that is correlated against the perceived sequence. This solution needs additional computation in order to adapt to non-uniform gesture performance times, and it is complex to use it at human interaction rates.

Other approaches use a simpler criterion to segment gestures from perceived sequences ([Lee and Xu, 1996](#); [Calinon and Billard, 2005](#)). These solutions use a short pause in the motion to signal the start and the end of a gesture. Each gesture is then composed by a certain motion

limited by two static poses. In the system proposed in this thesis this criterion is adopted, thus the amount of movement perceived for different body parts is considered. Equations 4.1 and 4.2 are respectively proposed to detect the starting and ending points of each gesture.

$$\exists \|\overrightarrow{p_n(t + 0.2 \cdot \Delta_t)} - \overrightarrow{p_n(t)}\| > \sigma_{mov} \quad n \in [1 \dots N] \quad (4.1)$$

$$\|\overrightarrow{p_n(t + 5.0 \cdot \Delta_t)} - \overrightarrow{p_n(t)}\| < \sigma_{mov} \quad \forall n \in [1 \dots N] \quad (4.2)$$

where  $N$  is the amount of tracked body parts used to mark the starting and ending points of a gesture (e.g. it would be interesting for a certain application to indicate these points using only the motion of the right hand).  $\overrightarrow{p_n(t)}$  is the XYZ position of body part  $n$  at instant  $t$ .  $\sigma_{mov}$  is a distance threshold.  $\Delta_t$  is a constant value that indicates the time interval employed to detect gestures. Lower values of  $\Delta_t$  may be used if gestures are performed faster, and viceversa.

As depicted, the total time interval used to detect the absence or presence of movement varies depending on the human being performing a gesture or not. Eq. 4.2 is used when the gesture is being performed to detect its ending point, thus the human needs to stand still for  $5.0 \cdot \Delta_t$  seconds after finishing the gesture to allow the system to detect its end. However, once stopped, the beginning of a new gesture is detected in only  $0.2 \cdot \Delta_t$  seconds, using Eq. 4.1. This variation in the time threshold allows for a fast and accurate detection of the starting point of a new gesture and an adequate estimation of the ending point, while reduces the false gesture ending points detections due to small pauses in the gesture performance.

Thus, once the robot detects that a person has not performed any movement for a certain time, it considers the gesture is finished, thus recognition and learning modules are executed. Then, the robot resets the gesture representation module to be ready to perceive the next gesture, that starts as soon as the human begins to significantly move again. While this strategy is not able to recognize static gestures by itself, it is easily extensible by capturing the static poses performed by the human between dynamic gestures, and including them in the process as possible gestures.

An interesting alternative to mark the beginning and end of a certain performed gesture is to use specific verbal commands (Nicolescu and Matarić, 2003). This solution is easy and effective and avoids false gesture limit detections induced by very slow movements, tracking losses or non continuous performances. While an auditive system is a very useful feature for any social robot (Breazeal et al., 2003), it lies beyond the objectives of this thesis.

## 4.4 Reduction to a latent space of human motion

If feature-based representations are used, both global and local features could be encoded in a subspace of lower dimensionality as perceived motions and motor signals, used to transfer the skill, are highly redundant (Vijayakumar et al., 2005). To search for the latent space which encapsulates the main characteristics of the tracked trajectories, several linear and nonlinear methods have been proposed. Among these methods, Principal Component Analysis (PCA) is widely used (Duda et al., 2001) as a pre-processing step to reduce the dimensionality for further analysis.

Local dimensionality reduction techniques has been also proposed to deal with the nonlinearities of the motion in latent space. However, the number of required local models grows exponentially with the number of input dimensions. This forced these techniques to evolve from memory-based to model-based representations, and from batch learning processes to incremental learning strategies. This evolution led to contributions as the work of Vijayakumar et al. (2005), where they propose the use of LWPR as a low complexity algorithm that is able to learn rapidly and to deal with a large number of redundant inputs.

Another option to reduce the dimensionality of data is to extract certain features, or signatures, from perceived trajectories. As commented above, Croitoru et al. (2005) propose to use a set of global features to encode the motion. These features are extracted in two steps: (i) the trajectories are normalized by using PCA to decompose them into eigenvalues referred to the center of mass. Additional computation to solve sign ambiguity is also addressed to ensure invariance respect to translation, rotation, reflection and scaling; (ii) the signatures are obtained as the Global Spherical Coordinates of normalized trajectories. The trajectory is then represented as a sequence of a fixed number of triplets, each one composed by two angles and a distance. While this representation is robust and able to efficiently compare different trajectories, it could be argued that an up-down gesture should not be confused with a left-right gesture, and that wide movements should not be confused with small displacements. The inclusion of additional global features could help in addressing these issues.

Local features have been extensively used to represent trajectories. In Computer Graphics and Animation it is common to represent motion using a set of key-points, and generate continuous trajectories from them by interpolation (Shreiner et al., 2005; Boardman, 2005). Huang et al. (2001) and Tang et al. (2004) use key-points to encode walking gaits for a humanoid robot. Third order splines are used to produce smooth and natural trajectories from these points. The use of these splines also helps in fulfilling human motion constraints and avoid

singularities. In the field of gesture representation, there are also several examples of extraction of key-points to encode a perceived gesture (Asfour et al., 2006a; Aleotti and Caselli, 2006). These representations rely on detecting zero-velocity crossings, joint angle variations over a certain threshold or more complex events to mark key-points. Calinon (2007) remarks different drawbacks of these approaches, that heavily rely on an efficient segmentation and require fine-tuning of the segmentation parameters. These issues, however, could be successfully addressed by a representation that manages to filter noise and preserve trajectory features (Bandera et al., 2000).

## 4.5 Proposed approach

Fig. 4.1 shows an overview of the system proposed to recognize and learn perceived gestures. As depicted, the input is the gesture performed by the user and captured using the robot perceptual system. As in RLbI scenarios robots usually rely on vision to perceive human movements, this design considers that the input is composed by a set of Cartesian 3D trajectories,  $\vec{S}_{in}$ . Each trajectory corresponds to the movement of a certain perceived body part (i.e. hands, head, shoulders...).

As commented above, a representation based on 3D Cartesian trajectories may easily be affected by the *curse of dimensionality* (Bellman, 1957). Thus, a reduction in the dimensionality of input data is required. In this thesis, the dimensionality of input data  $\vec{S}_{in}$  is reduced by extracting certain features from 3D trajectories. Gestures  $P_i$  stored in the knowledge database are also represented by using these certain features.

As pointed by Alajlan et al. (2007), it is possible to consider two different types of features, local and global, when describing perceived gestures. Each type of features presents its advantages and drawbacks. In this thesis both local and global features are used, as depicted in 4.1. Local matching provides discrimination ability, but local results are reinforced using global features, in order to increase the robustness of the results respect to outliers and noise. Thus, the set of input trajectories  $\vec{S}_{in}$  is used in two different modules that compute local and global features, respectively.

- Local features are computed as dominant points of perceived 3D trajectories. Dominant point extraction is basically based on computing the curvature function of each trajectory and selecting points in which curvature experiences high variations. While the use of standard curvature estimations has been reported to originate segmentation problems

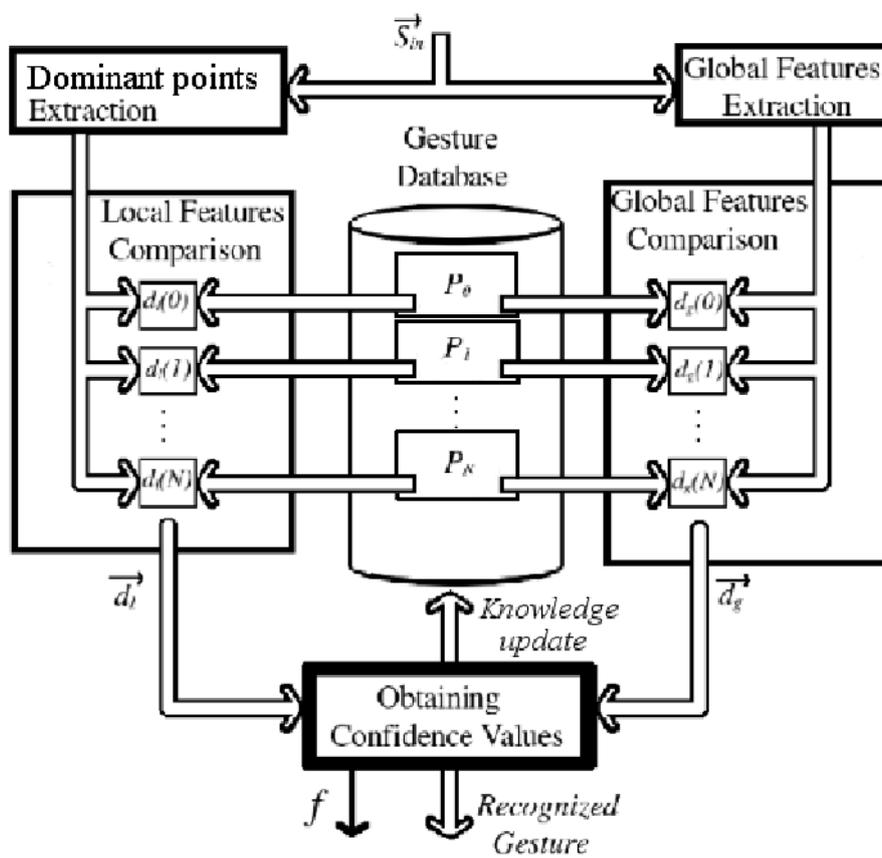


Figure 4.1: Overview of the proposed gesture recognition system.

in RLbI scenarios (Calinon, 2007), it is proposed in this thesis to use a new adaptive curvature function to alleviate these problems and provide an accurate and robust curvature estimation.

- In the proposed system global features are used not to discriminate between gestures, but to reinforce local matching. Thus only some robust and simple global cues are extracted from perceived trajectories. As detailed in section 4.7.3, these global features are measures of perceived absolute and relative amplitudes.

Once local and global features are extracted, they are used to recognize perceived gestures. While HMMs represent the current state of the art in gesture recognition, their training requirements represent an issue for systems that have to learn incrementally in interactive, on-line scenarios (Rajko et al., 2007). As commented above, the limitation in the number of states that can be implemented also becomes a drawback for a HMM-based recognizer. While a strong body of work are currently addressing these problems, this thesis presents a different approach.

Dynamic Programming (DP) alignment techniques have been widely used for matching purposes, became a standard in speech recognition and are commonly used to match 3D trajectories (Chen et al., 2005; Croitoru et al., 2005). They are not bounded by training requirements as HMM, SVM, PCA or LDA, and they can deal with time shifting, that commonly appears when matching different performed gestures. Thus, recent works have incorporated these techniques to gesture recognition systems (Calinon, 2007; Müllig et al., 2009). This thesis follows these contributions and tests different DP algorithms to perform local matching of the extracted sequences of dominant points. The results obtained using these different techniques are evaluated to select the most suitable DP algorithm. These results will be also compared with the ones obtained using different traditional classifiers in order to validate the proposed approach.

As commented above, local matching is reinforced by a factor that depends on the global features of compared gestures. This factor is computed using a fast analytic method that is explained in section 4.7.4 and that pretends to improve recognition results without significantly increasing the overall complexity of the algorithm. Section 4.7.5 discusses the effects of using global reinforcement over local distances.

Once the performed gesture has been compared with all gestures that have been memorized by the robot, the learning component of the architecture uses these results to reinforce the knowledge database. In this thesis a threshold-based method is proposed to decide whether the perceived gesture is recognized or not, and also to decide if it has to be incorporated to

the repertoire or used to update memorized gestures. This concept of *learning*, as the process of acquiring a new behaviour, follows the definitions of Demiris and Hayes (2002). Learning as used here does not imply generalization nor adaptation to different circumstances or any other processes as used in the field of machine learning. As depicted in Fig. 4.1, the algorithm returns a flag  $f$  indicating whether the performed gesture has been recognized or not. It also returns the most similar memorized gesture, as next modules in the robot control architecture may need this information.

Although it is not explicitly depicted in Fig. 4.1, it is also very important to consider the capability of the proposed system to *reproduce* a movement, as in RLBI scenarios the robot needs not only to correctly perceive human motion, but also to imitate it. As detailed in chapter 3, in our system imitation can be directly achieved from perception thanks to the use of an on-line model-based HMC algorithm and a combined retargeting system. However, in order to exhibit higher social abilities, the robot should also be able to reproduce a movement extracted not from its perception system, but from its memorized repertoire. Chapter 2 shows that this reproduction is achieved by reconstructing perceived motion from stored representation and using this reconstructed motion as the input for the retargeting system. The robot motion correctly imitates the stored one as long as this reconstruction is accurate enough. Thus, in this chapter this accuracy is evaluated by computing the distance between originally performed 3D trajectories and the ones reconstructed from the sets of local and global features that conforms the representations of the gestures.

## 4.6 3D trajectory representation

The proposed approach considers that each gesture is composed by multiple trajectories, followed by different body parts (i.e hands, shoulders or head). At the representation stage, each of these trajectories is described by a set of significant dominant points. These points are extracted from the curvature function associated to the trajectory. As significant trajectory changes may be defined at different scales, an adaptive, non-iterative approach is used to estimate the curvature function. This section describes the steps proposed to convert complete 3D trajectories into reduced sets of dominant points and validates the usage of these sets as representations of the trajectory.

### 4.6.1 Extraction of dominant points

Let a trajectory in 3D Cartesian Space be described by a sequence of points  $\{x_i, y_i, z_i\}_{i=1\dots m}$  ( $m > 3$ ), the first problem is to develop a curvature estimation algorithm which allows to filter the noise but avoids to eliminate curvature features. [Rodriguez et al. \(2004\)](#) and [Vlachos et al. \(2004\)](#) have proposed different approaches to estimate the curvature by associating a movement vector to each trajectory point. Several authors have pointed out that this vector is excessively affected by noise ([Croitoru et al., 2005](#); [Vlachos et al., 2004](#)). In order to improve the curvature estimation, the movement vector can be calculated using  $k$  points of the trajectory. However, this implies to filter the trajectory at a fixed cut frequency and only features unaffected by this process may be detected. Iterative, multiscale approaches which use several values of  $k$  have been also proposed in the context of 2D shape description. However, as pointed out by [Bandera et al. \(2000\)](#), they are slow and, in any case, they must choose the cut frequencies for each iteration.

A solution to these problems is the use of a  $k$  value which is adaptively changed according to the local information of the boundary ([Bandera et al., 2000](#)). This approach can successfully estimate the curvature of a planar shape, and similar approaches have been subsequently described ([Reche et al., 2002](#); [Marji and Siy, 2004](#)). All these approaches, however, deal with 2D contours. In this thesis the concept of adaptive curvature is extended to the representation of 3D trajectories. Several differences must be taken into account in order to perform this adaptation. Thus, 3D curvatures cannot be represented using only one angle, but need more information. In this work, 3D trajectories are projected both in XZ and YZ planes. Then, two angles are used to represent each point of the 3D curvature. On the other hand, trajectories have a starting point, an ending point, and a direction. They also last for a certain time (i.e. each point in the trajectory is associated with a certain time stamp).

The proposed method estimates the curvature at each trajectory point using the  $k$ -vicinity on both sides of the point (forward and backward). The  $k$  values are automatically changed depending on the local properties of the trajectory around the working point. Thus, for each 3D point  $i$ , the algorithm consists of the following steps:

- Calculation of the maximum length of trajectory presenting no discontinuities on the right and left sides of the working point  $i$ :  $K_f[i]$  and  $K_b[i]$ , respectively.  $K_f[i]$  is calculated by comparing the Euclidean distance from  $i$  to its  $K_f[i]$ -th neighbor ( $\|i, i + K_f[i]\|_2$ ) with the length of the trajectory between both points ( $l(i, i + K_f[i])$ ). Similarly,  $K_b[i]$  compares

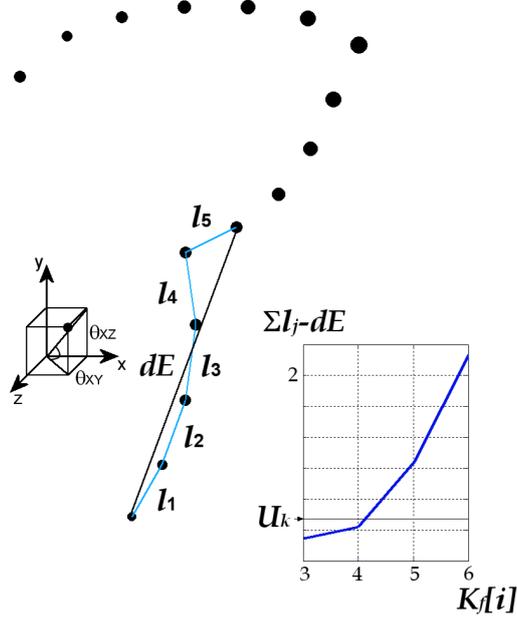


Figure 4.2: Calculation of the maximum length of trajectory presenting no significant discontinuity on the right side of point  $i$  ( $K_f[i]$ ). The graph shows different values for  $\sum l_j - dE$ , where  $\sum l_j = l(i, i + K_f[i])$  and  $dE = d(i, i + K_f[i])$ . In this case, the chosen  $K_f[i]$  value is equal to 4.0.

$(\|i, i - K_b[i]\|_2)$  with  $(l(i, i - K_b[i]))$ . The lengths of the trajectories between points,  $l(i, i + K_f[i])$  and  $l(i, i - K_b[i])$ , are defined as

$$\begin{aligned} l(i, i + K_f[i]) &= \sum_{j=i}^{K_f[i]-1} \|j, j + 1\|_2 \\ l(i, i - K_b[i]) &= \sum_{j=K_b[i]+1}^i \|j, j - 1\|_2 \end{aligned} \quad (4.3)$$

Euclidean distances and the corresponding trajectory lengths tend to be equal in absence of curvature changes, even if trajectories are noisy. Otherwise, the Euclidean distance is quite shorter than the trajectory length. Thus,  $K_f[i]$  and  $K_b[i]$  are the largest value that satisfies

$$\begin{aligned} l(i, i + K_f[i]) - \|i, i + K_f[i]\|_2 &< U_k \\ l(i, i - K_b[i]) - \|i, i - K_b[i]\|_2 &< U_k \end{aligned} \quad (4.4)$$

being  $U_k$  a constant value that depends on the noise level tolerated by the curvature estimator. Fig. 4.2 shows the process to extract one  $K_f[i]$  value. In order to set the  $U_k$  value, it must be only taken into account that it should not be very large or very small. If the value of  $U_k$  is very large,  $K_f[i]$  and  $K_b[i]$  tend to be large and trajectory details may be missed, and if it is very small,  $K_f[i]$  and  $K_b[i]$  are also small and the resulting function is noisy.  $U_k$  has been experimentally fixed to 1.0 cm. in all tests.

- Calculation of the local vectors  $\vec{f}_i$  and  $\vec{b}_i$  associated to each point  $i$ . These vectors present the variation in the three cartesian axis between points  $i$  and  $i + K_f[i]$ , and between  $i$  and  $i - K_b[i]$ . If  $(x_i, y_i, z_i)$  are the Cartesian coordinates of the point  $i$ , the local vectors associated to  $i$  are defined as

$$\begin{aligned}\vec{f}_i &= (x_{i+K_f[i]} - x_i, y_{i+K_f[i]} - y_i, z_{i+K_f[i]} - z_i) \\ &= (f_{x_i}, f_{y_i}, f_{z_i}) \\ \vec{b}_i &= (x_{i-K_b[i]} - x_i, y_{i-K_b[i]} - y_i, z_{i-K_b[i]} - z_i) \\ &= (b_{x_i}, b_{y_i}, b_{z_i})\end{aligned}\tag{4.5}$$

- Calculation of the angles associated to each trajectory point. Two angles are required to compute a 3D curvature. In the proposed approach, the 3D curvature is projected to XZ and YZ planes. Then, the angles at point  $i$  can be estimated by using the equations (Rosenfeld and Johnston, 1973) (Fig. 4.2):

$$\begin{aligned}\theta_{XZ_i} &= \arccos\left(\frac{(f_{x_i}, f_{z_i}) \cdot (b_{x_i}, b_{z_i})}{|(f_{x_i}, f_{z_i})| \cdot |(b_{x_i}, b_{z_i})|}\right) \\ \theta_{YZ_i} &= \arccos\left(\frac{(f_{y_i}, f_{z_i}) \cdot (b_{y_i}, b_{z_i})}{|(f_{y_i}, f_{z_i})| \cdot |(b_{y_i}, b_{z_i})|}\right)\end{aligned}\tag{4.6}$$

- The curvature associated to each trajectory point is defined by the values

$$\begin{aligned}\kappa_{XZ_i} &= \theta_{XZ_{i+1}} - \theta_{XZ_i} \\ \kappa_{YZ_i} &= \theta_{YZ_{i+1}} - \theta_{YZ_i}\end{aligned}\tag{4.7}$$

Fig. 4.3 shows  $\kappa_{XZ_i}$  values associated to different demonstrations of the same example gesture. In this simplified case, the gesture is composed by only one trajectory. It can be noted that the sequences of  $\kappa_{XZ_i}$  values exhibit significant differences, but they also present the same overall aspect. Thus, they could be used to infer the performed gesture. In order to understand the necessity of extracting dominant points from these trajectories, it has to be considered that performed gestures will be usually characterized by more than one trajectory. Fig. 4.4 shows the amount of data associated to typical captured gestures composed by 13 trajectories and represented using 130 points per curvature function. It is difficult to manage this amount of data *on-line*, thus dominant points are extracted from them.

As stated by Marji and Siy (2004), the capability of a set of points to represent a trajectory depends on the distance from the original trajectory to one generated from these points using some method of interpolation. It is usual to find large intervals in a performed trajectory in which no corners appear. This could lead to bad representations of the trajectory if only corners were marked as dominant points. Thus, a minimum number of dominant points per trajectory must be guaranteed.

Dominant points are extracted from the obtained adaptive curvature function using the following procedure:

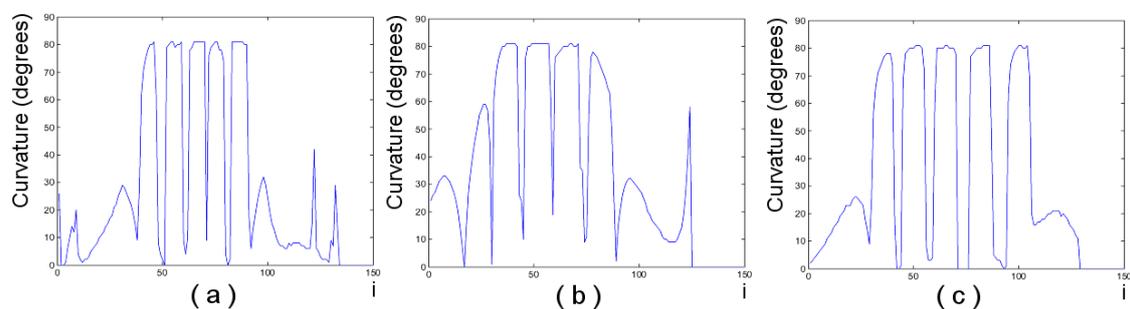


Figure 4.3: a-c)  $\kappa_{XZ_i}$  values associated to three different demonstrations of an example gesture composed by only one trajectory.

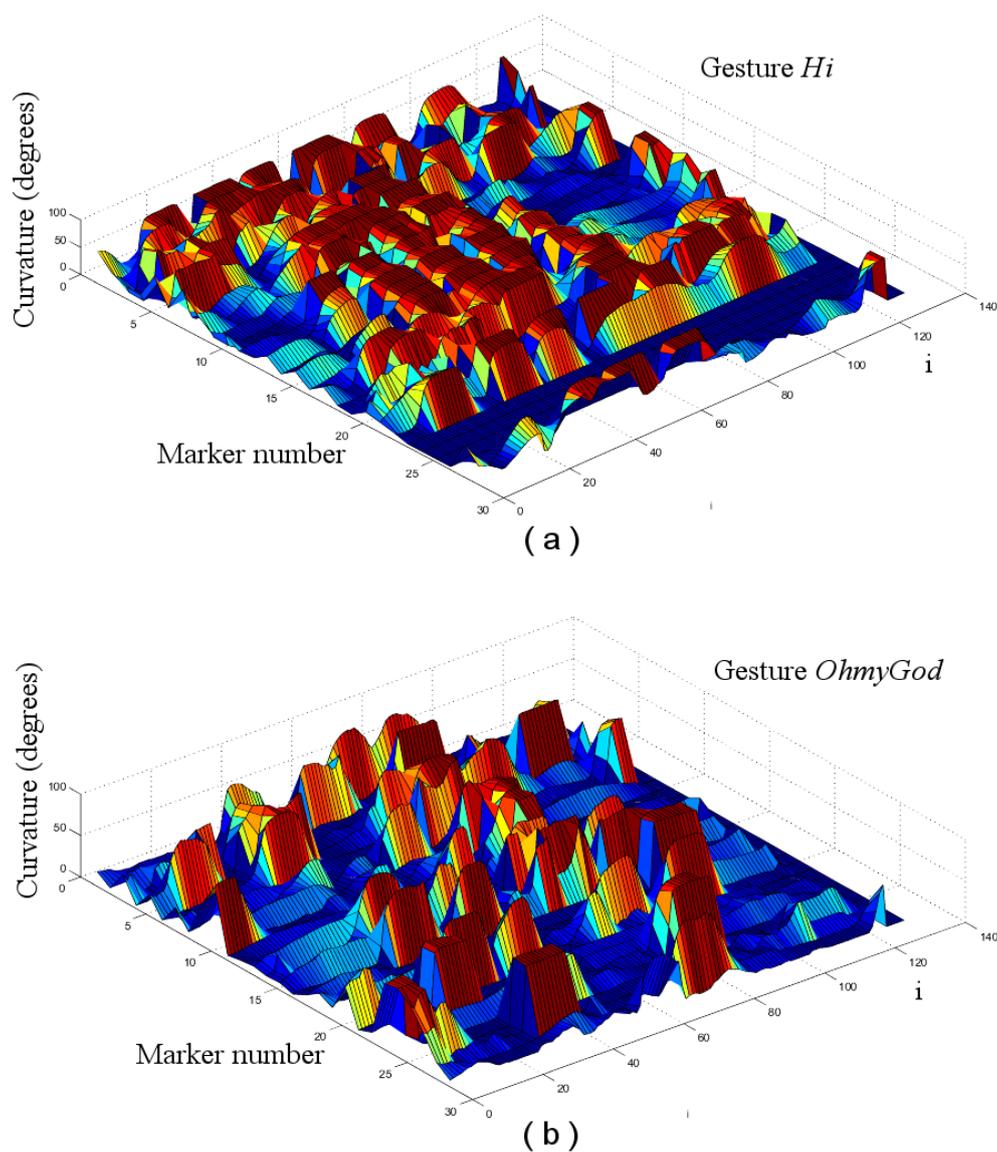


Figure 4.4: a-b) Curvature-based descriptors associated to typical gestures.

- Corners are extracted as local maxima and minima, that satisfy

$$|\kappa_{XZ_i}| > \sigma_c \cup |\kappa_{YZ_i}| > \sigma_c \quad (4.8)$$

where  $\sigma_c$  is a fixed threshold.

- A maximum distance  $\varphi$  is defined as a fraction of trajectory length  $m_\phi$  ( $\varphi = m_\phi/5$  unless a different value is specified). Given two corners  $i$  and  $i + \delta$ , an additional dominant point will be added in the position  $i + \delta/2$  if  $\delta > \varphi$ .
- Finally, two additional points associated to the initial and final part of the trajectory are included.

Each of these dominant points will be characterized by its 3D XYZ position, which will be normalized with respect to a global reference. In this thesis, the trajectories represent the motion of different body parts. Thus, the used global reference is the human waist, a common reference in HMC systems, specially when only the movements of the upper-body are considered.

#### 4.6.2 Validation

Fig. 4.5 confronts results obtained using the adaptive curvature with results obtained using different fixed  $k$  values.  $U_k$  has been experimentally fixed to 1.0 in all tests, and  $\varphi = m_\phi/5$ . The curvature is normalized between [-1...1], and the curvature threshold  $\sigma_c$  is set to 0.8. As Fig. 4.5 shows, lower  $k$  values are very sensitive to noise, while higher  $k$  values tend to filter fast variations thus losing relevant information. This conclusion has been corroborated using more extensive tests. These tests involve the computation of average values of Compression Rate (CR), Integral Square Error (ISE) and the Figure Of Merit (FOM) associated to different sequences. The average time employed to obtain these dominant points is also computed.

The CR is defined by the ratio between the length of the sequence  $m$  and the number of dominant points  $m_{DP}$  used to represent it. The ISE is a measure of the accuracy of the representation. It is computed as depicted in Eq. 4.9, where  $d_i$  are the Euclidean distances between the 3D trajectory points and the polygonal approximation obtained from extracted dominant points.

$$ISE = \sum_{i=0}^m d_i^2 \quad (4.9)$$

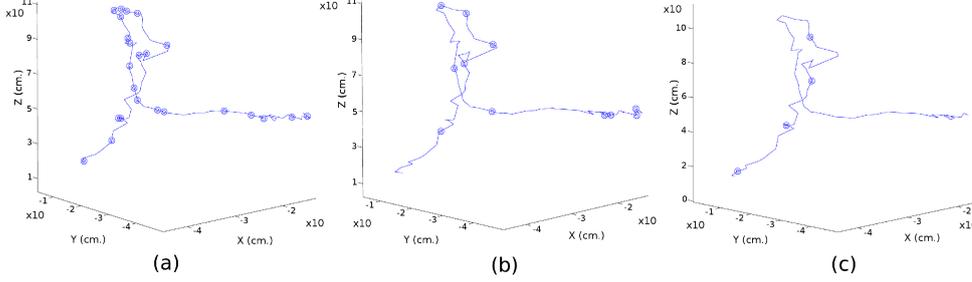


Figure 4.5: Dominant points extraction using different curvature functions. (a)  $k=50$ ; (b) Adaptive curvature; and (c)  $k=500$ .

The FOM is defined by Sarkar (1993) as the ratio between the CR and the ISE associated to the corner-based representation (Eq. 4.10). The FOM measures the quality of the representation, as it increases when either the number of dominant points  $m_{DP}$  or the ISE have low values.

$$FOM = \frac{CR}{ISE} = \frac{m}{m_{DP} \cdot ISE} \quad (4.10)$$

Table 4.1 shows the results obtained when the algorithm is executed on an Intel Core Duo T2400 8 at 1830 MHz. The testing sequences are 20 3D hand trajectories, obtained at a sampling rate of 100 Hz using the motion capture system based on active markers detailed in section 3.7.1.

Table 4.1: Compression rates and execution times for different trajectories.

	$\overline{n^{\circ}frames}$	Hz	$\overline{CR}$	$\overline{ISE}_{\cdot 10^6}$	$\overline{FOM}$	$\overline{msecs}$
Adaptive Curv.	9963	100	315	3.69	$1.95 \cdot 10^{-2}$	421.37
K=50	9963	100	137	0.37	$1.90 \cdot 10^{-2}$	390.97
K=500	9963	100	655	5.84	$1.17 \cdot 10^{-2}$	399.70
CSS	9963	100	327	4.23	$1.76 \cdot 10^{-2}$	1635.23

As depicted, the FOM associated to higher  $k$  values is low, as their ISE values are large. On the other hand, the FOM associated to lower  $k$  values is affected by small CR values. Our experiments involved different fixed  $k$  values but none of them improves the results obtained by an adaptive  $k$  value. The proposed method has also been compared with a standard shape descriptor, the CSS algorithm (Mokhtarian and Mackworth, 1986). CSS has been executed over the XZ and YZ projections of the previous 3D sequences, as it is a 2D shape descriptor. Results

are very similar in terms of compression rates, but the proposed method is faster, thus it is more suitable for on-line applications. Its related ISE values are also slightly lower, thus it may be a better choice for motion reconstruction purposes.

## 4.7 Trajectory matching

Many distance functions have been proposed to evaluate similarity of time series. While most of them focus on the evaluation of a single 2D (or 1D) trajectory, it is possible to extend them to compare gestures composed by multiple 3D trajectories. This section first introduces different distance functions that have been evaluated to be used in the proposed gesture recognition system. The validity of these functions when applied to the proposed gesture representation is also discussed. Finally, this section details the usage of new global features to improve the matching results.

### 4.7.1 Local distance computation

Let  $A, B$  be two gestures composed by  $n$  trajectories. Let  $\Upsilon^A = (\vec{v}_1^A, \vec{v}_2^A \dots \vec{v}_n^A)$  and  $\Upsilon^B = (\vec{v}_1^B, \vec{v}_2^B \dots \vec{v}_n^B)$  be the sets of dominant points for these gestures, and let vector  $\vec{v}_i^X = (v_i^X(1), v_i^X(2) \dots v_i^X(m_{X_i}))$  be the dominant points for trajectory  $i$  of gesture  $X$ . The distance  $d^{AB}$  between gestures  $A$  and  $B$  can be defined as follows:

$$d^{AB} = \alpha_1 \cdot d_1^{AB} + \alpha_2 \cdot d_2^{AB} + \dots + \alpha_n \cdot d_n^{AB} \quad (4.11)$$

where  $\alpha_i \in \mathfrak{R}+$ , and it is set to 1.0 in absence of prior knowledge about the gesture.  $d_i^{AB}$  is the distance between  $\vec{v}_i^A$  and  $\vec{v}_i^B$ . The different methods evaluated to obtain these distances are presented in Table 4.2, in which equations consider two trajectories (in our case, two set of dominant points)  $\Psi$  and  $\Phi$ , with lengths  $m_\psi$  and  $m_\phi$ . The first of these methods is the evaluation of the Euclidean distance, defined as Formula 1 in Table 4.2. As Euclidean distance requires the compared trajectories to have similar lengths, the shorter trajectory is enlarged to match the longest one. DTW is defined as Formula 2 in Table 4.2 and does not require two trajectories to have the same length, thus  $\vec{v}_i^A$  and  $\vec{v}_i^B$  sets can be directly compared. In opposition to Euclidean Distance, and Lp-norm methods in general, DTW can also handle the local time shifting by duplicating previous elements in the sequence. Edit Distance on Real Sequence (EDR), ERP and LCSS share these characteristics of the DTW, although the time shifting is handled in different ways. EDR, defined as Formula 3 in Table 4.2, is based on the

well-known Edit Distance (ED) algorithm, used to measure the distance between two strings. As dominant points are not characters but 3D numerical values, it is important to define properly *matching* between two dominant points.

**Definition 1.** Two dominant points  $\vec{\psi}_i$  and  $\vec{\phi}_i$  are said to match if and only if

$$\|\vec{\psi}_i - \vec{\phi}_i\| \leq \epsilon_\kappa \quad (4.12)$$

where  $\epsilon_\kappa$  is the matching threshold.

As depicted in Table 4.2, EDR will provide a distance based on binary costs, rather than real distances. Thus, the choice of the matching threshold becomes a key decision for this method. ERP follows the same approach as EDR as it is also based on ED, but it uses real distance measurements instead of binary costs. ERP is defined by Formula 4 in Table 4.2, and it introduces a constant value  $g$  as the gap of edit distance. Chen et al. (2005) demonstrate that as long as this  $g$  value is fixed, it satisfies the triangle inequality that characterizes ERP method as a metric. They propose the use of  $g = 0$  (in this thesis,  $\vec{g} = (0, 0, 0)$ ).

The last evaluated distance function is the LCSS, defined as Formula 5 in Table 4.2. LCSS has been applied to time series matching by different authors (Croitoru et al., 2005; Vlachos et al., 2004). It returns the longest subsequence common to the pair of compared sequences. As EDR, it requires the use of a threshold to determine whether or not two elements match. LCSS, on the other hand, does not consider different gap sizes between similar subsequences.

The ability of these methods to match sets of XYZ trajectories is evaluated in the following section. However, a prior description about the validity of these distances can be deduced if some characteristics of the considered recognition framework are taken into account:

- a A gesture  $A$  is represented by a set of dominant points  $\Upsilon^A$ . Each dominant point is stored as a 3D XYZ position.
- b Similar gestures may be performed by different subjects at different locations and/or orientations depending on the global position of the person (Croitoru et al., 2005).
- c While a global pose normalization is possible by translating, rotating and scaling the whole perceived motion, relative positions, orientations and amplitudes are relevant to recognize a gesture (i.e. reaching one's stomach usually means 'hungry', while touching one's chest usually means 'me').

Table 4.2: Distance functions (Chen et al. 2005).

$$\begin{aligned}
(1) \quad Eu(\Psi, \Phi) &= \sqrt{\sum_{i=1}^{m_\psi} \|\vec{\psi}_i - \vec{\phi}_i\|^2} \\
(2) \quad DTW(\Psi, \Phi) &= \begin{cases} 0 & \text{if } m_\psi = m_\phi = 0 \\ \infty & \text{if } m_\psi = 0 \text{ xor } m_\phi = 0 \\ \|\vec{\psi}_1 - \vec{\phi}_1\|^2 + \min\{DTW(Rest(\Psi), Rest(\Phi)), \\ DTW(Rest(\Psi), \Phi), DTW(\Psi, Rest(\Phi))\} & \text{otherwise} \end{cases} \\
(3) \quad EDR(\Psi, \Phi) &= \begin{cases} m_\psi & \text{if } m_\phi = 0 \\ m_\phi & \text{if } m_\psi = 0 \\ \min\{EDR(Rest(\Psi), Rest(\Phi)) + subcost, \\ EDR(Rest(\Psi), \Phi) + 1, EDR(\Psi, Rest(\Phi)) + 1\} & \text{otherwise} \end{cases} \\
&\text{where } subcost = 0 \text{ if } match(\vec{\psi}_1, \vec{\phi}_1) = \text{true} \text{ and } subcost = 1 \text{ otherwise} \\
(4) \quad ERP(\Psi, \Phi) &= \begin{cases} \sum_1^{m_\psi} \|\vec{\psi}_i - \vec{g}\|^2 & \text{if } m_\phi = 0 \\ \sum_1^{m_\phi} \|\vec{\phi}_i - \vec{g}\|^2 & \text{if } m_\psi = 0 \\ \min\{ERP(Rest(\Psi), Rest(\Phi)) + \|\vec{\psi}_1 - \vec{\phi}_1\|^2, \\ ERP(Rest(\Psi), \Phi) + \|\vec{\psi}_1 - \vec{g}\|^2, \\ ERP(\Psi, Rest(\Phi)) + \|\vec{\phi}_1 - \vec{g}\|^2\} & \text{otherwise} \end{cases} \\
(5) \quad LCSS(\Psi, \Phi) &= \begin{cases} 0 & \text{if } m_\psi = 0 \text{ or } m_\phi = 0 \\ LCSS(Rest(\Psi), Rest(\Phi)) + 1 & \text{if } match(\vec{\psi}_1, \vec{\phi}_1) = \text{true} \\ \max\{LCSS(Rest(\Psi), \Phi), LCSS(\Psi, Rest(\Phi))\} & \text{otherwise} \end{cases} \\
&\text{note : } Rest(\Pi) = \Pi - \vec{\pi}_1
\end{aligned}$$

The use of an adaptive curvature function to extract dominant points reduces noise. This improves the performance of all distance functions, but specially of Euclidean distance, DTW and ERP that are more sensitive to it (Chen et al., 2005). Local time shifting is still present, thus Euclidean distance is expected to offer worse results respect to other approaches. On the other hand, outliers can still introduce errors when using these methods based on real distances. EDR and LCSS are less sensitive to these outliers, but their dependency on a fixed threshold makes them very sensitive to relative transformations, that will be common between different performances. Chen et al. (2005) proposes the normalization of the processed trajectory using its corresponding mean and standard deviation, but characteristic (c) prevents us from using this approach.

#### 4.7.2 Distance functions evaluation

The different distance functions has been evaluated using a large database containing 11 different upper-body gestures (Fig. 4.6) that were performed multiple times by six different people, given a total amount of 190 performed gestures. These gestures were captured using the Codamotion CX1 motion capture system detailed in section 3.7.1. The duration of recorded gestures vary from less than 1 second up to 7.5 seconds, and the average amount of samples per gesture is 361.5 (as 13 markers were tracked to capture the motion, each gesture will be characterized by

an average of  $13 \cdot 361.5 \simeq 4700$  XYZ values). Global pose normalization was achieved by using the positions of the marker in the waist to infer the person position, orientation and height. As stated in section 4.7.1, local normalization was not applied to individual trajectories<sup>1</sup>.

Thresholds values for EDR and LCSS were set to a quarter of the maximum standard deviation of the evaluated trajectory, as recommended in (Chen et al., 2005). The  $\vec{g}$  value for ERP was set to  $(0, 0, 0)$ . Three executions of 8 gestures were stored as learned, labelled data (given a total amount of 24 gestures in the database, distributed in 8 classes). The remainder three gestures were left out of the database to further test the learning capabilities of the system.

The performance of the methods is measured by using  $k$ -nearest neighbor ( $k$ -NN) algorithm for two different  $k$  values (1,3). We also evaluate the results using two more restrictive measures: 3/3-NN considers a gesture as correctly recognized if and only if the three closest neighbors belong to the same class. 3/4-NN is equivalent to 4-NN, but using only three items per class and considering tied votes a miss. The experiments also measure the execution time of the distance evaluation, and for each comparison the difference between the first and fourth nearest neighbors,  $\Delta conf$ , is stored as a measurement of the discrimination ability of the method. Table 4.3 shows the obtained results.

Table 4.3: Evaluation of different distance functions to measure local similarity between gestures represented as sets of trajectories.

	<b>DTW</b>	<b>EDR</b>	<b>ERP</b>	<b>LCSS</b>	<b>Euclid.</b>
$\Delta conf$	0.3	0.16	0.21	0.21	0.18
secs.	0.55	0.52	0.74	0.25	0.02
1-NN	95%	60%	97%	59%	87%
3-NN	93%	23%	93%	23%	75%
3/3-NN	77%	5%	57%	10%	31%
3/4-NN	83%	8%	69%	11%	48%

As predicted, EDR and LCSS are heavily penalized because of the utilization of a threshold to discriminate whether two dominant points match or not. This leads to failures when different performers execute the same movement in a slightly different pose, as Fig. 4.7 depicts. These errors are more usual and their accumulative effects are much more significant as the number of evaluated trajectories increases. Normalization of compared trajectories does not significantly improve the success rate, while it produces a higher amount of false positives as local distances between different gestures are reduced. A possible solution for these methods to be used in the considered gesture recognition scenarios would be to evaluate independently,

<sup>1</sup>captured motion data are available at [www.grupoisis.uma.es](http://www.grupoisis.uma.es)

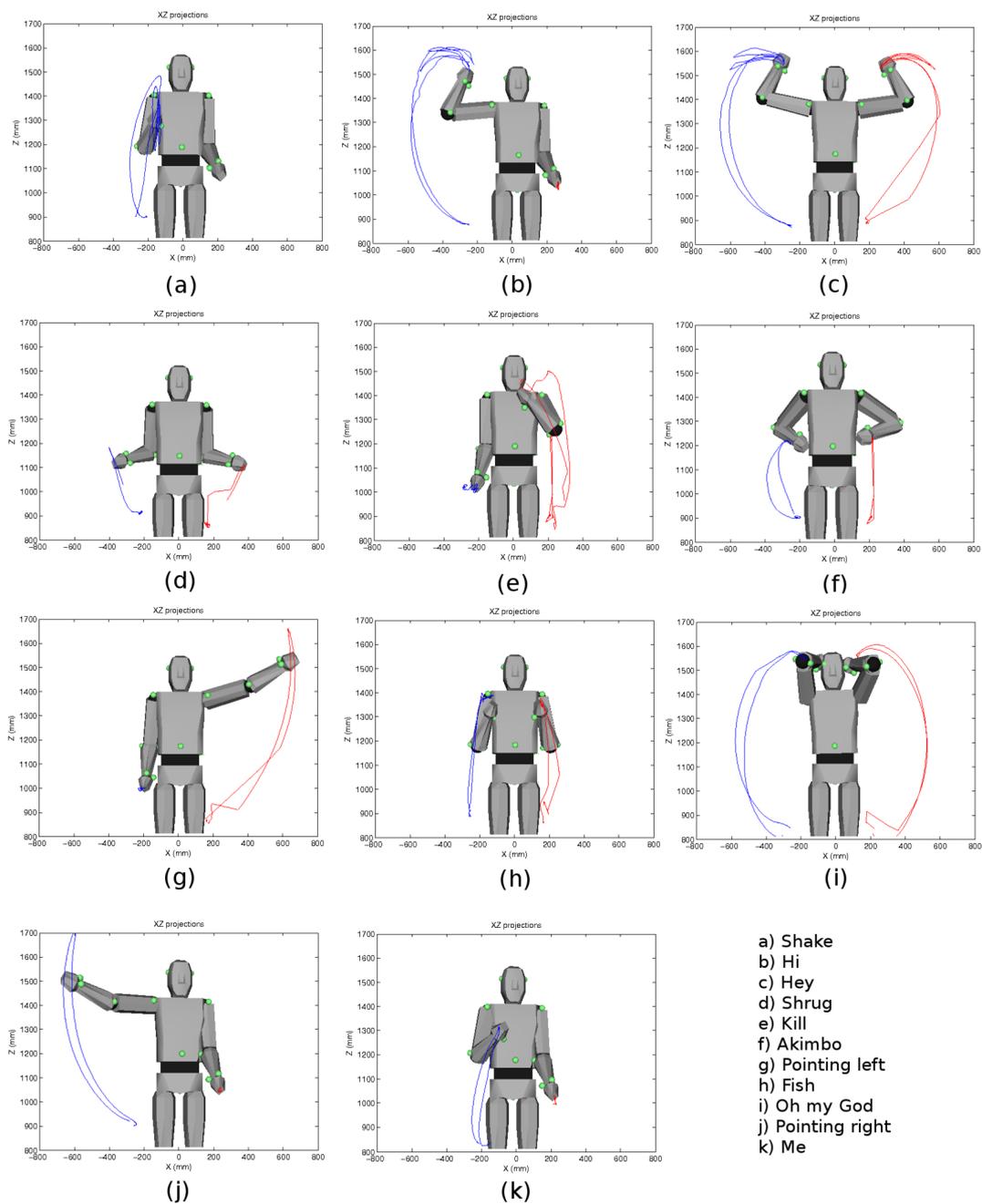


Figure 4.6: Upper-body social gestures used to test the gesture recognition and learning system. The trajectories of the left and right hands have been marked over the frontal view of a 3D model of the human performer.

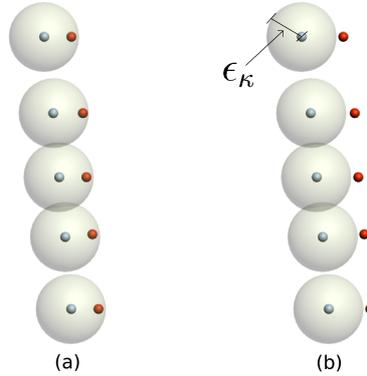


Figure 4.7: Effects concerning the use of a matching threshold: (a) Blue and red trajectories are considered to match perfectly; and (b) Due to a small global variation in the red trajectory, blue and red trajectories are now considered completely different.

for each trajectory, both original and normalized distances and to obtain the final distance as a combination of these trajectory sub-distances. But even in this situation, EDR and LCSS would confront the problem that dominant points are sparse measures, very difficult to differentiate from the outliers these methods are designed to avoid (Chen et al., 2005; Croitoru et al., 2005).

Euclidean distance, DTW and ERP behaves as expected. Euclidean distance is the fastest method, but DTW offers better recognition rates and its accuracy does not decrease when considering  $k > 1$  in the  $k$ -NN algorithm. Its  $\Delta conf$  value is also higher, as it is able to deal with local time shifting thus improving the discrimination capability of the recognizer. The results using the more restrictive 3/3-NN and 3/4-NN approaches confirm the robustness of discrimination based on DTW, and ERP, over Euclidean distance and costs-based methods.

While these data suggests that both DTW and ERP are adequate to measure local distances between gestures, DTW seems slightly superior in terms of robustness. Thus, this is the algorithm selected to be integrated in the proposed vision-based gesture recognition system.

### 4.7.3 Global features

A set of simple geometric features are used to further increase the discrimination ability of the previously evaluated distance functions. These global features are less sensitive to noise and outliers, but they face major difficulties in capturing fine details and therefore dissimilar trajectories may have very similar global features (Alajlan et al., 2007).

Two global features are used in this thesis, both related to trajectory amplitudes. The first global feature is computed as the difference between the minimum and maximum X, Y,

Z values that were reached by each trajectory while performing the gesture, while the second global feature stores the relative movement between different trajectories of a certain gesture. If a gesture  $A$  is composed by  $n$  trajectories, its first global feature is a set of  $n$  vectors,  $\vec{a}_k^{abs}$  ( $k \in [1\dots n]$ ), while the second feature is a set of  $n - 1$  vectors,  $\vec{a}_k^{rel}$  ( $k \in [1\dots(n - 1)]$ ). These features are computed as follows:

$$\begin{aligned} \vec{a}_k^{abs} &= (\Delta X_k^A, \Delta Y_k^A, \Delta Z_k^A) & \forall k \in [1\dots n], k \in \mathbb{N}^+ \\ \vec{a}_k^{rel} &= \left( \frac{\Delta X_k^A}{\sum_{j=1}^n (\Delta X_j^A)}, \frac{\Delta Y_k^A}{\sum_{j=1}^n (\Delta Y_j^A)}, \frac{\Delta Z_k^A}{\sum_{j=1}^n (\Delta Z_j^A)} \right) & \forall k \in [1\dots(n - 1)], k \in \mathbb{N}^+ \end{aligned} \quad (4.13)$$

where  $\Delta P_k^A \in \mathbb{N}^+$  is the difference between the maximum and minimum values the coordinate  $P$  of trajectory  $k$  reached for gesture  $A$ .

While these features include considerable information about the global properties of the motion, it may be complex to use them to represent a trajectory, since some dissimilar trajectories have comparable global features. Therefore, instead of using these global features to index the perceived gestures into a database, they are used to reinforce the local confidence value, as detailed below.

#### 4.7.4 Confidence Reinforcement

Demiris and Hayes (2002) introduced the concept of confidence value, as an indicator of how confident an imitator's behaviour is that it can match an observed behaviour. In this thesis, the confidence value  $C_i(A)$  for an observed gesture  $A$  displays its similarity with the  $i$  gesture stored in the database,  $G_i$ , and is computed using Eq. 4.14. As commented above, in this Equation the roles of global and local features are different, and thus  $C_i(A)$  can be described as the local similarity reinforced by global factors.

$$C_i(A) = \left( \frac{\rho^{AG_i}}{M_g} \right) \cdot \left( 1 - N \cdot \frac{d^{AG_i}}{\sum_{k=1}^N d^{AG_k}} \right) \quad (4.14)$$

$N$  is the number of gestures in the database,  $d^{AG_i}$  the local distance between the input gesture  $A$  and the  $i$ th stored gesture ( $G_i$ ), and  $\rho^{AG_i} \in [0\dots 1]$  gives a measure of the global similarity between compared gestures.  $M_g$  is the maximum  $\rho^{AG_i}$ . As depicted, a low global similarity

strongly penalizes the confidence value. The following equations are used to compute  $\rho^{AG_i}$ :

$$\begin{aligned} \rho^{AG_i} &= \frac{1}{2} \cdot (\rho_1^{AG_i} + \rho_2^{AG_i}) \\ \rho_1^{AG_i} &= \frac{1}{3n} \cdot \left( \sum_{k=1}^n \left( \frac{\min(\Delta X_k^A, \Delta X_k^{G_i})}{\max(\Delta X_k^A, \Delta X_k^{G_i})} + \frac{\min(\Delta Y_k^A, \Delta Y_k^{G_i})}{\max(\Delta Y_k^A, \Delta Y_k^{G_i})} + \frac{\min(\Delta Z_k^A, \Delta Z_k^{G_i})}{\max(\Delta Z_k^A, \Delta Z_k^{G_i})} \right) \right) \\ \rho_2^{AG_i} &= \frac{1}{n-1} \cdot \left( \sum_{k=1}^{n-1} \left( 1 - \frac{1}{3} \cdot \|\vec{a}_k^{rel} - \vec{g}_{ik}^{rel}\|_2 \right) \right) \end{aligned} \quad (4.15)$$

where  $n$  is the number of trajectories associated to the gesture. Fig. 4.8 depicts an example of global comparison between an input gesture and a certain amount of memorized gestures. For this example, gestures composed by only the trajectories of the right and left hands are used. All gestures except the one labeled 'shake' are performed in the frontal plane, XY, and thus, with this exception, only a little amount of movement is associated to Z coordinate. Gestures have been captured using the vision-based motion capture system detailed in chapter 3. As depicted, the second stored gesture, labeled 'pointing left', obtains the lower global similarity. This is the expected result, as the gesture is performed using a different arm and only Z components of global features are similar. On the other hand, gesture 'shake' is the only one that involves using the same hand than the input gesture. However, extracted global features are able to differentiate movement in different planes. Thus, as gesture 'shake' includes a movement in the Z coordinate that is not present in the input gesture, the global similarity is penalized. Finally, gestures 'hey' and 'shrug' are very similar in terms of relative movements. As depicted,  $\rho_2^{AG_i}$  is quite similar in both cases. Gesture 'hey' obtains a higher similarity because its amplitudes are more similar to those of the input gesture, as  $\rho_1^{AG_i}$  shows.

As commented above, selected global features are not able to offer a robust comparison by themselves. They should instead be used to reinforce local matching. Next section will evaluate the performance of the gesture recognition with and without using global features reinforcement, in order to validate this proposal.

#### 4.7.5 Reinforced confidence value evaluation

The complete trajectory matching algorithm has been evaluated by considering the same experimental scenario that was described in section 4.7.2, but including global reinforcement to obtain the confidence values. Table 4.4 confronts these new results (marked as **R**) with the previously obtained ones. As depicted, the proposed reinforcement factor improves both discrimination (increasing  $\Delta conf$ ) and successfully recognized gestures. Besides, time consumed to compute global features is negligible, thus it is a reasonable addition to the system.

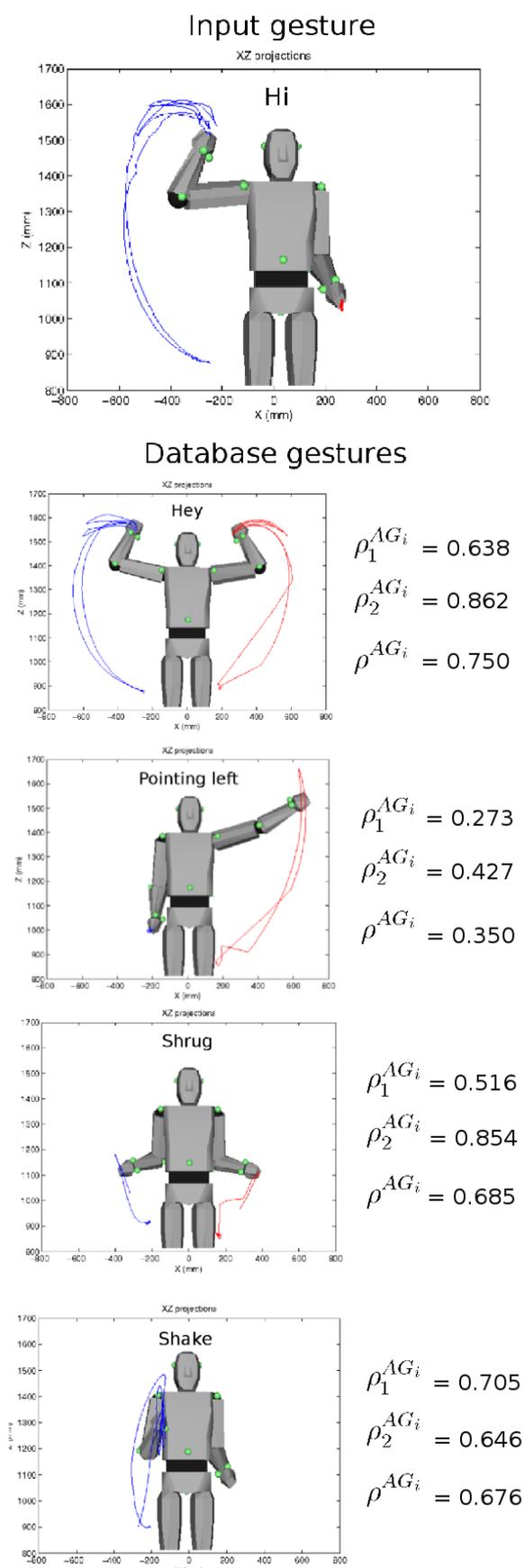


Figure 4.8: Global similarities between different gestures.

Table 4.4: Evaluation of different distance functions to measure local similarity between gestures represented as sets of trajectories, including global reinforcement.

	DTW		EDR		ERP		LCSS		Euc. dist.	
		R		R		R		R		R
$\Delta\text{conf}$	0.3	0.32	0.16	0.17	0.21	0.4	0.21	0.27	0.18	0.19
secs.	0.55	0.55	0.52	0.52	0.74	0.74	0.25	0.25	0.02	0.02
1-NN	95%	95%	60%	64%	97%	97%	59%	77%	87%	88%
3-NN	93%	96%	23%	30%	93%	96%	23%	57%	75%	79%
3/3-NN	77%	82%	5%	5%	57%	66%	10%	19%	31%	38%
3/4-NN	83%	86%	8%	8%	69%	75%	11%	26%	48%	56%

## 4.8 Knowledge update

Once the human demonstrator has performed a certain gesture, the confidence values  $C_i(A)$  associated to each stored gesture  $i$  are evaluated in order to determine whether the gesture should be incorporated to the repertoire or not. As commented above, this concept of *learning* does not imply generalization nor adaptation to different circumstances. In this thesis, *learning* refers to the process of acquiring new behaviours, as in [Demiris and Hayes \(2002\)](#).

Chapter 3 described the stereo cameras used to capture motion data in this thesis. As detailed, the gesture representations obtained when using this perceptual system may be sparse and noisy. It is necessary, then, to use a robust decision criterion to prevent false matchings without affecting the gesture recognition rates. This section details the knowledge update algorithm used in this thesis. This algorithm is based on the use of an absolute threshold and a relative threshold. The absolute threshold  $\Omega$  selects all gestures in the memorized repertoire that are close to the performed one. A stored gesture  $G_i$  is considered to be close to the performed gesture  $A$  if  $C_i(A) > \Omega$ . On the other hand, the relative threshold  $\omega$  compares the two highest confidence values,  $C_{i_1}(A)$  and  $C_{i_2}(A)$ , obtained for a performed gesture. The relative threshold is used to mark a gesture if it is not only similar enough to the stored gesture with the highest  $C_i$ , but also if it is different enough to the rest of stored gestures. For the experiments described in this thesis, the absolute threshold  $\Omega$  was set to 0.55, and the relative threshold  $\omega$  was set to 0.25. Other values in these ranges may also offer adequate results.

The dataflow of the final knowledge update algorithm is depicted in Figure 4.9, where  $C_{i_1}$  and  $C_{i_2}$  are the first and second maximum confidence values, respectively.  $\Omega$  and  $\omega$  are the absolute and relative decision thresholds.

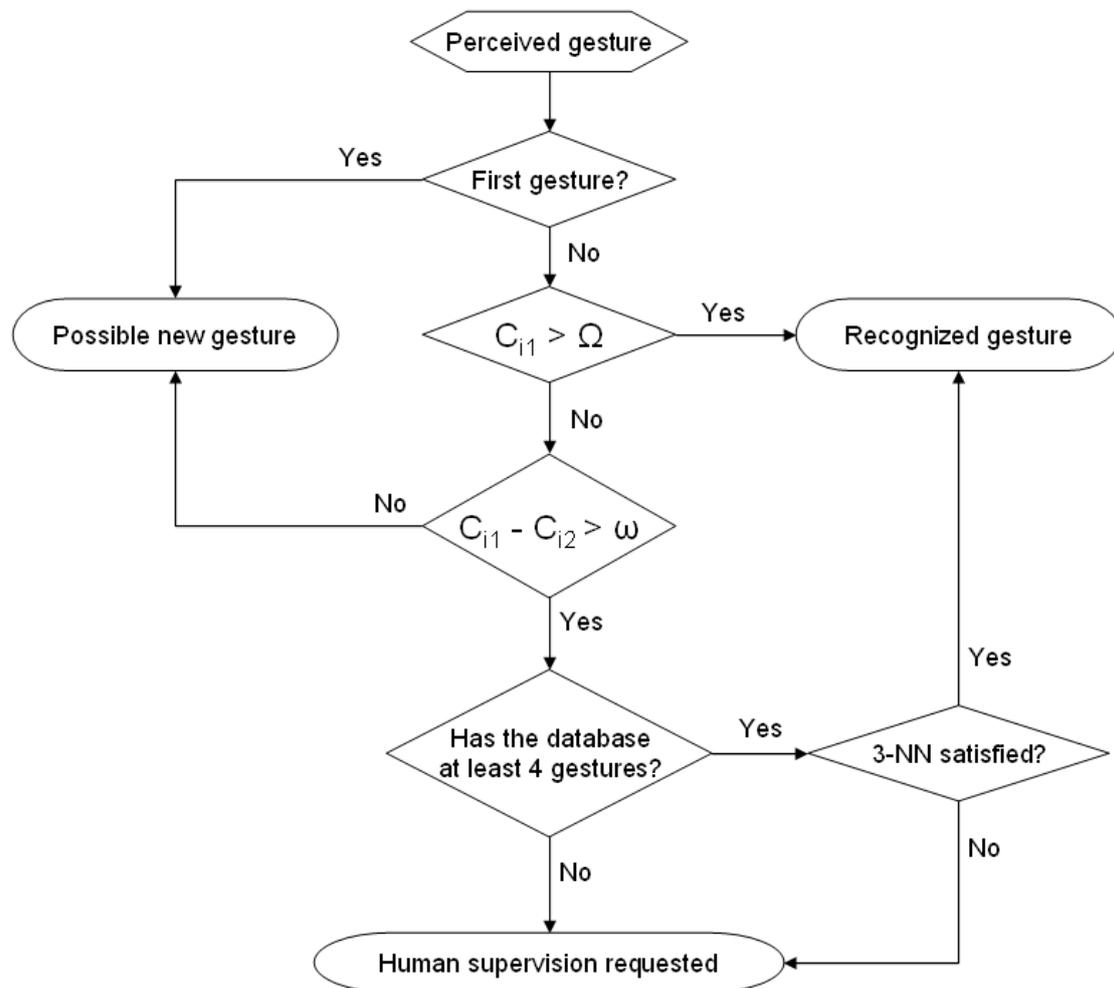


Figure 4.9: Dataflow of the knowledge update algorithm employed to test the RLbI system in real scenarios.

It can be seen in this dataflow that human supervision appears in the process. While some degree of autonomous learning is a desirable feature for any social robot, it is also true that social learning requires interaction (Mosterín, 2005). Humans do correct and supervise their interaction partners when they teach them new gestures or behaviours. They also ask for assistance or additional explanations or demonstrations if they are not sure whether they are correctly imitating a demonstrated gesture or not. Thus, including human supervision in the learning process is not a hard constraint. It is also a solution adopted by many researchers in this area (Breazeal et al., 2004; Kojo et al., 2006). Calinon (2007) even mentions that this human supervision phase is not only a light constraint, but a necessary addition for a RLbI algorithm.

It is important to consider that new gestures are not directly incorporated to the gesture database, but they are selected as *probable* new gestures. The social robot will always ask the human demonstrator for confirmation before adding gestures to the database. Thus, although sometimes the system perceives incorrect or partial gestures (Fig. 6.12), or merge different gestures in the same one, this validation allows to avoid inclusion of incorrect gestures in the database.

The first perceived gesture is directly selected as possible new gesture, thus human supervision is requested to include it in the database (i.e. the human is told to name the gesture). On the other hand, gestures that are very similar to an already stored one, i.e. its  $C_{i_1} > \Omega$ , are automatically recognized. The process becomes more complex for more ambiguous gestures. As Fig. 4.9 shows, if  $C_{i_1} \leq \Omega$  then it is necessary for the perceived gesture to satisfy both the relative distance condition and a 3-NN condition in order to be automatically recognized. Otherwise the robot will ask for human supervision to classify the gesture, discard it or add it to the database as a new gesture. The use of a  $k$ -NN criterion to reinforce the decision is motivated by the noisy and sparse nature of perceived data. Selecting an adequate  $k$  requires a trade-off solution: higher  $k$  values are more robust against false recognitions, but they are also more restrictive in matching, thus they may produce false detections of new gestures. Higher  $k$  values also require to raise the minimum number of executions stored for each gesture. Besides, they take more time in performing recognition as more comparisons need to be executed. Although the probability of error decreases as the number of gestures in the database increases (Duda et al., 2001), it is necessary to ease a fast response for the social robot. Thus, only three executions of each gesture are stored in the database, and the 3-NN criterion is used, a criterion that has offered adequate results for the experiments performed in this thesis. The use of advanced methods to reduce computational complexity of the  $k$ -NN algorithm (Duda et al.,

2001) may allow to increase the number of stored gestures if required.

To summarize, there are two circumstances in which a perceived gesture can be recognized:

- $C_{i_1} > \Omega$
- $(C_{i_1} - C_{i_2}) > \omega \wedge 3 - NN$  satisfied.

These conditions allow to recognize a gesture even when partial information, noise and errors avoid that gesture obtaining high confidence values in any comparison, but still the perceived motion is remarkably closer to the executions of a certain stored gesture.

Finally, it is important to notice that in the first stages of the gesture database construction, i.e. when less than four different gestures have been included, the amount of required human supervision grows, and the 3-NN condition is not checked as there are too few different gestures in the database. Thus, in this initial stage, if the absolute threshold condition is not meet but the relative threshold condition is meet, then the human supervisor is directly asked about the gesture. As before, the human supervisor can at this point classify the gesture, discard it or mark it as a new gesture.

## 4.9 Comparison between PCA+LDA and DPD+DTW

As commented in section 4.5, a feature based representation has been used instead of a generic dimensionality reduction technique. In order to validate this decision, the proposed gesture learning system has been compared with a traditional classification system based on the combined use of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). This system uses PCA to encode perceived motion. While PCA-based classification has also been employed for comparison purposes, the proposed approach follows the suggestion of Pang et al. (2005), and uses LDA to classify the compressed representations of the trajectory. This system has been implemented for evaluation purposes, thus complex incremental learning (Pang et al., 2005; Ozawa et al., 2008) has not been considered. Batch learning has been used instead. The resulting recognition rates will be confronted with the ones obtained using the proposed DPD+DTW based system. As both PCA+LDA and DPD+DTW approaches can equally benefit from global reinforcement, this step is not applied. Thus, it is the *local* discrimination ability of these two techniques what is compared in this section.

As commented above, we firstly apply Principal Component Analysis (PCA) to reduce the amount of data to be processed in the recognition step. PCA is an approach for dimensionality reduction which determines the directions along which the variability of the data is maximal (Jolliffe, 1986). PCA is conducted by extracting the eigenvectors of the total scatter matrix of the database  $S_T$ , defined as

$$S_T = \sum_{i=1}^N (G_i - \bar{G})(G_i - \bar{G})^T \quad (4.16)$$

where  $\bar{G}$  is the mean value of the database of  $N$  gesture descriptors  $G_i$ .

Eigenvectors  $W_i$  (*eigengestures*) and associated eigenvalues  $\lambda_i$  are calculated by solving

$$S_T W_i = \lambda_i W_i \quad \forall i \in \{1, \dots, d\} \quad (4.17)$$

The transformation matrix is then defined as  $\mathcal{W} = \{W_1, W_2, \dots, W_K\}$ , where  $K$  is the minimal number of eigengestures used to obtain a satisfying representation of the data. Thus,  $\mathcal{W}$  is an orthogonal transformation that diagonalizes the covariance matrix  $S_T$ .

For the comparison with the proposed DPD-DTW method, 90 % of the gestures will be used to extract the set of eigengestures in the training phase. The remainder gestures will be used to test the system. Then, the compressed feature vector associated to a gesture descriptor will be obtained by projecting it onto this set. This projection onto a latent space not only reduces dimensionality but also decorrelates the data. When considering a small database of high dimensionality, this decorrelation can be useful for further encodings, due to the sparsity of data in high-dimensional space (Calinon, 2007).

Figs. 4.10.a and 4.10.b shows the feature vectors respectively associated to the example gesture descriptors depicted in Figs. 4.4.a and 4.4.c. In this case, a set of  $k = 35$  eigengestures has been chosen to ensure that the projection of the data onto this reduced set covers at least 90% of the data's spread,  $\sum_{i=1}^K \lambda_i / \sum_i \lambda_i > 0.9$ .

Let us now assume that  $N$  training gestures  $\{G_i\}_{i=1\dots N}$  in  $M$  groups have been presented so far. These gestures have been projected onto a low-dimensional subspace using PCA, obtaining a set of feature vectors  $g_i = \mathcal{W}^T G_i$ . Let the between-class scatter matrix  $S_b$  and the within-class scatter matrix  $S_w$  of this database be defined as

$$\begin{aligned} S_b &= \sum_{c=1}^M n_c (\bar{g}_c - \bar{g})(\bar{g}_c - \bar{g})^T \\ S_w &= \sum_{c=1}^M \sum_{g \in \{g_c\}} (g - \bar{g}_c)(g - \bar{g}_c)^T \end{aligned} \quad (4.18)$$

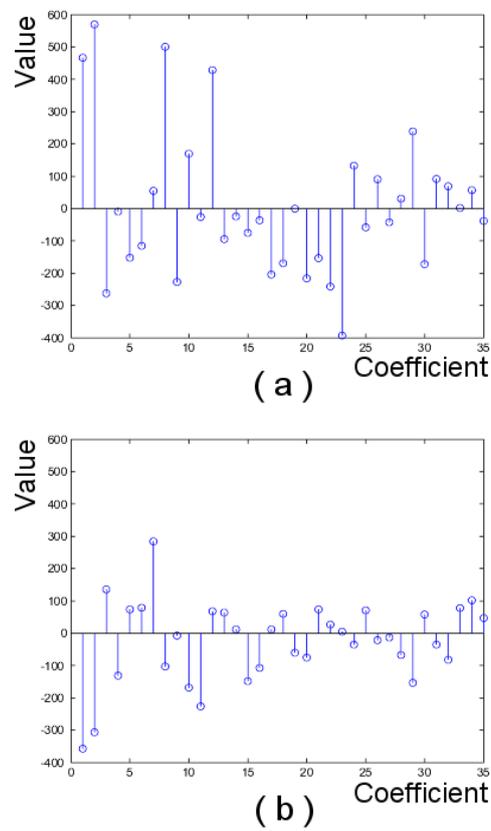


Figure 4.10: a-b) PCA projections of the gesture descriptors in Figs. 4.4a-b

where  $n_c$  and  $\bar{g}_c$  are the number of samples and the mean vector of the group  $c$ , respectively, and  $\bar{g}$  is the mean vector of the whole set of feature vectors  $\{g_i\}$ .

Linear Discriminant Analysis (LDA) seeks a linear transformation  $U$  over the set of gestures in such a way that the ratio of the between-class scatter matrix,  $S_b$ , and the within-class scatter matrix,  $S_w$ , is maximized. Then, the transformation matrix  $\mathcal{U}$  can be obtained by conducting an eigenvalue decomposition of the matrix  $S_w^{-1}S_b$ .

$$S_w^{-1}S_b\mathcal{U} = \mathcal{U}\Lambda \quad (4.19)$$

The columns of  $\mathcal{U}$  correspond to the discriminant eigenvectors. Similar to the PCA algorithm, eigenvectors with small eigenvalue can be discarded.

Once the transformation matrix  $\mathcal{U}$  has been obtained, Euclidean distance can be used to measure the similarity between two gesture descriptors  $T$  and  $G$ . This distance  $d^{TG}$  is computed as

$$d^{TG} = \|\mathcal{U}^T t - \mathcal{U}^T g\|_2 \quad (4.20)$$

where  $t$  and  $g$  are the PCA-transformed versions of gestures  $T$  and  $G$ , respectively.

Experiments measure the performance of the two compared approaches using  $k$ -nearest neighbor ( $k$ -NN) algorithm for two different  $k$  values (1,3). For comparison purposes, the same tests were conducted in two more scenarios: (i) using Euclidean distance to directly measure the similarity between feature vectors provided by PCA; and (ii) using DTW over complete trajectories instead of over sets of dominant points.

For the PCA-LDA recognition approach, the eigengestures were obtained using 90 % of gestures in the training phase, while the remaining 10 % were used to test the system. The PCA algorithm was used to obtain a eigenbase of 35 components. While the quality of reconstruction from sequences of dominant points was previously discussed in section 4.6, the accuracy of the resulting PCA projections should be evaluated before using them to recognize perceived gesture. This evaluation was achieved by comparing original gestures  $G$  and the reconstruction of their projections  $G'$ . The comparison, performed by computing the minimum square distance between  $G$  and  $G'$ , shows that only 47,61 % of the original gesture is recovered from its projection. This low reconstruction rate be explained if we consider that PCA algorithm is not encoding temporal shifting, thus different performances of the same gesture may not be projected into the eigenbase as expected.

Table 4.5 shows the results provided by the different approaches. It can be noted that the combination of PCA and LDA offers better classification results than PCA, as stated in Pang et al. (2005). The results obtained when DTW is used to classify gestures improve both PCA and PCA-LDA results, due to its ability to deal with temporal shifting. As it is depicted in Table 4.5, the DTW applied over complete curvature functions provides better results, but they are only slightly better than the ones provided by the Dominant Points Detector (DPD)-DTW which, on the contrary, encodes gestures in a more reduced feature vector.

Table 4.5: Evaluation of the different recognition approaches

	<b>PCA</b>	<b>PCA-LDA</b>	<b>DTW</b>	<b>DPD-DTW</b>
1-NN	74 %	82 %	97 %	95 %
3-NN	72 %	77 %	96 %	93 %

As commented above and detailed in section 4.7.5, global reinforcement may be used to improve these results.

## 4.10 Conclusion

In this section the knowledge component of the proposed RLbI architecture has been detailed. There are three main modules in this component that deal with gesture representation, recognition and learning. This thesis contributes in these three different elements.

The representation of perceived gestures is based on features, instead of HMMs, that require *a priori* training phases and are very sensitive to segmentation errors. In this thesis both local and global features are used in a combined representation. Local features are sets of dominant points extracted using an algorithm based on adaptive curvatures. These local representations have been validated by comparing them with other representations. Global features, on the other hand, are simple geometric relations, which usefulness for the proposed system is tested in the recognition stage.

Encoded gestures are recognized using DTW to compute local distances, and a fast analytic equation to compare global features. DTW proves to offer better recognition rates and be more robust than other dynamic programming alignment techniques, for the sequences of dominant points considered in this thesis. The usage of the proposed global features is validated by comparing results obtained before and after adding them to the recognition system. The sensible improvement in the results obtained when only these simple global features are used points towards the possibility of including more complex global features in further work.

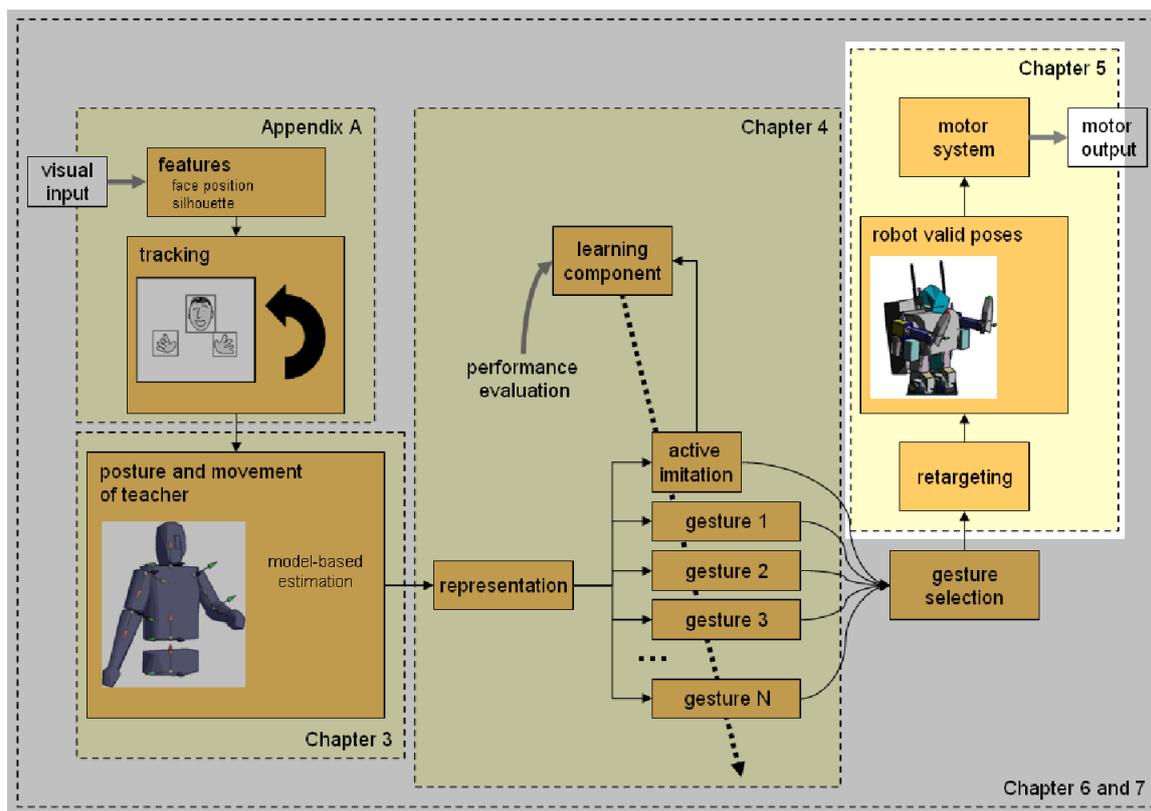
An additional comparison has been performed, between the proposed gesture representation and recognition system, and a traditional classification system based on PCA and LDA. Results of this evaluation show that the proposed method improves the performance and recognition rates of the later, even when *a priori* extensive training phases, not usually found in RLbI scenarios, are allowed.

The proposed architecture, finally, uses a novel supervised learning stage. This element provides satisfactory results, although it is limited by the necessity of human supervision. The inclusion of teachers in the learning process is usual in both biology and robotics. Automatic, unsupervised learning mechanisms could be evaluated in further work, although it would be interesting to consider before whether they are suitable for a social robot or not.



# Chapter 5

## Motion translation



### 5.1 Outline of the chapter

Previous chapters detail how human gestures are captured and encoded by the social robot. Gesture recognition and learning, as explained in chapter 2, are executed in the human motion space thus these processes are not limited by the physical characteristics of the robot. But RLbI does not only require the robot to perceive, encode, recognize and learn human gestures.

It should also be able to *imitate* them using its motor abilities. Thus, human movements have to be translated to robot movements, an operation called *retargeting*. Direct mapping ( $\theta_i^r = \theta_i^p \forall i$ , where  $\theta_i^r$  are the robot joint angles and  $\theta_i^p$  are the person joint angles) may be a reasonable and an easy solution to implement if robot and human bodies are very similar (Safonova et al., 2003; Ude et al., 2004). However, the bodies of the human and the social robot may be very different. In these cases, direct correspondence is usually not only meaningless, but even dangerous for the safety of the social robot, as it may be unable to directly perform perceived human poses. Thus, more elaborated retargeting strategies have to be used in more generic RLbI scenarios. This thesis proposes to consider both end-effector positions and joint angle values, in a combined retargeting approach, to allow transferring perceived motion from human to robot while preserving its main features.

This chapter details the retargeting module, that conforms the last stage in the knowledge component of the proposed architecture, prior to the generation of motor commands. The chapter is organized in the following sections:

- Section 5.2 analyzes the *correspondence problem*, or the problem of transferring the motion from human to robot.
- Section 5.3 details the combined retargeting solution proposed to address the previously described correspondence problem.
- The proposed retargeting strategy has been quantitatively evaluated in several tests, that involved different types of gestures and robot imitators. The results of these tests are presented in section 5.4.
- Section 5.5 concludes the chapter discussing the results obtained in the previous evaluation process, and suggesting the direction of future researches in this particular element of the proposed architecture.

## 5.2 The correspondence problem

In the field of computer vision, given two or more images of the same 3D scene, the *correspondence problem* is to find a set of features in one image that can be identified as the same features in the other images. In stereo vision, the correspondence problem -stereo correspondence- can be defined as the problem of pairing up the features in the left and right images (Scharstein et al., 2002).

While these concepts are related with the proposed system, that also uses stereo vision, in this section the correspondence problem is described in the field of social imitation and learning. In this context, the correspondence problem is to identify a *mapping* between the demonstrator and the imitator bodies (Nehaniv and Dautenhahn, 2002). A solution to this problem involves a translation from perceived motion to imitator movements. This translation has been referred to by some computer animation researchers as *retargeting* (Gleicher, 1998). If the bodies of demonstrator and imitator are very similar the correspondence problem may be obviously solved as body parts, actions and sensory experience can be mapped straightforward. But if the bodies of demonstrator and imitator are different, then a more complex correspondence is required, involving not only one-to-one mapping but also many-to-one, one-to-many and many-to-many mappings. Nehaniv and Dautenhahn (2002) state that direct mapping may not be possible even when the same embodiment is shared, as many different factors that influence the behaviour of both demonstrator and imitator must be taken into account (Fig. 5.1). It is also important to define at which level correspondence is addressed. Nehaniv and Dautenhahn (2002) take account of mapping in the following levels:

- **State.** Simpler approaches may consider only the states of the demonstrator and the imitator, but more complex scenarios involve also the states of objects, other agents and the environment.

When using the previously detailed kinematic representation of the agent, the state of the system (the body, in this case) can be defined as the vector  $S$  containing the values of the DOF in the kinematic chain (Alissandrakis et al., 2007).

- **Actions.** Actions, and sequences of actions, transform the state, including internally generated actions, and external ones, such as sensory stimuli or other events.

If a state is represented using a vector  $S$ , an action  $A$  can be defined as the difference between two state vectors  $S'$  and  $S$ ,  $A = S' - S$ .

- **Effects.** Effects can be defined as changes to the body-world relationship of the agent and/or to positions, orientations, and states of external objects (Alissandrakis et al., 2007).
- **Goals.** Goals are defined as the configuration of state (and/or possibly action sequences) that meet an external or internal criterion.

The retargeting system presented in this thesis is restricted to states and actions. Thus, the proposed system consider not only the sets of joint angles for each frame (states), but



Figure 5.1: Effects of different environments on the same behaviour -lay the box on the table- for two individuals that share the same embodiment.

also the motion that has been performed to change from one state to another. The addition of effects should be considered when the addressed RLbI scenarios involve interaction of the human performer with objects and/or other people. Goals, finally, can only be defined by a superior knowledge layer that lies beyond the scope of this thesis.

Considering the previous mapping levels, [Nehaniv and Dautenhahn \(2002\)](#) propose the following general statement of the correspondence problem, where 'model' refers to 'demonstrator':

*"Given an observed behavior of the model, which from a given starting state leads the model through a sequence (or hierarchy) of subgoals -in states, actions and/or effects, while possibly responding to sensory stimuli and external events, find and execute a sequence of actions using one's own (possibly dissimilar) embodiment, which from a corresponding starting state, lead through corresponding subgoals -in corresponding states, actions and/or effects, while possibly responding to corresponding events."*

In RLbI scenarios, the autonomous robot that is trying to imitate or learning from a human demonstrator must solve this correspondence problem -in other words, it must retarget perceived human motion to its own body. [Safonova et al. \(2003\)](#) propose a three-component retargeting algorithm that considers preservation of oscillations, imitation of body links configu-

ration and joint limits avoidance. They use a one-to-one mapping approach that has been tested in controlled environments, where human motion was captured using optical markers, and using a particular robot which had a body configuration which is very similar to a human body.

While the previous conditions can be met in a particular RLbI scenario, a solution to the correspondence problem in robotic social learning should consider that the *body* of a robot will usually be very different to a human body. In this case, retargeting can not be achieved using a direct (one-to-one), complete mapping. Instead, *partial* mapping appears as an useful approach, in which only some of the perceived states and actions are described for the imitator (Nehaniv and Dautenhahn, 2002; Alissandrakis et al., 2007). For example, a robot that has only a left arm might successfully imitate left hand human gestures, but it may not map right hand perceived gestures as it has no right arm. Alissandrakis et al. (2007) propose to use a *correspondence matrix* to achieve not one-to-one partial mappings. This matrix sets the influence of each DOF of the demonstrator  $\phi_i$  over each DOF of the imitator  $\theta_j$ . Many-to-one, one-to-many and many-to-many correspondences can be easily implemented by using different  $\phi_i$  to influence a certain  $\theta_j$ , or using a certain  $\phi_i$  to influence several  $\theta_j$ . Partial mappings are also achieved by adjusting these influences. The choice of the matrix depends in general on the particular task, and retargeting is achieved by multiplying sequences of states and actions by this matrix. Manual tuning requirements and difficulties to achieve generalization are the main drawbacks of these matrices.

The correspondence problem has not only been addressed in the field of robotic social learning, but also -and mainly- in the field of 3D graphics animation, due to the increasing use of these technologies in films, games or arts. Thus, Choi and Ko (2000) describe a retargeting system that considers states and actions, and that uses the implicit redundancy in the kinematic mapping that relate joint angles and end-effector positions to adjust the solution given by the pseudo-inverse Jacobian and preserve the characteristics of the motion. Shin et al. (2001) propose a real-time, state retargeting system based on the combination of two different factors, that tends to preserve joint angles or end-effector positions, respectively. The weights of these factors in the final solution depends on the proximity to objects or other agents in the scene, as Fig. 5.2 depicts. While the use of a real robot imposes more restrictive experimental constraints, these approaches have inspired the retargeting algorithm proposed for our RLbI scenario.

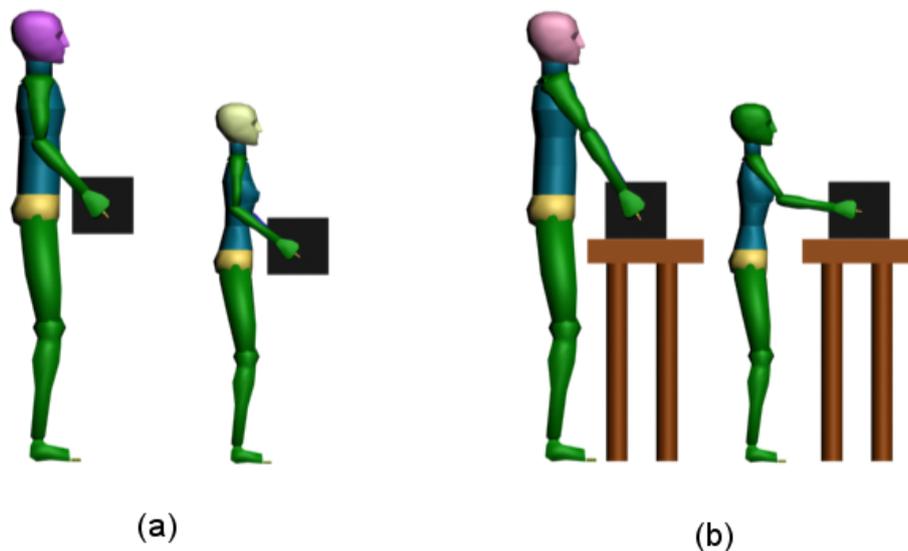


Figure 5.2: Illustration of the combined retargeting approach of Shin et al. (2001): (a) Joint angles preserved in absence of external objects; and (b) End-effector positions preserved when external objects are close.

### 5.3 Combined retargeting

The implemented retargeting system is based on the ideas of Shin et al. (2001), but it considers not only retargeting of state, but also retargeting of actions. Thus, the characteristics of the motion executed to reach a certain state are considered. This approach proved to be useful to retarget a motion *on-line* to very different imitators. It can also be easily adapted to new scenarios or requirements by changing the combined strategies and/or their importance factors. Finally, new retargeting strategies may also be added to extend the system to different applications, increase its imitation possibilities or produce more accurate results for certain movements. It can be concluded from these observations that the combined retargeting has the potential to address the generalization required to adapt to different scenarios and robot body configurations.

The proposed approach extends this importance-based retargeting approach to social robotics imitation scenarios, that usually involve highly constrained information about the performer and require a more careful evaluation of incorrect poses in order to preserve robot safety. It also includes both states and actions, thus retargeted motion is considered a continuous movement instead of a sequence of static poses. As detailed in chapter 3, a human 3D model was used to help obtaining a valid pose from noisy visual information. The retargeting module employs

a robot 3D model to avoid incorrect robot poses. This model uses the same IK algorithm than the human model, and thus *on-line* response is achieved. Safe movements are guaranteed as the movements of the robot model are constrained using real robot joint limits and the 3D meshes associated to the different bones in the kinematic chain correspond to detailed representations of real robot body parts.

Fig. 5.3 presents an overview of the proposed retargeting system. The inputs for this system are data about human pose. These data can be provided by different sources such as optical or magnetic motion capture systems, human motion databases or different perceptual systems. The only requisite for the input data is to include head and hands 3D positions. Additional information about human pose, such as torso orientation, elbow positions, neck orientation, etc. can be also incorporated to restrict and refine the pose of the virtual robot model. As depicted in Fig. 5.3, in this thesis the human pose is provided by the model-based HMC system detailed in section 3.5. Thus, head and hand positions and torso orientation are provided, as well as the complete set of human model joint angles.

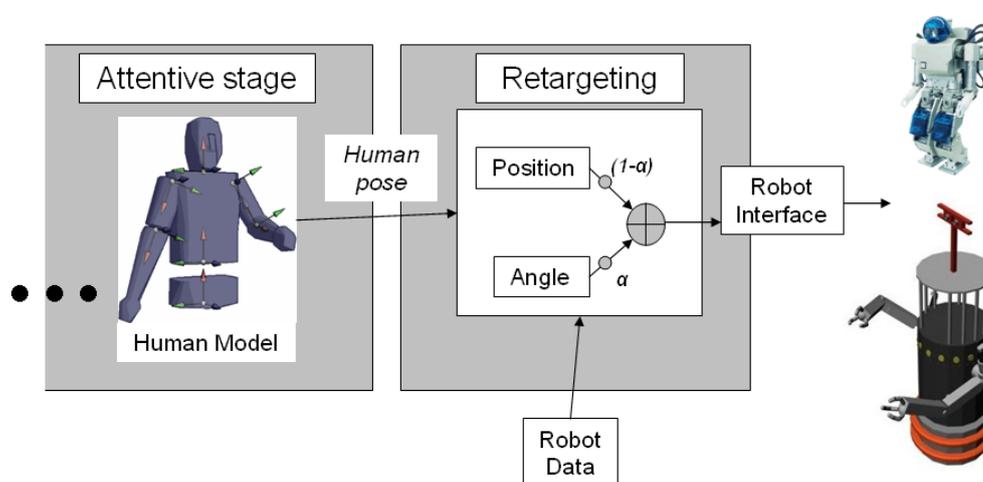


Figure 5.3: Combined retargeting system.

The retargeting module combines two different retargeting strategies to generate a robot pose: i) a position-based retargeting approach; and ii) an angle-based retargeting approach. The amplitude of perceived movements determines how these strategies are combined. As commented above, the retargeting module considers also data about the particular used robot, not only to retarget positions and angles, but also to ensure the robot is moving to a valid and safe pose. Once the set of joint angles for the robotic platform are computed, they are loaded into it using

its particular interface.

### 5.3.1 Position retargeting

Position retargeting translates the position of the human end-effector, or hand,  $\vec{P} = (P_x, P_y, P_z)$  in a reference frame local to the human model to an equivalent position for the end-effector of the robot arm, in a coordinate frame local to the robot. The robot arm pose is described as a set of joint angles  $R_p(\theta_i)$ . Therefore, it has to be executed twice if left and right arm poses are obtained. Fig. 5.4 shows the block diagram of this retargeting module. The first step of this process is shown in Eq. 5.1:

$$\begin{aligned}\vec{P}' &= \vec{P} - \overline{SP} - \overline{OP} \\ \vec{R}' &= \begin{pmatrix} L_r \\ L_p \end{pmatrix} \cdot \vec{P}'\end{aligned}\quad (5.1)$$

$$\begin{aligned}\overline{R}^{\text{ini}} &= \vec{R}' + \overline{SP} + \overline{OP} \\ \overline{R}_p(\theta_i) &= IK(\overline{R}^{\text{ini}})\end{aligned}\quad (5.2)$$

Fig. 5.5 depicts  $\overline{SP}$  and  $\overline{SR}$ , that are the XYZ position of human and robot shoulders, respectively. The upper-body movements, such as the ones considered in the proposed RLbI system, can be referred to the pelvis (Boardman, 2005). Thus,  $\overline{SP}$  and  $\overline{SR}$  are given in coordinates local to coordinate frames attached to human and robot pelvises, respectively. The choice of these reference points is arbitrary and should be made according to the application. The reference could for example lie on an object external to the teacher and the imitator, if a grasping behaviour were to be imitated. In this thesis social upper-body gestures are imitated, and thus the references are located on the waists, that are common reference points in HMC and character animation applications.  $\overline{OP}$  and  $\overline{OR}$  are the global coordinates of the local frames for the human and the robot.  $L_r$  and  $L_p$  are the lengths of the stretched arms of the robot and the human, from end-effector to shoulder, as shown in Fig. 5.5. These lengths are used to scale the positions the robot has to reach.  $\overline{R}^{\text{ini}}$  is a first approximation to the retargeted XYZ position of the corresponding robot end-effector. But, as human and robot bodies are different, a valid pose for the human model can be set to an invalid robot pose if only Eq. 5.1 is used. Thus, the previously obtained position  $\overline{R}^{\text{ini}}$  is fed to the IK module (Bandera et al., 2007) detailed in

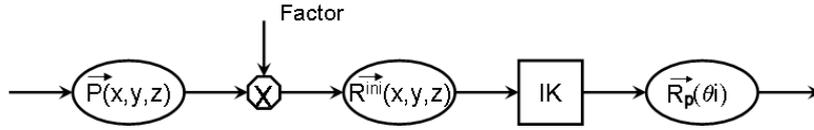


Figure 5.4: Position retargeting.

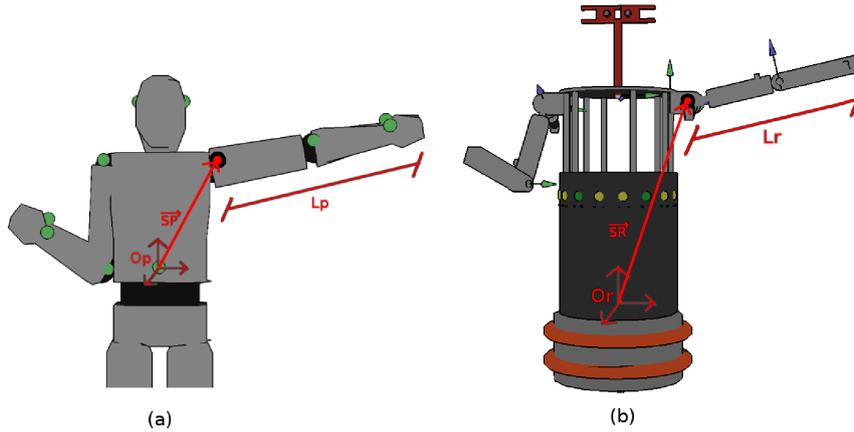


Figure 5.5: 3D models showing the local coordinate frames, the left shoulder position and the length of the stretched left arm for: (a) Human model; and (b) Robot model.

section 3.5. This module takes into account self-collisions and joint limits to obtain an arm pose  $\overrightarrow{R_p}(\theta_i)$  that leaves the end-effector in a position as close as possible to  $\overrightarrow{R}^{ini}$  (Eq. 5.2).

### 5.3.2 Angle retargeting

Fig. 5.6 shows the block diagram of the angle retargeting system used in this thesis. The human model used in the HMC element provides the joint angles of the person,  $\overrightarrow{P}(\theta_i)$ . Then, each of these joint angles is retargeted to the corresponding robot joint angle,  $\overrightarrow{R_a}(\theta_i) = f(\overrightarrow{P}(\theta_i))$ . Instead of direct equivalence, Eq. 5.3 is proposed to perform this retargeting for each  $i$  value.

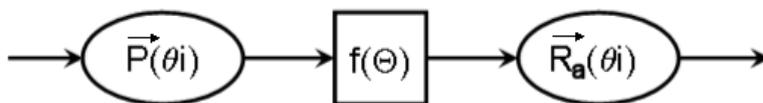


Figure 5.6: Angle retargeting.

$$R_a(\theta_i) = \begin{cases} P(\theta_i) & \text{if } m_i^T < P(\theta_i) < M_i^T \\ \left(\frac{P(\theta_i) - m_i^P}{m_i^T - m_i^P}\right) \cdot (m_i^T - m_i^R) + m_i^R & \text{if } P(\theta_i) < m_i^T \\ \left(\frac{P(\theta_i) - M_i^T}{M_i^P - M_i^T}\right) \cdot (M_i^R - M_i^T) + M_i^T & \text{if } M_i^T < P(\theta_i) \end{cases} \quad (5.3)$$

where  $M_i^P$  and  $m_i^P$  are the limits of person model joint  $i$ ,  $M_i^R$  and  $m_i^R$  are the limits of robot model joint  $i$ ,  $M_i^T = \min(M_i^P, M_i^R)$  and  $m_i^T = \max(m_i^P, m_i^R)$ .

Fig. 5.7 depicts an example in which a human joint angle range of  $[100, 180]$  degrees is adapted to a HOAP-1 robot range of  $[90, 150]$  degrees (see chapter 6 for further details about this robot). When compared with the direct approach that simply truncates the joint movement, this use of a different transformation near limits produces a smoother motion, and helps preserving the characteristics of perceived trajectories.

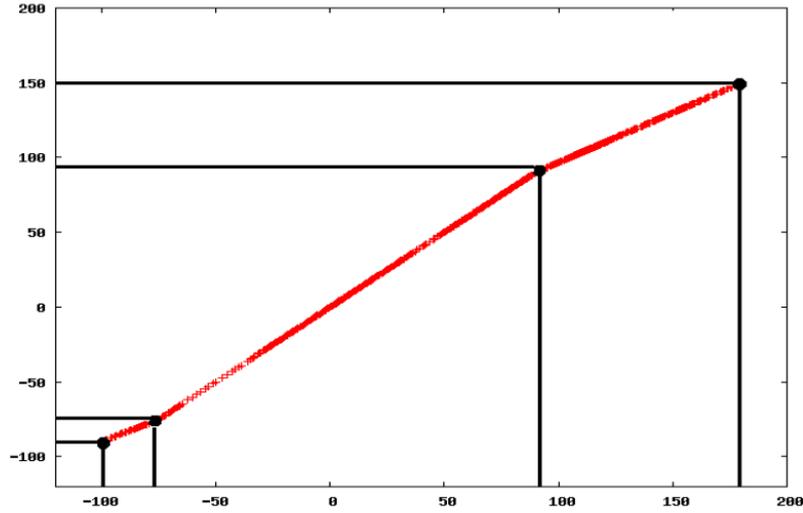


Figure 5.7: Example of angle retargeting function.

### 5.3.3 Combined retargeting

Position retargeting is more adequate when the end-effectors must reach an objective with certain precision, i.e. for pointing or grasping gestures. On the other hand, angle retargeting should be used for gestures in which the information is not in the positions, but in the trajectories, i.e. waving hands to mean 'hello' or dancing.

[Shin et al. \(2001\)](#) proposes a system in which angle and position retargeting are linearly combined. In this thesis a very similar approach is used, in which final joint angles are obtained

as follows:

$$R(\theta_i) = \alpha \cdot (R_a(\theta_i)) + (1 - \alpha) \cdot (R_p(\theta_i)) \quad (5.4)$$

Instead of considering distance to objects, as [Shin et al. \(2001\)](#), in the proposed approach the value of  $\alpha \in [0..1]$  depends on the amplitude of perceived motion,  $A = \|\vec{P}_{\max} - \vec{P}_{\min}\|$ , and it is computed using Eq. 5.5 ([Shin et al., 2001](#)), where  $d$  is a distance limit that can be empirically set to 50 cm. for upper-body hand gestures. Fig. 5.8 shows  $\alpha$  values in this case.

In conclusion, this thesis proposes to retarget static or slow, precise gestures, that are associated to small  $A$  values, using position retargeting. Thus, pointing gestures, static hand signals or manipulation demonstrations that involve constrained movements will preserve end-effector positions. On the other hand, a second group of social gestures like waving hands to mean 'hello' or 'goodbye', or touching one's chest to mean 'me', do not require a very precise end-effector position, but they need to preserve the main characteristics of joint angle trajectories in order to be correctly imitated. These dynamic gestures, in which the  $A$  value is high, will then rely more on angle retargeting. This differentiation satisfies the classification established by [Smyth and Pendleton \(1990\)](#) for human movements, in which they are divided into *location movements*, that may be identified with the prior, and *configured movements*, that includes the later.

As commented above, more different retargeting strategies and conditions can be easily added to the proposed scheme if required. For instance, if interaction with objects is addressed, the proximity criterion employed by [Shin et al. \(2001\)](#) can be incorporated.

$$\alpha = \begin{cases} 3.0 \cdot \left(\frac{A}{d}\right)^2 - 2.0 \cdot \left(\frac{A}{d}\right)^3 & \text{if } A < d \\ 0.0 & \text{otherwise} \end{cases} \quad (5.5)$$

The combined retargeting algorithm is purely analytic. Its complexity does not depend on the amount of perceived data used to pose the human model. Left and right arms are independently retargeted, thus the complexity of the combined retargeting can be depicted as  $O(n)$  where  $n$  is the amount of retargeted limbs. The results of applying this retargeting strategy in real RLBI scenarios are detailed in chapter 6.

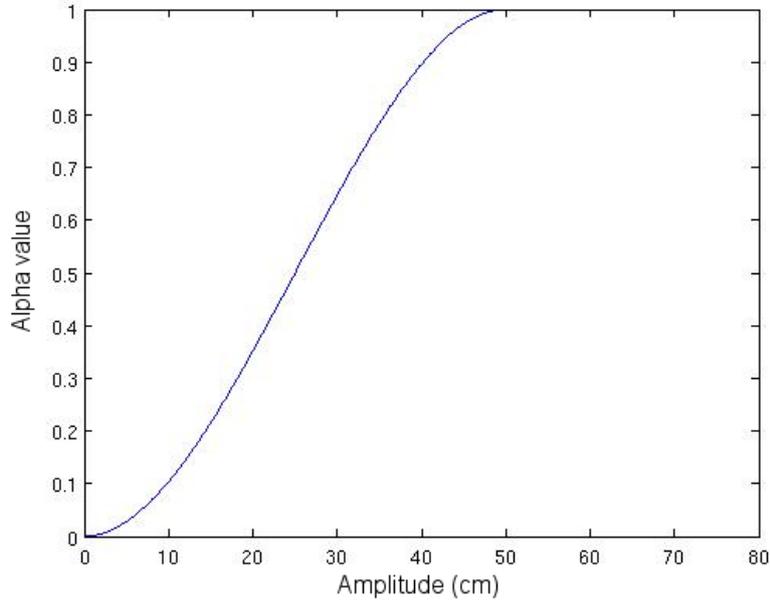


Figure 5.8:  $\alpha$  values ( $d=50$  cm.).

## 5.4 Evaluation of the retargeting module

As detailed in section 5.3, given a certain human hand position  $\vec{P}$ , the retargeting element proposed in this thesis obtains a final set of robot joint angles, for the corresponding robot arm, that imitates the perceived pose. This set of joint angles,  $\vec{R}(\theta_i)$ , is obtained by combining position and angle retargeting approaches. The inputs of the retargeting system for the experiments performed in this chapter are human poses obtained using the HMC system detailed in chapter 3. The output is the set of robot joint angles,  $R(\theta_i)$ . In order to test the validity of the retargeting system, perceived motion is retargeted to two different robotic platforms: a HOAP-1 from Fujitsu and NOMADA, a social robot that is currently being developed in our research group. These platforms are described in chapter 6. Here it is only important to remark that these robots present significant differences with respect to a human model, and also with respect to each other, regarding kinematics and reachable workspace. In order to ease the experiments and isolate retargeting errors from other error or noise sources (e.g. the motor controllers) ideal virtual models of the robots are used, instead of the real platforms.

Once a certain pose has been retargeted from the human demonstrator to the particular robot imitator, the accuracy of the proposed retargeting approach is evaluated using two different error measurements: i) the error in the final position  $\vec{R} = (R_x, R_y, R_z)$  reached by the robot end-effector; and ii) the error in its final joint angles  $R(\theta_i)$ . These errors are computed using

Table 5.1: Right arm mean errors ( $E(q)$ ) and standard deviations ( $\sigma_q$ ) for an imitation sequence performed by HOAP-1 robot.

$\alpha$ value	$E_x$ (cm.)	$E_y$ (cm.)	$E_z$ (cm.)	$E(\theta_0)$ (degrees)	$E(\theta_1)$ (degrees)	$E(\theta_2)$ (degrees)	$E(\theta_3)$ (degrees)
0.0	0.020	0.044	0.110	18.307	14.875	16.355	16.043
1.0	0.104	0.083	0.152	0.085	3.307	0.000	10.200
$\alpha$ value	$\sigma(E_x)$ (cm.)	$\sigma(E_y)$ (cm.)	$\sigma(E_z)$ (cm.)	$\sigma(\theta_0)$ (degrees)	$\sigma(\theta_1)$ (degrees)	$\sigma(\theta_2)$ (degrees)	$\sigma(\theta_3)$ (degrees)
0.0	0.029	0.043	0.048	10.096	9.768	12.062	10.064
1.0	0.080	0.065	0.070	0.331	6.045	0.000	9.401

Eq. 5.6 and Eq. 5.7.

$$\vec{E} = \vec{R} - \vec{R}^{\text{ini}} = FK(IK(\vec{R}^{\text{ini}})) - \vec{R}^{\text{ini}} \quad (5.6)$$

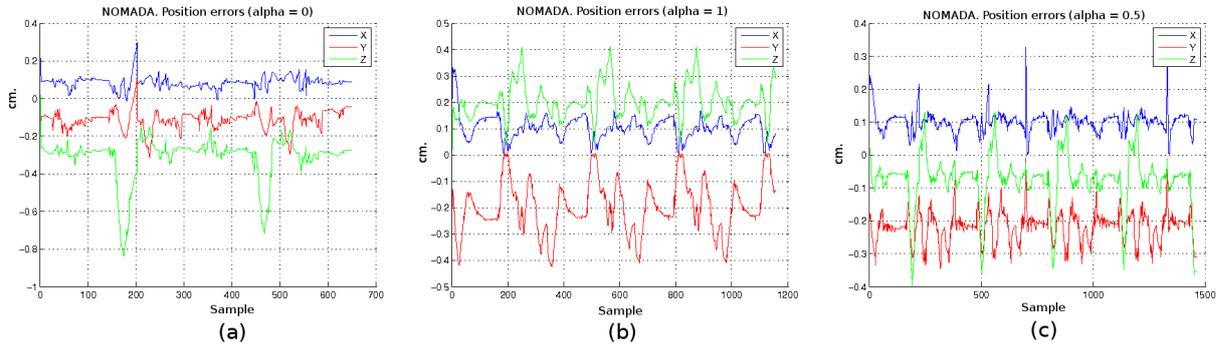
$$E(\theta_i) = R(\theta_i) - P(\theta_i) \quad (5.7)$$

In order to compute the position errors ( $\vec{E} = (E_x, E_y, E_z)$ ) the end-effector robot positions have been obtained by applying Forward Kinematics (FK) to the robot final set of joint angles, thus  $\vec{R} = FK(R(\theta_i))$ . Then, these positions  $\vec{R}$  are scaled to human height, and human model hand positions  $\vec{P}$  are subtracted to these normalized values. On the other hand, joint angle errors  $E(\theta_i)$  are obtained by simply subtracting robot joint angles  $R(\theta_i)$  and human joint angles  $P(\theta_i)$ .

Table 5.1 shows the right arm retargeting errors when using the HOAP-1 robot model to imitate different gestures. As commented above, these gestures were recorded using the same external pair of Videre stereo cameras that were mounted in the NOMADA robot, as the cameras mounted in the HOAP-1 head were not able to capture gestures in which arms are stretched. As expected, position retargeting ( $\alpha=0.0$ ) reduces position errors, but increases joint angle errors. This is mainly due to body differences between the human and the robot, that force the robot to adopt a different configuration to reach a certain end-effector position. Angle retargeting ( $\alpha=1.0$ ), on the other hand, produces very small joint angle errors, but greater position errors.  $\theta_3$ , that corresponds to the DOF at the elbow joint, presents noticeable mean errors in any case, as the HOAP-1  $\theta_3$  joint limit is set to only 90 degrees in flexion, while human elbows usually can bend up to 145 degrees.

Table 5.2: Right arm mean errors ( $E(q)$ ) for dynamic gestures imitated by NOMADA robot.

$\alpha$ value	$E_x$ (cm.)	$E_y$ (cm.)	$E_z$ (cm.)	$E(\theta_0)$ (degrees)	$E(\theta_1)$ (degrees)	$E(\theta_2)$ (degrees)	$E(\theta_3)$ (degrees)
0.0	0.086	-0.112	-0.003	50.554	14.057	-24.312	-45.210
combined	0.101	-0.195	0.203	20.031	3.267	7.140	-34.955
1.0	0.104	-0.197	0.482	0.065	1.597	0.004	-34.533

Figure 5.9: Position errors (Right hand): (a)  $\alpha = 0.0$  (position retargeting); (b)  $\alpha = 1.0$  (angle retargeting); and (c)  $\alpha = 0.5$  (combined retargeting).

Tables 5.2 and 5.3 show the results obtained when position retargeting, angle retargeting or combined retargeting are used to make the NOMADA robot imitate dynamic and static gestures, respectively. Position and joint angle instant errors for one of these -dynamic- gestures are also depicted in Fig. 5.9 and Fig. 5.10. The dynamic gestures used in this test are the ones conforming the database of social gestures further detailed. Static gestures include pointing and touching gestures, in which the position of the end-effector is the key information. As before, NOMADA  $\theta_3$  joint limit is only 90 degrees, thus errors in this joint are much longer than expected due to human arm moving to unreachable positions for the robot. The combined retargeting strategy uses an  $\alpha$  value computed using Eq. 5.5.

As expected, angle retargeting preserves joint angle variations and thus it is more adequate for dynamic gestures, while position retargeting is better suited to static gestures. The proposed combined approach is able to adapt to each particular situation, offering a more powerful tool when imitating generic gestures.

In order to provide a qualitative illustration of these tests, Fig. 5.11 shows results obtained when the retargeting system is used to imitate one certain social gesture ('hello'). This figure illustrates both the ability of the retargeting system to adapt to different robotic platforms, and the different results obtained when different retargeting strategies are chosen.

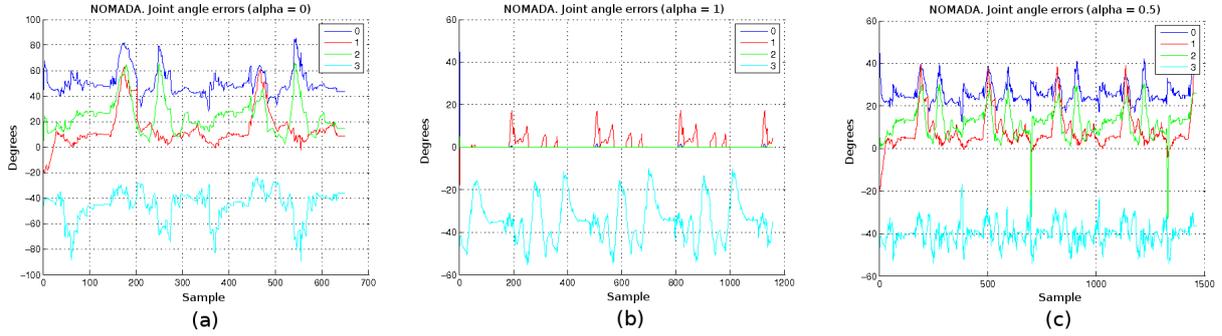


Figure 5.10: Joint angle errors (Right arm): (a)  $\alpha = 0.0$  (position retargeting); (b)  $\alpha = 1.0$  (angle retargeting); and (c)  $\alpha = 0.5$  (combined retargeting).

Table 5.3: Right arm mean errors ( $E(q)$ ) for static gestures imitated by NOMADA robot.

$\alpha$ value	$E_x$ (cm.)	$E_y$ (cm.)	$E_z$ (cm.)	$E(\theta_0)$ (degrees)	$E(\theta_1)$ (degrees)	$E(\theta_2)$ (degrees)	$E(\theta_3)$ (degrees)
0.0	0.074	-0.084	-0.002	44.281	9.957	17.220	-40.428
combined	0.075	-0.087	-0.001	41.001	8.154	14.129	-41.503
1.0	0.127	-0.389	0.453	0.000	0.000	0.000	-46.831

As depicted, preserving joint angles leads to limited movements if robot joint limits are more restrictive than human ones (as occurs when using NOMADA or HOAP-1 arms). The position retargeting strategy, on the other hand, tends to produce odd arm poses as certain positions are only reachable by the robot if joint angles are moved to unnatural values (i.e. the HOAP-1 right arm depicted in Fig. 5.11.b). It is also more usual, when using this strategy, that a certain arm does not move at all as the complete perceived motion lies beyond the robot reachable positions (i.e. the HOAP-1 left arm in Fig. 5.11.b). The use of a combined retargeting strategy, in which the  $\alpha$  value is dynamically changed according to perceived motion amplitudes, improves the obtained results, as can be qualitative appreciated in Fig. 5.11.

## 5.5 Conclusion

The evaluation of the retargeting module has involved two different virtual robots, that have been used as imitators. These virtual models share with their real counterparts joint limits, kinematics relations and shape. These models are different from each other and also different from the human demonstrator. Both static and dynamic gestures have been used to test the retargeting module. The results show that the proposed combined retargeting strategy is able

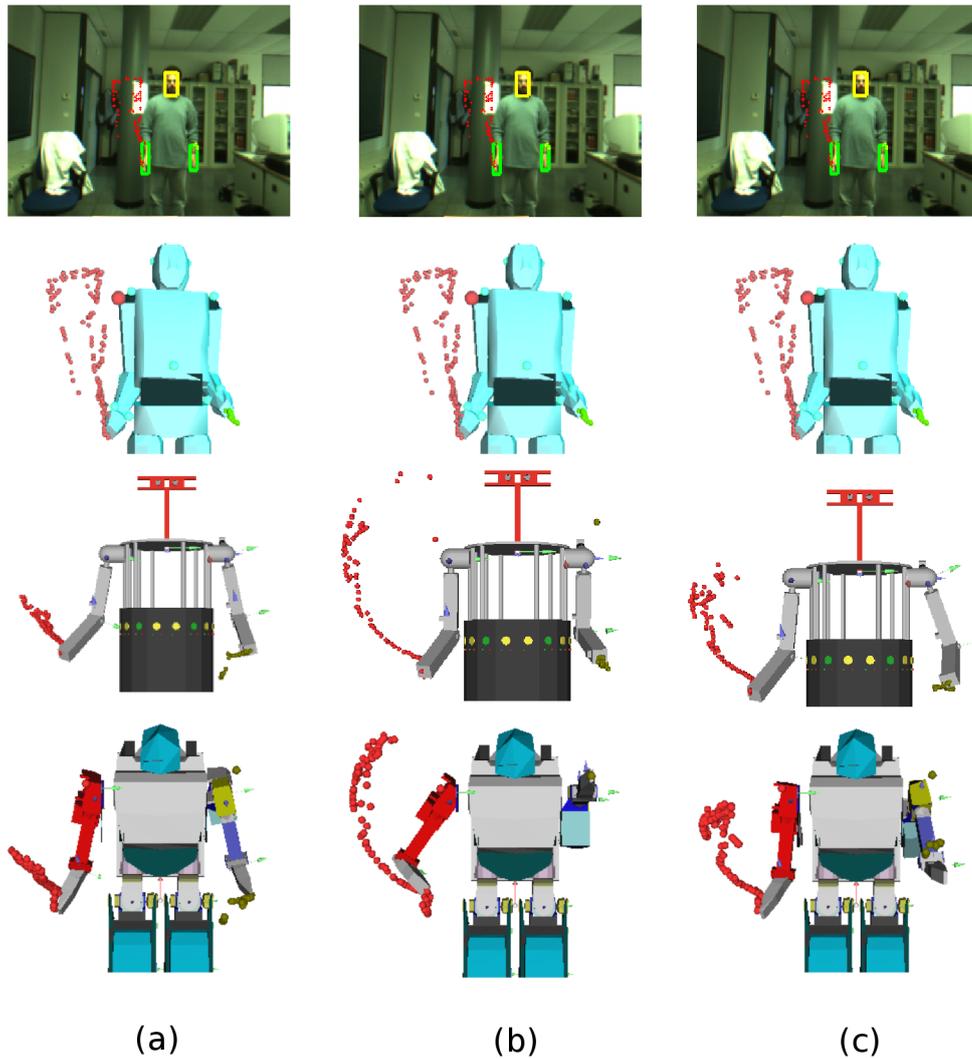


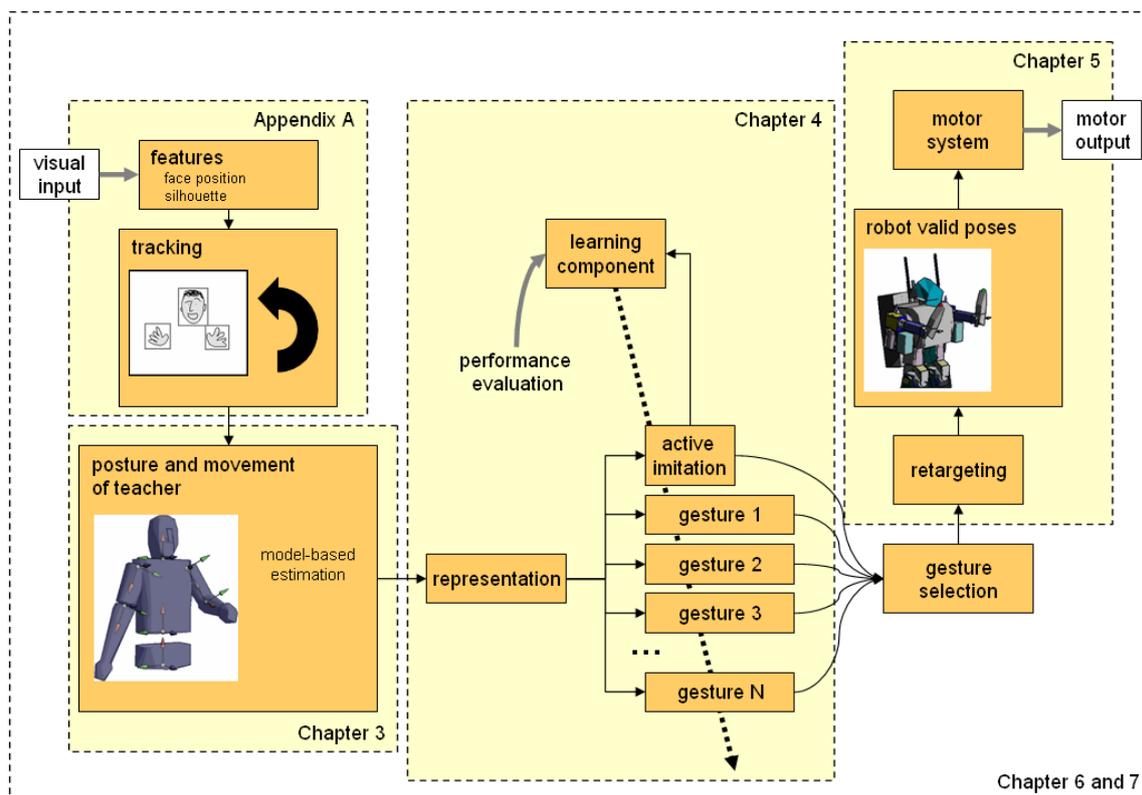
Figure 5.11: Perceived movement retargeted to two different robotic platforms using different retargeting strategies: (a)  $\alpha = 1.0$  (angle retargeting); (b)  $\alpha = 0.0$  (position retargeting); and (c) dynamic  $\alpha$  (combined retargeting).

to adapt to different robotic platforms, and also to different types of gestures. It may be interesting to extend the  $\alpha$  computation adding environmental information, or to incorporate new retargeting criteria to the combined strategy. Thus, it may be concluded that the proposed retargeting strategy is adequate for the considered gesture demonstration situations, while it has the advantage of being easily extensible to more complex scenarios if required.



## Chapter 6

# Testing the learning by imitation system



### 6.1 Outline of the chapter

This chapter describes the experiments that have been conducted over virtual and real robotic platforms to test the proposed RLbI system. Some of the algorithms detailed in previous chapters

had to be modified when marker-based motion detection was replaced by stereo vision, and controlled environments by dynamic, real indoor environments. These modifications are detailed in this chapter, along with the results obtained in the different conducted experiments. The chapter is organized as follows:

- Section 6.2 details the two different robotic platforms that have been used to test the RLbI system.
- Section 6.3 describes the experimental setup used along these tests.
- Finally, section 6.4 presents the results of the experiments. It is organized as follows:
  - The results obtained when the gesture segmentation algorithm is applied to visually perceived trajectories are firstly presented in this section.
  - Chapter 4 described the proposed gesture recognition and learning algorithm. However, in that chapter experiments were conducted using a commercial motion capture system based on active markers. Here the proposed recognition algorithm is applied to gestures perceived using only stereo vision. The results of these tests are discussed and compared against previous ones.
  - Finally, the section details the results obtained when the RLbI system is used in a real human-robot interaction scenario, in which uncontrolled environment, untrained users and no prior knowledge about performed gestures are considered.

## 6.2 Robotic platforms used in testing

This section describes the robotic platforms that have been used to perform the experiments. The first of these platforms, the HOAP-1 from Fujitsu, is a miniature robot that allowed to test the first version of the proposed RLbI system, presented in section 2.5 [Bandera et al. \(2006\)](#). The characteristics of this robot, however, make it not suitable to be used in real, uncontrolled human environments, thus a new social robot is currently being developed in the ISIS research group, where this thesis has been elaborated. A virtual model of this robot, and some of its elements (the head and the right arm) are already available. The stereo cameras to be mounted in the robot are used to perceive. Final motion experiments have been conducted over the virtual robot. The movements of the virtual robot right arm are sent to the available real robot right arm to test this *hardware*.

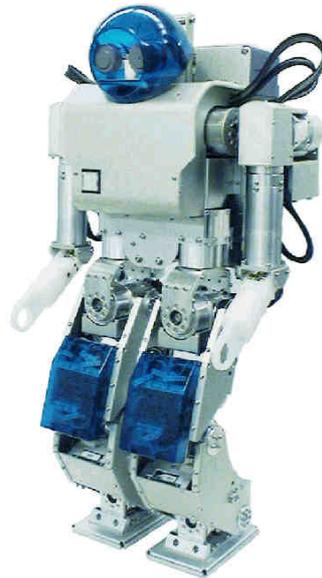


Figure 6.1: HOAP-1 humanoid robot.

### 6.2.1 HOAP-1

The main characteristics of the HOAP-1 from Fujitsu (Fig. 6.1), where HOAP stands for *Humanoid for Open Architecture Platform*, are listed below:

- 20 DOF. Most of these DOF are located in the legs. Only the 8 DOF depicted in Fig. 6.2 are used in the considered upper-body RLbI scenarios, while the rest of the joints are set to a constant position to support the robot in a stable, upright posture.
- 6 Kg.
- 48 cm. tall.
- Four pressure sensors in each foot.
- Accelerometer and gyro.
- The original HOAP-1 had no vision system. A pair of color stereo cameras, with parallel optic axes, was mounted in its head when it was received at ISIS group. This stereo system has been deeply explained in section 3.6.

While the HOAP-1 robot is an useful platform to study biped locomotion and stability criterions [Carmona et al. \(2007\)](#), it does not fit the requirements of a social robot for a number

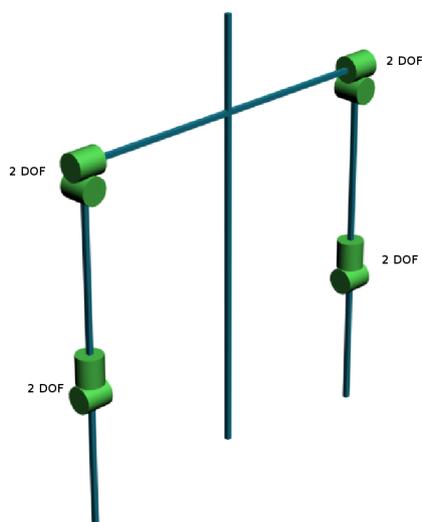


Figure 6.2: DOFs for the upper torso of the HOAP-1.

of reasons, including:

- *Size.* It is difficult for a robot measuring only 48 cm. to help people solving everyday tasks. It is also complex for such a robot to interact with people as it has no neck joints that would allow it to look up. Thus, it is necessary to put the robot in a table or use any other special environment to perform RLbI using this robot [Calinon \(2007\)](#).
- *Strength.* The HOAP-1 robot is not able to carry many objects that are commonly used in real human environments, and that are expected to be handled by a companion robot (i.e. trays, jars, baskets,...).
- *Speed.* Even if the most advanced biped locomotion algorithms are used to make HOAP-1 walk around, it would be very difficult for this robot to move at human walking speeds, due to the structure and mechanic characteristics of its legs.
- *Stability.* It is easy for the HOAP-1 robot to fall down if any problem occurs during locomotion. Such problems are more common in dynamic environments as the one considered in this thesis.
- *Stereo baseline.* The small head of the HOAP-1 can only contain a stereo pair with a very limited baseline. Such a small baseline is not adequate to correctly perceive social gestures.

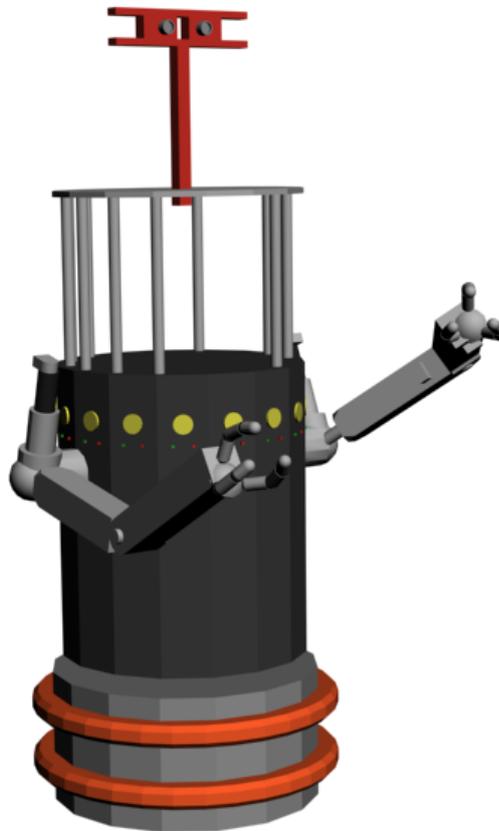


Figure 6.3: Virtual model of the NOMADA social robot.

### 6.2.2 NOMADA

The previous issues suggested the use of a different platform to implement a social robot. It is possible to acquire different models of social robots in the market, e.g. the HRP-2 from Kawada, that has announced its HRP-3 to be released in 2010. But these platforms are expensive (HRP-3 is expected to cost more than 90,000 euros) and it may take a long time to repair a broken piece or find replacements or updates.

It was decided instead that one of the robotic platforms available in the ISIS research group could be used to build a social robot that will meet the requirements of research in social robotics. This robot, named NOMADA, is still under development. It will be composed by different parts organized as sets of subsystems that are connected via different communication buses. Only the right arm and the stereo head have been already built. Fig. 6.3) shows a virtual model of the complete robot, that has been used to test the RLbI system.

### 6.2.2.1 Locomotion platform

There are many social robots that are equipped with a pair of legs to walk around (Asimo from Honda, HRP-2 from Kawada or Qrio from Sony). NOMADA platform, on the contrary, uses an holonomic locomotion system composed by three wheels. This characteristic makes not possible for the NOMADA to step up or down stairs, or to move in certain rough or narrow environments. On the other hand, the use of a wheeled locomotion system provides a higher stability, simplifies navigation and odometry calculations and also reduces power consumption and mechanical complexity. Besides, most real indoor environments can be completely navigated around using a wheeled platform. Thus, NOMADA finally will be equipped with a wheeled locomotion system, as other companion and social robots such as Robonaut from NASA, HRP-2W from Kawada or MDS (Mobile/Dexterous/Social) from MIT.

The locomotion platform of NOMADA uses the sensors, actuators and mechanical structure of the NOMAD 200 from Nomadic (Fig. 6.4). The *firmware* (*hardware* and *software*) has been completely replaced in the ISIS research group by newer components. Three DOF are located in the NOMADA body: two of them are related to locomotion and indicate the magnitude and direction of the velocity vector. The third one is located in the waist as depicted in Fig. 6.5. This DOF allows the robot to rotate its torso so it is able to move its field of view to a certain direction while it is moving to another.

As depicted in Fig. 6.3, some elements are added to attach arms, additional sensors, and the robot vision system, that is finally located at a height of around 160 cm.

### 6.2.2.2 Articulated arms

The arms of NOMADA (Fig. 6.6) have 4 DOF. Three of them are located in the shoulder, and one in the elbow, as Fig. 6.5 shows. Each of these DOF use commercial MAXON controllers that allows precise movements. The current arm version has a strong limitation in the elbow movement, that can only flex 90 degrees. The arm design is currently being modified to allow higher limits in this joint (around 150 degrees). While the current arms have not end-effector actuators, a gripper system will be further developed to conform two 'hands' for the social robot, that will be able to grasp and manipulate objects.

### 6.2.2.3 Perceptual system

The perceptual system of the NOMADA will be composed by the following elements:



Figure 6.4: NOMAD 200 from Nomadic.

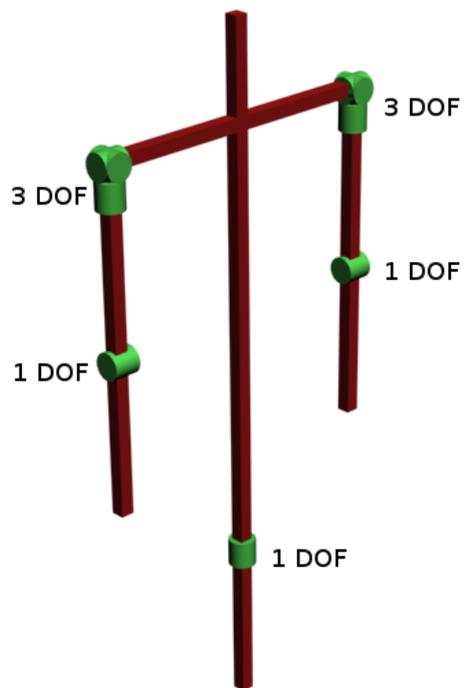


Figure 6.5: DOFs for the upper torso of the NOMADA.

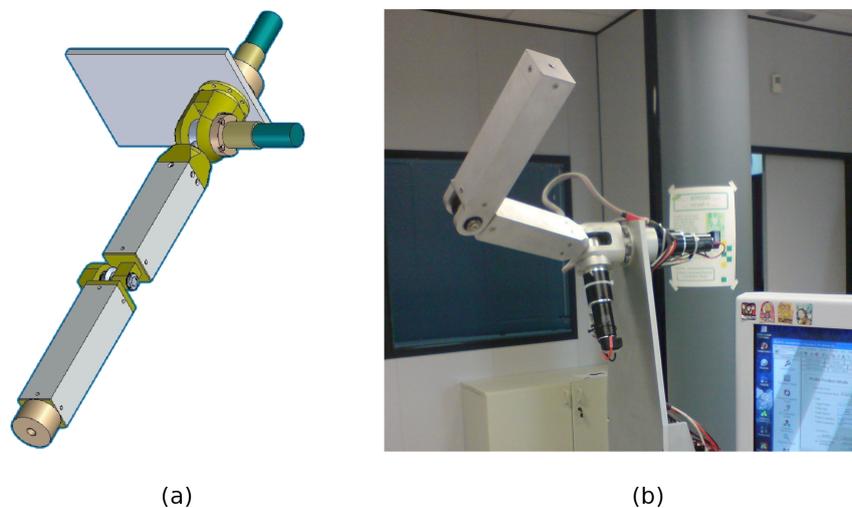


Figure 6.6: (a) Virtual model of the NOMADA arm; and (b) real NOMADA arm mounted by the ISIS research group at Malaga University.

- Bumpers ring to detect collisions.
- Infrared sensors ring to detect close obstacles.
- Sonar sensors ring to detect obstacles at medium or long distances.
- SICK LD OEM1000 360° laser to obtain precise distance measurements. As discussed later in this thesis, a laser range finder may be an interesting alternative to distance computation based on stereopsis.
- KVH C100 compass.
- Stereo vision system. This system is composed by a pair of static STH-DCSG-VAR-C cameras from Videre Design (Fig. 3.39). The height of the NOMADA robot allows these static cameras to correctly capture human motion. In the future this pair of stereo cameras will most probably be replaced by a Biclops robotic head from Metrica, that is able to perform pan, tilt and vergence movements (Fig. 6.7).

### 6.3 Experimental setup

Chapter 3 offered a quantitative evaluation of the vision-based HMC system used to perceive human movements. The same pair of stereo cameras that were used for this evaluation process are used in this chapter to test the performance of the complete RLbI system.



Figure 6.7: Biclops pan-tilt-vergence head.

Different experiments have been performed in order to analyze the different steps that lead to correctly imitate a perceived motion. All these experiments use a dataset consisting of 53 upper-body gestures performed by six different people. For each of these gestures, the motion of the left and right hands is recorded at an average sampling rate of 15 Hz. The average amount of samples per gesture is 103.5, thus each gesture, composed by two trajectories, is characterized by an average of 207 XYZ values.

The gestures in the dataset are different executions of 8 upper-body gestures, that are commonly found in social interaction scenarios. Table 6.1 gives their description, and indicates whether they are performed using one hand or both hands. Fig. 6.8 shows the trajectories perceived for one execution of each of the different types of gesture that compose the dataset. As these eight gestures are among the ones used in chapter 4, it is possible to compare these visually perceived trajectories against the ones captured using the Codamotion system (Fig. 4.6). It can be seen that visually perceived gestures are noisier and more irregular.

The people who performed the previously detailed gestures stood in front of the vision-based system of the NOMADA robot. The distance between the performer and the cameras varies from 1.30 to 1.80 meters. No specific clothes were used to perform the experiments. Fig. 6.9 shows some of the different lab rooms in which these tests were executed. As depicted, these locations correspond to real indoor environments, that changes dynamically. Thus, lighting changes, people walking around during experiments or environment variations (i.e. chairs or objects moved from one place to another) occurred during the execution of the gestures.

The first series of experiments test the ability of the proposed system to recognize a type of gesture that has already been stored in the knowledge database of the robot. Thus, the system is provided with three demonstrations of each particular gesture. These demonstrations

Table 6.1: Description of the social gestures performed to test the RLbI system.

Gesture name	Gesture description
Shake hands	Shake hand gesture. The performer offers his/her right hand. Sometimes he/she shakes it
Hi	The performer waves his/her right hand near his/her face
Attention	This movement is similar to Hi, but the movement is made with both hands
Left	The performer points left with his/her left hand
Right	The performer points right with his/her right hand
Shrug	Shoulders shrug and right and left hand moves up slightly
Fish	Both hands move forward and perform a parallel little up-down movement. Different distance between hands ('fish sizes') were considered
Me	The performer reaches his/her chest with his/her right hand

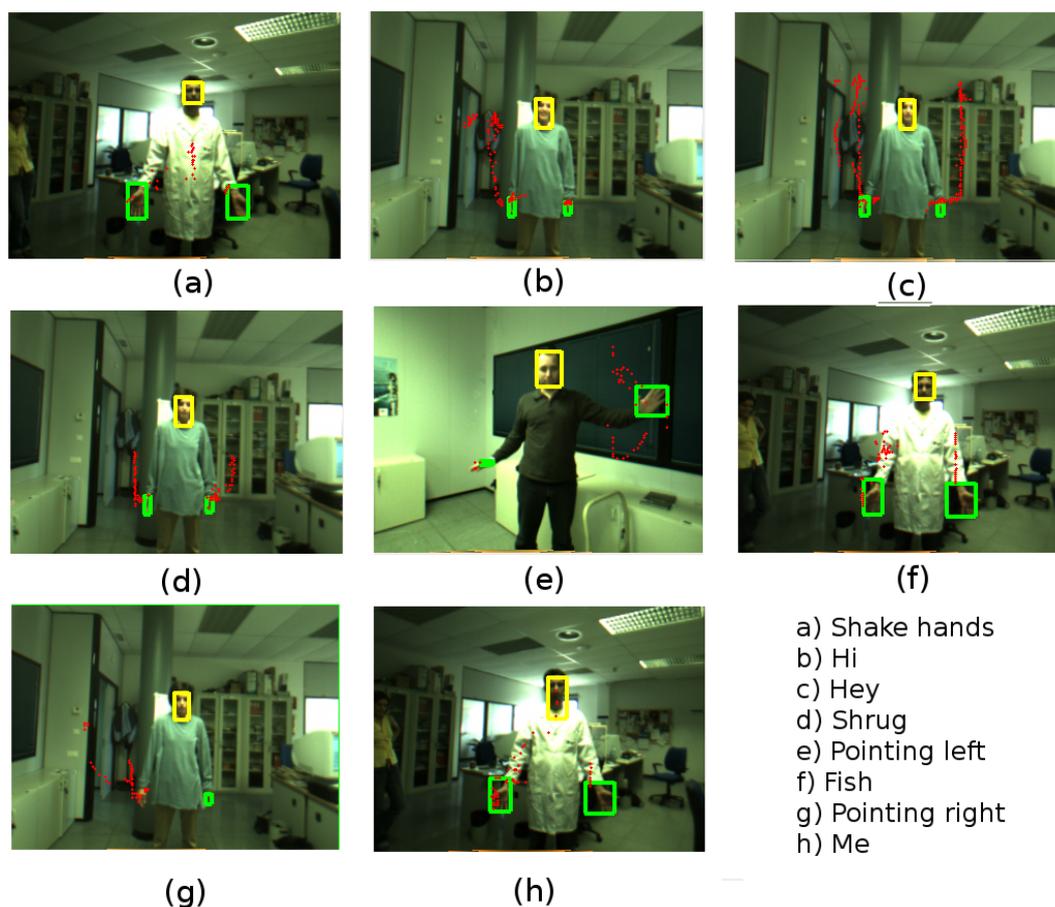


Figure 6.8: Upper-body social gestures used to test the proposed RLbI system. The trajectories of the left and right hand have been marked over the left frame.



(a)



(b)



(c)

Figure 6.9: Real indoor scenarios used to test the proposed HMC system.

are correctly classified and labeled. Then, the remaining 29 gestures in the dataset are fed to the gesture recognition system. The results obtained against the recognition rates achieved in chapter 4. The use of the proposed HMC system is expected to lead to worse results when compared against the Codamotion system. If the recognition system is robust enough, however, this difference should be constrained to low values.

The final experiments test the complete RLbI system in a real HRI scenario. In these experiments, the robot has no initial knowledge about the gestures that are going to be performed by different people. The robot should perceive these gestures, confront them against the already perceived ones and recognize them. New gestures will be incorporated *on-line* to the database using the criterions detailed in section 4.8. The imitative abilities of the robot are also test during this process. Thus, gesture is retargeted *on-line*, while it is being perceived. The virtual model of the robot is used to check the validity of the retargeted motion. Finally, the movements of the right arm are feed *on-line* to the available robot arm to test the output provided by the proposed RLbI architecture.

## 6.4 Experimental results

As detailed above, this chapter tests the validity of the RLbI system proposed in this thesis. This validation is achieved by two different sets of experiments that test respectively the gesture recognition system and the *on-line* gesture perception, recognition and learning process. This section details the results obtained in these experiments.

### 6.4.1 Gesture representation

The adaptive curvature algorithm used to represent perceived motion was validated in chapter 4. Sequences of hand movements captured at 100 Hz, using the Codamotion CX1 system, were used in these experiments. In order to evaluate the complete architecture, in this chapter the same experiments are conducted but using the detailed pair of stereo cameras to capture hands motion, at a sampling rate of 15 Hz.

Table 6.2 shows the results obtained for these tests. It can be seen that, as expected, the use of stereo vision increases the ISE with respect to table 4.1, in chapter 4. On the other hand, the comparative results presented in tables 4.1 and 6.2 are very similar. Thus, when using stereo vision, the proposed representation again offers better results than methods based on fixed  $k$  values. Its FOM is slightly superior to the one obtained by the CSS, and it is faster. It can be

concluded, after these tests, that the proposed representation based on adaptive curvatures is adequate to be used in RLbI scenarios.

Table 6.2: Compression rates and execution times for different trajectories.

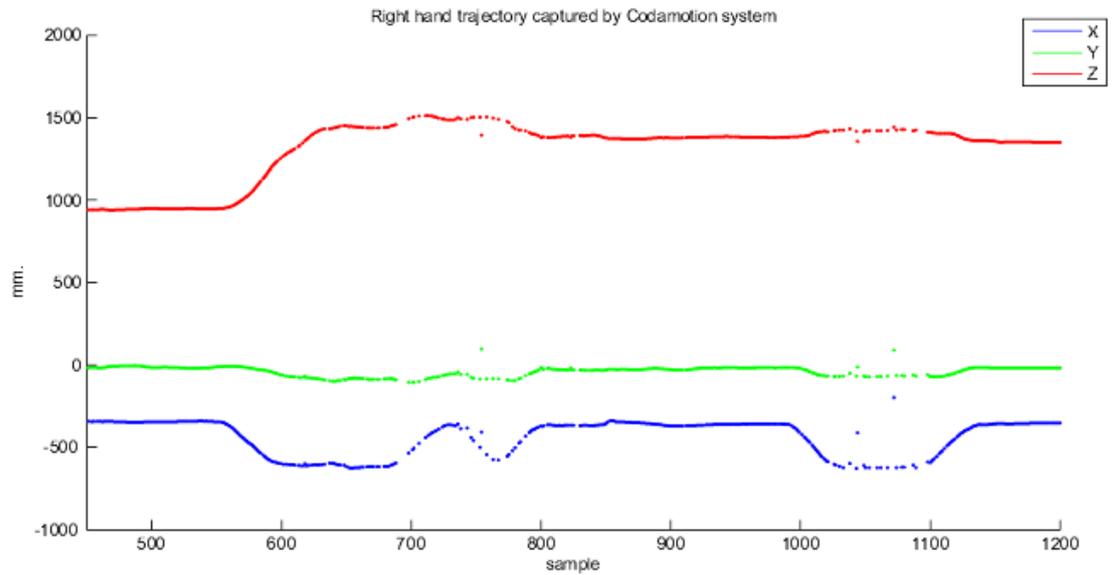
	$\overline{n^\circ frames}$	Hz	$\overline{CR}$	$\overline{ISE} \cdot 10^{13}$	$\overline{FOM}$	$\overline{msecs}$
Adaptive Curv.	5437	15	339	1.26	$2.70 \cdot 10^{-11}$	229.75
K=50	5437	15	181	0.85	$2.14 \cdot 10^{-11}$	193.45
K=100	5437	15	267	1.21	$2.38 \cdot 10^{-11}$	205.32
K=500	5437	15	630	2.51	$2.51 \cdot 10^{-11}$	207.15
CSS	5437	15	420	2.05	$2.55 \cdot 10^{-11}$	810.01

### 6.4.2 Gesture recognition

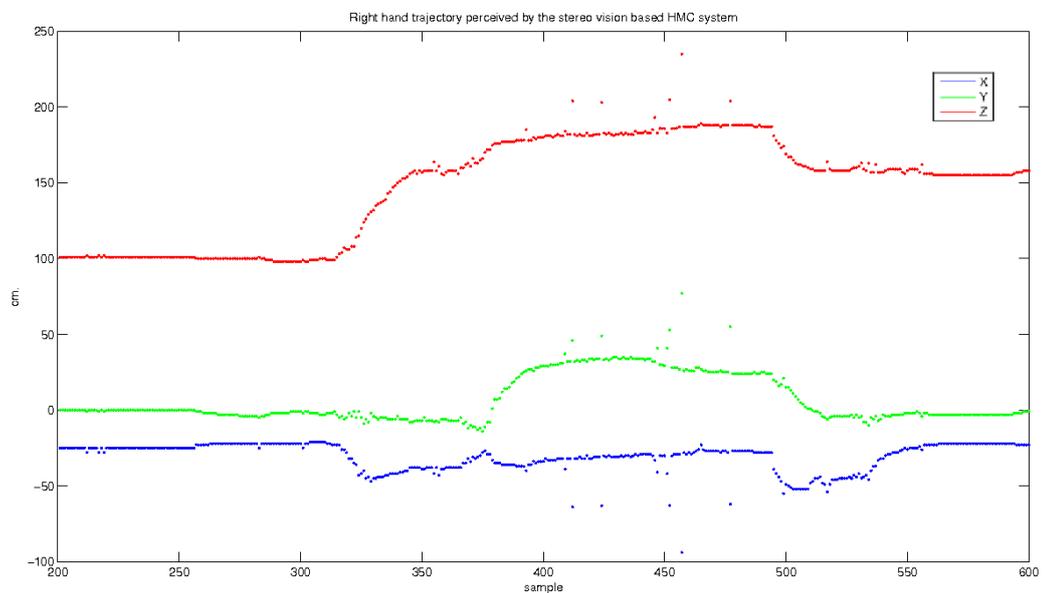
Chapter 4 validates the proposed gesture recognition and learning system. In that chapter the test gestures were captured using the Codamotion CX1 system, that recorded the motion of 13 markers, attached to the human performer, at a frame rate of 100 Hz. The HRI system considered in this thesis, however, has to deal with a much more limited input. Only the trajectories of the face and the hands, and the human silhouette, are directly perceived by the pair of stereo cameras mounted on the head of the robot. As it has been described in the previous chapter, the proposed HMC system is able to offer an estimation of the upper-body pose of the perceived human from this limited input data, being the average position errors constrained enough as to allow to recognize a set of social gestures.

The poses perceived using the stereo vision system are usually noisier and rougher than the ones captured using the Codamotion CX1 system. Figs. 6.10 and 6.11 show two examples of the same motion captured using both the Codamotion system and the stereo vision system. The first remarkable difference is the higher amount of outliers that appear when the stereo vision system is used. On the other hand, the number of samples collected by this perceptual system is also smaller. But probably the most important issue that affects the vision-based HMC are the disparity errors that can be appreciated in the Y coordinate of Figs. 6.10 and 6.11, e.g. the variation in the Y values depicted in Fig. 6.10(b) from samples 400 to 500. Data captured by the Codamotion system (Fig. 6.10(a)) show that there was nearly no variation in the Y coordinate in these frames.

The matching algorithm detailed in chapter 4 may find difficulties in correctly recognizing gestures under these circumstances. Some of the evaluated distance functions are specially sensitive to noise and roughness, thus it may be interesting to extend the gesture recognition

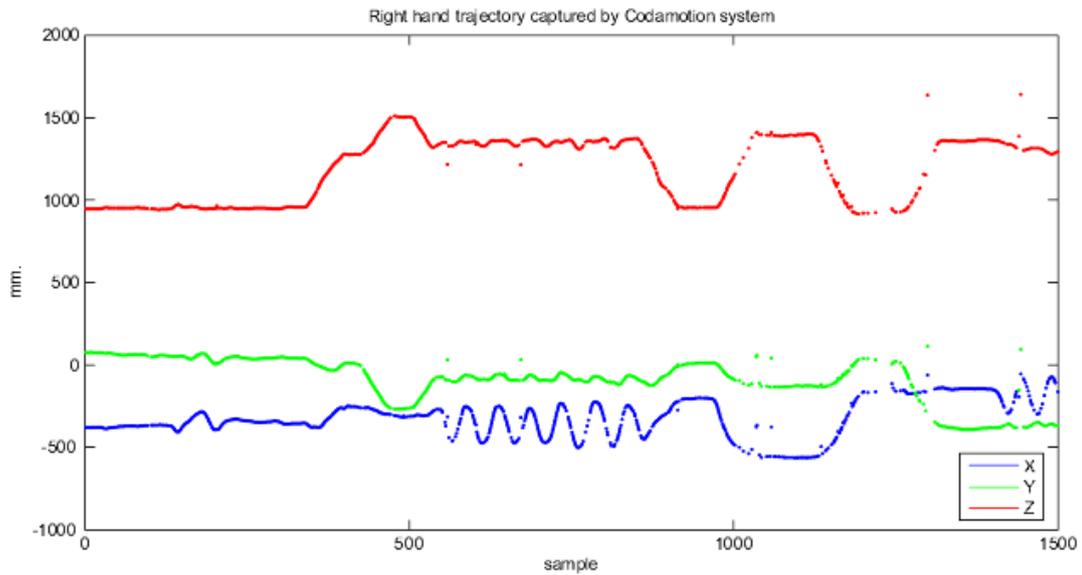


(a)

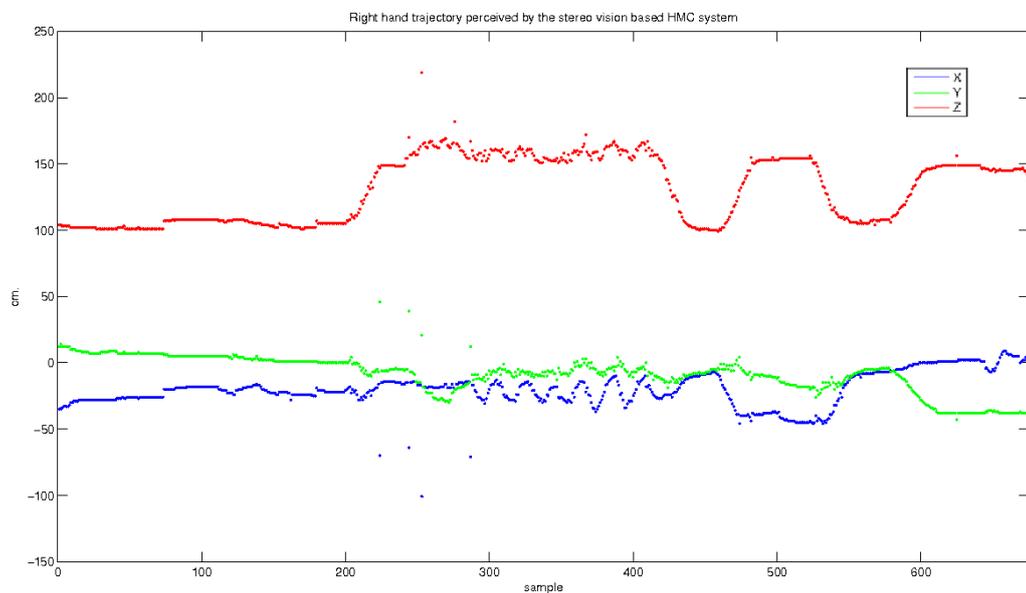


(b)

Figure 6.10: Trajectories of the right hand, captured for the same gesture using the Codamotion system and the vision-based system, respectively.



(a)



(b)

Figure 6.11: Trajectories of the right hand, captured for the same gesture using the Codamotion system and the vision-based system, respectively.

tests not only to DTW and ERP, but to all evaluated distances. Table 6.3 shows the results obtained when three executions of each of the 8 gestures are stored in the database and the remaining 29 gestures in the dataset are used to test the recognition system. As in chapter 4,  $k$ -nearest neighbor ( $k$ -NN) algorithm is used to measure the performance of evaluated methods. Two different  $k$  values (1,3), and two more restrictive measures (3/3-NN and 3/4-NN), are used. Execution time of the distance evaluation is also measured.  $\Delta conf$  is the distance from the first to the fourth nearest neighbour. Finally, for each tested method confidence values are obtained before and after (**R**) applying global reinforcement. As commented above, only two trajectories compose each of the evaluated gestures. Besides, the sequences themselves are composed by less key-points due to lower capture rates. These circumstances allow to match trajectories at higher speeds, as can be seen by comparing Tables 4.4 and 6.3.

Table 6.3: Evaluation of different distance functions. The motion has been perceived using the proposed vision-based motion capture system.

	<b>DTW</b>		<b>EDR</b>		<b>ERP</b>		<b>LCSS</b>		<b>Euc. dist.</b>	
		<b>R</b>		<b>R</b>		<b>R</b>		<b>R</b>		<b>R</b>
$\Delta conf$	0.16	0.19	0.19	0.22	0.15	0.12	0.16	0.2	0.13	0.12
secs.	0.06	0.06	0.06	0.06	0.08	0.08	0.03	0.03	$\rightarrow 0$	$\rightarrow 0$
1-NN	93%	100%	72%	76%	21%	45%	69%	86%	24%	41%
3-NN	76%	79%	48%	52%	28%	41%	48%	62%	14%	21%
3/3-NN	45%	52%	28%	31%	3%	24%	14%	28%	7%	10%
3/4-NN	57%	62%	31%	38%	10%	24%	24%	38%	10%	10%

EDR and LCSS distances behave better than before, as only two hand trajectories are now considered and these algorithms are very sensitive to the accumulative effects of small global variations (Fig. 4.7). As Table 6.3 depicts, although global reinforcement improves the discrimination ability of these algorithms, they keep on offering low recognition rates when more restrictive criterions are used.

ERP offered better results than cost-based methods and Euclidean distance in the experiments performed in chapter 4. It appeared as a solid alternative to DTW. However, the proposed HRI scenario represents a complete failure for ERP. The  $\overline{ISE}$  values in Table 6.2 show that the two trajectories captured using stereo vision are a much more inaccurate representation of perceived trajectories than Codamotion trajectories. As commented above, both DTW and ERP are sensitive to outliers and noise. But while DTW allows a more elastic matching, ERP incorporates the gap penalization when a time shift is performed. It seems possible that this penalization induces errors in ERP matching if trajectory lengths present a high deviation.

The amount of gaps introduced by ERP when comparing two sequences is equal to the difference between the lengths of these sequences. Thus, in order to determine if the gap penalization is the cause of the different performances of ERP, it is necessary to look for a relation between the results offered by ERP and the variations of sequence lengths. Two different deviations in sequence lengths are computed for each of the datasets: the deviation between different executions of the same gesture,  $\sigma_{intra}$ , and the total deviation obtained when all gestures in the dataset are considered,  $\sigma_{inter}$ . Table 6.4 shows the different obtained deviation values. As depicted, gestures captured using the Codamotion system offer a  $\sigma_{intra}$  value of 13.41%, while  $\sigma_{inter}$  grows to 33.61%. It can be seen that these gestures are highly differentiated: different executions of the same gesture tend to have more similar trajectory lengths than different executions of different gestures. On the other hand, gestures captured using the vision-based HMC system offer  $\sigma_{intra} = 28.31\%$  and  $\sigma_{inter} = 39.14\%$ . Thus, these gestures not only present a higher deviation between different executions of the same gesture, but this deviation is closer to the one concerning different executions of different gestures. Consequently, ERP will statistically have to fill more gaps when the vision-based HMC system is used. Besides, given a certain gesture, the number of gaps to fill when compared with a similar gesture will tend to be closer to the number of gaps to fill when compared with a different gesture. This reduces the discrimination ability of ERP. Thus, it can be concluded that this higher sensitivity of the ERP method with respect to DTW makes it less suitable to compare gestures in which the lengths of the trajectories tend to experience high deviations.

Table 6.4: Deviation values obtained for different gesture datasets.

Dataset	$\sigma_{intra}$	$\sigma_{inter}$
Gestures perceived using the Codamotion system	13.41%	33.61%
Gestures perceived using the proposed vision-based HMC system	28.31%	39.14%

There are few variations in the results obtained when using the two remaining distance-based methods, Euclidean distance and DTW, to match the hand trajectories captured by the stereo vision. As the rest of the evaluated algorithms, its robustness decreases in this scenario, as the more restrictive 3/3-NN and 3/4-NN measures shows. Euclidean distance keeps on being the fastest method, but it is not robust against noise, outliers and time shifting. This leads to poor recognition rates. The results provided by DTW, on the other hand, are again the best of the comparative study. DTW proves also to be able to adapt to the noisy trajectories used in this scenario, as its 1-NN recognition rates are nearly similar to those obtained when matching gestures perceived using the Codamotion system.

Finally, these results show again that the use of the proposed global similarity factor improves both discrimination (increasing  $\Delta\text{conf}$ ) and successfully recognized gestures. Besides, time consumed to compute global features is negligible, thus it is a reasonable addition to the system.

### 6.4.3 Gesture learning by imitation

The last series of experiments tests the ability of the RLbI system to learn and recognize gestures from different human demonstrators in a real interaction scenario. While the use of a virtual model of the robot is adequate to certain experiments, these last tests should use the real robotic platform in order to verify that not only the motion is correctly retargeted to robot body, but it is also correctly translated to the physical platform. As detailed above, HOAP-1 is not able to perceive social gestures due to its small baseline. The movements of HOAP-1 arms are also more constrained. Thus, we decided to use the NOMADA robot as the platform over which the final experiments are conducted.

The experiments described here make use of all the implemented parts of the NOMADA robot: its stereo vision system (the Videre cameras), and one of its arms. A detailed virtual model of the NOMADA robot, including IK and collision detection (see chapter 3) is also available. In the following experiments, the perceived human motion is retargeted to this model. Then, the motion of the right arm of the NOMADA model is transferred to the real right arm of the robot.

Fig. 6.9 shows the scenario in which these last tests are conducted. As depicted, a real indoor environment has been selected, and no special markers nor clothes are used. There are no fixed human-cameras distance. Four of the five human performers had never used the system before.

#### 6.4.3.1 Gesture segmentation

As detailed in chapter 4, the segmentation of the perceived movements into discrete gestures relies on the detection of pauses in the movement, that mark the start and the end of each gesture. These pauses must be correctly differentiated from small pauses that may occur during the execution of a certain gesture. The starting and ending points of the gesture should also be detected with enough precision as to avoid losing relevant parts of the motion. Section 4.3 presented the equations proposed in this thesis to achieve this detection, although thresholds

were not defined. For the experiments presented in this section, distance threshold  $\sigma_{mov}$  has been set to 5 cm. Time threshold  $\Delta_t$  is set to 0.25 seconds. Thus, the beginning of a new gesture is detected in only  $0.2 \cdot 0.25 = 0.05$  seconds, while the performer needs to stand still for  $5 \cdot 0.25 = 1.25$  seconds to allow the system detect the gesture ending point. These values have demonstrated to be intuitive and adequate for the human performers involved in the executed experiments. Other values in these ranges can offer similar results.

Figs. 6.12, 6.13, 6.14, 6.15 and 6.16 show the results obtained when the previously detailed method is used to detect starting and ending points over a real performance. During this test seven different gestures were executed by a human performer. It is important to consider that in this thesis only the trajectories of the head and the hands were considered to mark the starting and ending points of a gesture, although the trajectories of the shoulders, estimated by the proposed HMC system, are also depicted as some interesting observations can be extracted from them.

It can be seen that, after the initialization of the HMC, that last for some frames, the movement of the right hand triggers the starting point detector. The first gesture ends when the right hand stops moving, at around sample 175 (Fig. 6.12). The Z coordinate of the right hand movement shows that the hand is not stopped, but its movement is constrained below  $\sigma_{mov}$  for some time. This triggers the ending point detector that marks the end of the first gesture.

The second and fifth performed gestures involve wide movements of both hands. Starting and ending points are clearly visible in the figures. It is also interesting to notice how the head and shoulders (Figs. 6.14, 6.15 and 6.16) also experience some amount of movement. It is common, when some parts of the body move widely, that this motion is reflected by other body parts. Gestures three, four and seven, on the other hand, are performed with only one of the hands. However, as depicted in Fig. 6.12, the right hand moves slightly forward during execution of the third gesture, that is supposed to be performed using only the left hand. While this can be another example of involuntary movement, it can also be a consequence of disparity errors, as Fig. 6.10(b) showed before.

The sixth gesture, finally, represents a failure of the gesture starting and ending points detector. This error is motivated by the tracked body item remaining for too much time in the same position while executing the gesture. In this case, the right hand begins the movement at around sample 800, but then it stands still until sample 850 approximately. Unfortunately the gesture ending condition is met before this sample, thus a false gesture ending point is marked while gesture six is still being performed. If the duration of the segment of the sixth gesture

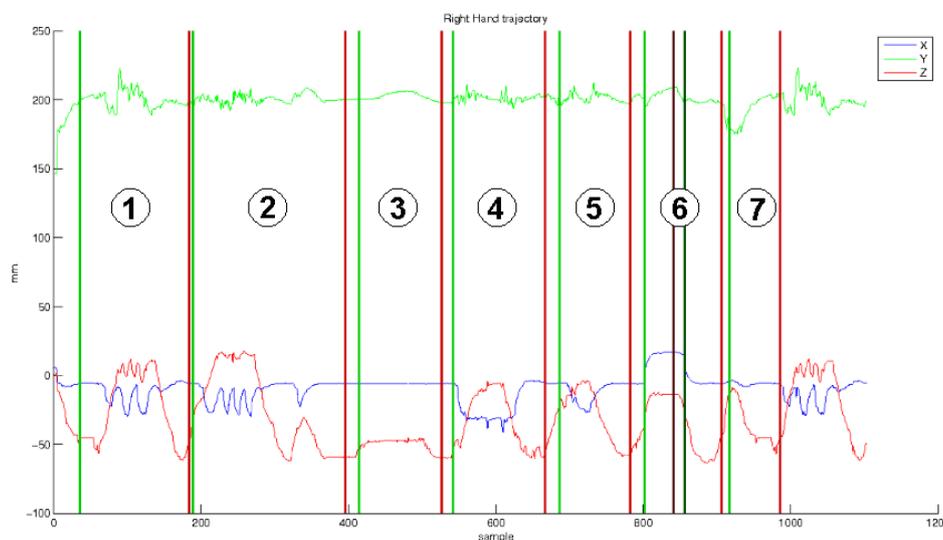


Figure 6.12: Right hand XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.

that produces the error is compared against the interval between consecutive gestures, it can be concluded that these errors are not going to be common as long as the performers are told to pause for a short time between gestures, and try not to stop for a long time while performing a certain gesture.

#### 6.4.4 Gesture learning

The knowledge update algorithm detailed in section 4.8 has been tested in RLbI scenarios in which stereo vision has been used as the only perceptual input. After some prior learning experiments, it was clear that the initial steps in the database building process were critical. Merging different gestures in the same class is a common issue if the amount of gestures in the database is low. A higher value of the  $\omega$  lower threshold avoids this error, but increases the number of wrongly unrecognized gestures in further stages of the experiment, as new executions of learned gestures are forced to be too similar to stored ones. On the other hand, the system uses few stored demonstrations of each gesture in order to allow *on-line* response. This decision can lead to an excessive sensitivity to incorrectly perceived or classified gestures.

In order to evaluate the influence of these issues, particular recognition rates of each stored gesture have been obtained for (i) the eleven Codamotion gestures depicted in Fig. 4.6; and (ii) for the eight gestures perceived using the stereo vision system mounted on the robotic head, and described in Table 6.1. These tests are also interesting to check if the system behaves

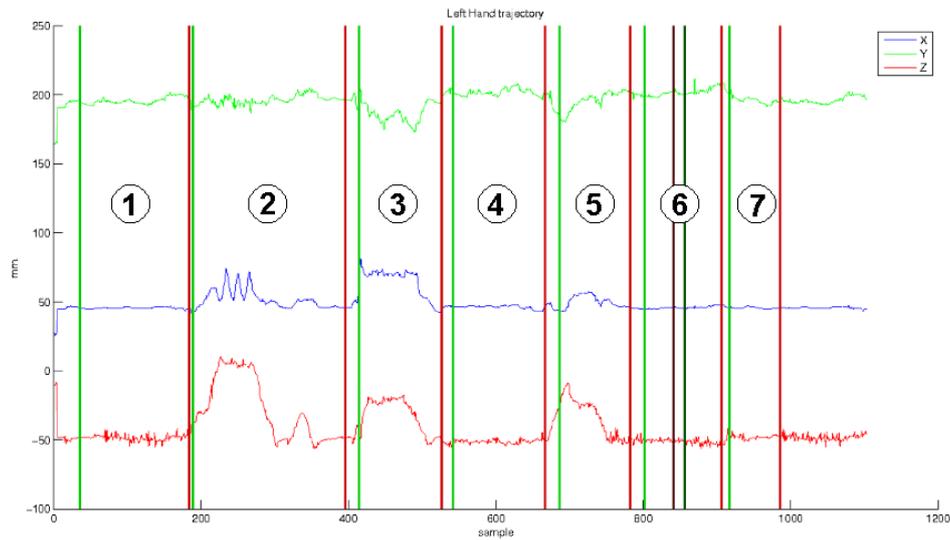


Figure 6.13: Left hand XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.

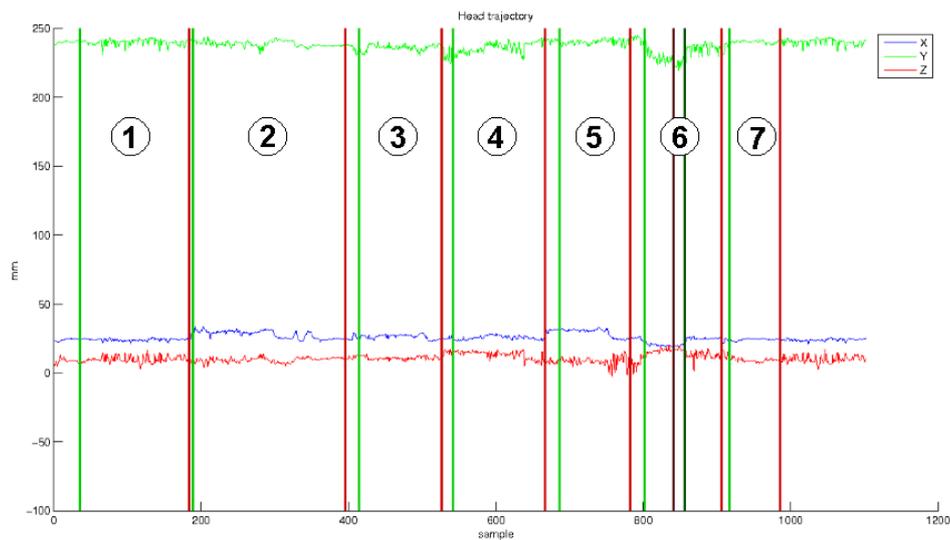


Figure 6.14: Head XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.

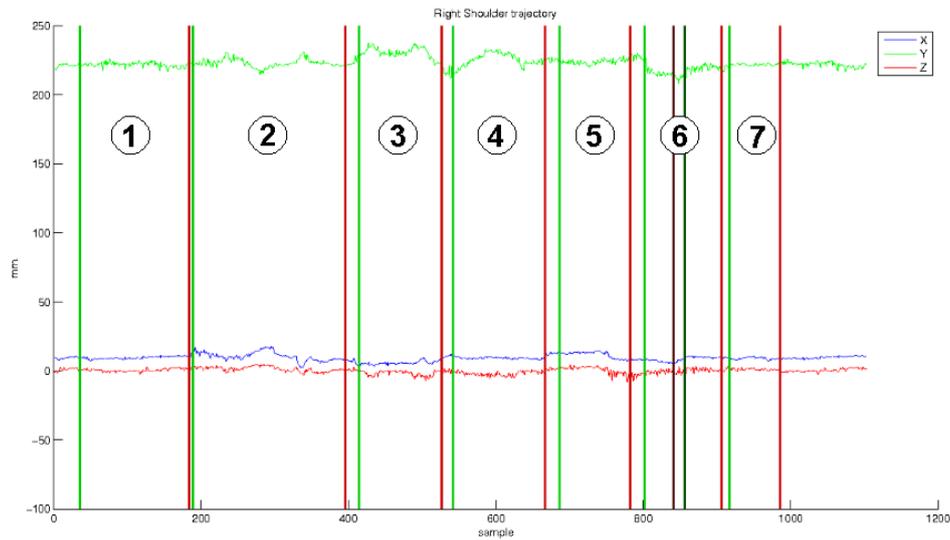


Figure 6.15: Right shoulder XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.

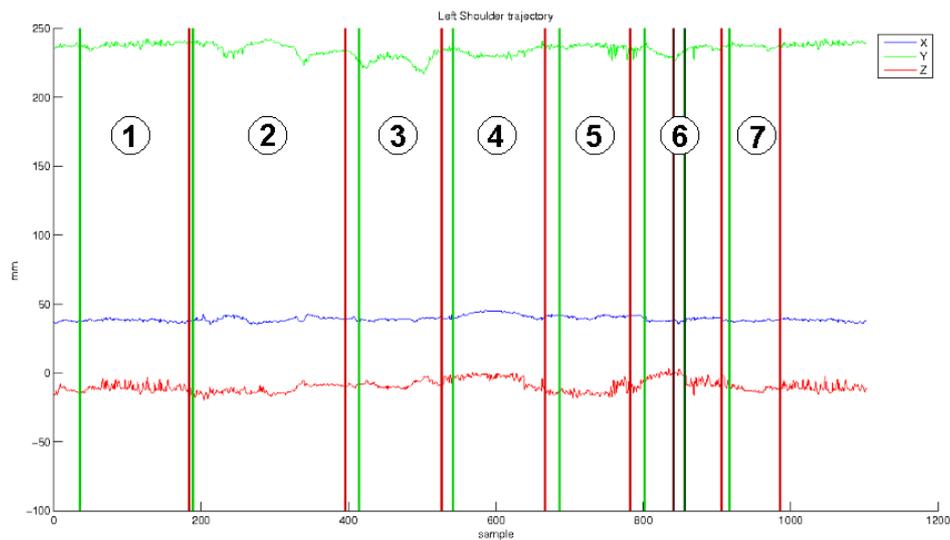


Figure 6.16: Left shoulder XYZ trajectory, over which gestures starting and ending points have been marked using green and red vertical bars, respectively.

differently for different perceptual systems. Tests involving Codamotion CX1 are useful to determine if these issues appear only when stereo vision is used or not. For these tests, DTW plus global features is used to compute matching distances, and 3-NN criterion is selected to decide whether a gesture has been recognized or not. Three executions of each type of gesture are stored at the database.

The obtained matching results for both types of gestures are detailed in the confusion matrices depicted at Tables 6.5 and 6.6. It can be seen that gesture #10 in the first database and gesture #1 in the second database are not behaving as expected. While it may be possible that these gestures were not adequate to be represented using the proposed method, these problems can be produced by a bad training as well. In order to clarify the source of these recognition errors, the stored executions of these gestures were removed from databases and three new demonstrations were stored for each gesture. Tables 6.7 and 6.8 depict the confusion matrices obtained after this correction. It can be seen that the recognition rates are improved for these two gestures. Besides, these tables show that the recognition rates are quite similar for all the gestures. Thus, if a correct set of gestures is selected to conform the database, the proposed representation performs well for different gestures regardless of their characteristics.

These tests demonstrate that a certain degree of human supervision should be used at least in the first steps of the learning process. As depicted in Fig. 4.9, the proposed algorithm includes this feature.

The last set of experiments tests the knowledge update algorithm in real indoor human-robot interaction scenarios. Apart from information about the estimated pause intervals between gestures, the system has no prior knowledge about any of the gestures that are going to be performed. The gesture database is thus empty when the first gesture is executed. The human untrained users only received a brief description of the different gestures to perform. This description is reproduced in Table 6.9.

The described gestures were performed by four different untrained performers and one additional person who had frequently used the system before. Each performer executed the gestures continuously, leaving a small pause between gestures in order to allow the system correctly segmenting the motion. It is noticeable that two of the untrained users did not need to be told to perform these pauses. The RLbI system was able to capture the motion, retarget it to the NOMADA arm, recognize gestures and update the knowledge database *on-line*. All users were satisfied with the system response, that was reported as adequate for human interaction rates. A quantitative evaluation of the temporal performance of the system is detailed in Table

### 6.10.

Fig. 6.17 shows one of the performers while executing gestures facing the stereo cameras. It can be seen that the robot right arm correctly imitates the motion of the human right arm. The NOMADA virtual model is used to check that both torso and left arms motions are also correctly imitated. As no speaker nor voice recognition system are still being used, the virtual NOMADA robot asks for supervision through textual information on a monitor. On the other hand, the human supervisor interfaces the robot using a mouse device and a keyboard. This prototype allows to test the RLbI system in real indoor environments and real interaction scenarios.

The gesture recognition rates for these experiments are around 80%, but it must be considered that the robot has no prior knowledge about performed gestures and thus it accumulates some recognition errors in the first stage of the process, until a large enough gesture database is constructed. As detailed above, human supervision is increased in these first stages of the process to avoid these errors polluting the gesture database. Considering this human supervision, all gestures performed during the experiments were correctly recognized or stored in the database as new gestures.

If the robot is not required to learn independently, i.e. without human supervision, the main issues the system has to deal with are not related to recognition or learning, but to perception. In general terms, considering both the qualitative evaluation of the human performers and the quantitative recognition results, it can be stated that the gesture is correctly recognized or learned if the motion is detected with some degree of accuracy. Thus, the main problems the system faces are that skin color detection can be noisy if direct strong light affects the scene, or if a very dark environment has to be used. On the other hand, although it is possible to recover from losing a tracked item, confusions are much more difficult to detect and avoid. The last chapter of the thesis will discuss these perceptual limitations and the options that could be addressed to solve them.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 6.17: Human performer facing the stereo cameras and executing a gesture while the NOMADA robotic arm imitates his right arm movements.

Table 6.5: Confusion matrices before correcting stored gestures. Motion perceived using a Codamotion CX1 system.

(a)

		Performed gesture										
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
<b>Result (3-NN)</b>	#1	18	0	0	0	0	0	0	0	0	0	0
	#2	0	18	0	0	0	0	0	0	0	2	0
	#3	0	0	18	0	0	0	0	0	0	0	0
	#4	0	0	0	18	0	0	0	0	0	0	0
	#5	0	0	0	0	18	0	0	0	0	0	0
	#6	0	0	0	0	0	18	0	0	0	0	0
	#7	0	0	0	0	0	0	18	0	0	0	0
	#8	0	0	0	0	0	0	0	18	0	1	0
	#9	0	0	0	0	0	0	0	0	11	0	0
	#10	0	0	0	0	0	0	0	0	0	7	0
	#11	0	0	0	0	0	0	0	0	0	0	9
	Unrecognized	0	0	0	0	0	0	0	0	0	2	0

Table 6.6: Confusion matrices before correcting stored gestures. Motion perceived using the vision-based motion capture system presented in Bandera et al. (2006).

(b)

		Performed gesture							
		#1	#2	#3	#4	#5	#6	#7	#8
<b>Result (3-NN)</b>	#1	0	0	0	0	0	0	0	0
	#2	0	2	0	0	0	0	0	0
	#3	0	0	3	0	0	0	0	0
	#4	0	0	0	4	1	0	0	0
	#5	0	0	0	0	4	0	0	0
	#6	0	0	0	0	0	4	0	0
	#7	0	0	0	0	0	0	4	0
	#7	0	0	0	0	0	0	0	4
	Unrecognized	3	0	0	0	0	0	0	0



Table 6.9: Description of the gestures given to performers testing the gesture learning by imitation system.

<b>Gesture</b>	<b>Description</b>
Left up	Point up using the left hand
Left	Point left using the left hand
Right up	Point up using the right hand
Right	Point right using the right hand
Right forward	Point forward using the right hand
Stop	Move left and right hands forward
Hello	Wave the right hand
Hands up	Move left and right hands up

Table 6.10: Quantitative temporal evaluation of the proposed RLbI system.

<b>Time parameter</b>	<b>Value</b>
Average capture rate	15 frames per second
NOMADA movement rate after interpolation	> 25 poses per second
Time paused between gestures	$\geq$ 1.25 seconds
Gesture recognition time	< 0.5 seconds

## Chapter 7

# Discussion and future work

### 7.1 Outline of the chapter

During the realization of this thesis different issues have been addressed, concerning different elements of the proposed RLbI architecture. The advantages and limitations of the given solutions to these issues have been discussed in the corresponding chapter, and experimental results have been used to evaluate them. There are, however, certain more general topics. These topics are discussed in this chapter, that also indicates some of the research lines and issues that are left open to further investigation.

### 7.2 Characteristics of the proposed RLbI architecture

Chapter 2 provided a detailed description of the proposed RLbI architecture. Other architectures, both previous and contemporary, were also described. After the proposed RLbI architecture has been tested in real scenarios, it is worth to revisit its characteristics and compare it with the other architectures presented in chapter 2.

As detailed before, the proposed system uses only stereo vision to perceive its environment. The use of a pure vision-based RLbI system meets the requirements of real learning by imitation scenarios in which speech, the other important sensory input in social learning, is not always present (e.g. in crowded or noisy environments it may be difficult to extract speech information). However, stereo vision presents strong limitations that are in fact discussed further in this chapter. While the inclusion of speech may solve some of these limitations (Breazeal et al., 2004), it may be an interesting option to consider other perceptual inputs. Proprioceptive and tactile information have been incorporated to different architectures (Demiris and Hayes, 2002;

Breazeal et al., 2004), and there is no practical reason why the social robot could not benefit from other sensory inputs, not necessarily biologically-inspired. Laser range finders, infrared sensors or omnidirectional cameras can be useful additions to the social robot.

Another important topic of the thesis is the capture of human movements. This capture is based on visual information, and there are elements of the proposed architecture that are dedicated to this task. Other architectures also have explicit elements that extract human motion from perceived visual cues (Schaal, 1999; Demiris and Hayes, 2002). The problem of implementing these elements, however, is complex as the requirements of RLbI scenarios are demanding for image processing, regarding response times, lighting variations and other environmental conditions. The HMC system proposed in this thesis meets these requirements. Quantitative evaluation shows that it provides an estimated pose close to the performed one, thus social gestures can be recognized from it. A more extensive discussion about the used vision-based perception system is presented further in section 7.3.

Knowledge representation, recognition and learning are issues that have been deeply addressed in previous works in RLbI. Most of these contributions rely on probabilistic methods, being HMMs one of the preferred approaches to solve this issue. These methods, however, require intense training phases, that may not be available in RLbI scenarios. The method proposed here follows a different approach that eliminates the necessity of a previous training phase. There are, however, several points that may be discussed about this method. The first of them comes from the definition of gesture and the implemented segmentation strategy. Gestures are supposed to contain certain semantic information. They have its own meaning. This aspect of gestures lies beyond the scope of this thesis, but it seems clear that the degree of human supervision grows when it is considered, as the person using the system has to discard many false gestures (e.g. a movement between two static gestures may not correspond to any dynamic gesture).

On the other hand, the system mechanisms to improve the gesture repertoire are limited. Human supervision is required to modify the contents of the knowledge database. There are different representations of repertoires as the one used in (Kojo et al., 2006) that solve this problem although, as commented before, they use to rely on probabilistic methods that require complex training phases. Thus, although the proposed learning component is functional and useful for this system, it could benefit from further application of different classification algorithms.

Finally, the combined retargeting strategy has proven to offer adequate results, while its location in the knowledge component makes it independent from gesture representation, recognition and learning systems. This characteristic reveals very useful in practice respect to

other architectures, where retargeting is performed in the perception component, as a last step in the motion capture process (Demiris and Hayes, 2002). In these architectures retargeting is always executed even if no physical robot motion is required. As discussed in chapter 2, it also introduces strong limitations in the gesture recognition and learning processes, as different perceived human gestures may be mapped to similar robot motion due to the differences between human and robot bodies. The proposed architecture represents an interesting alternative to these situation. The use of a human model in the perception, knowledge and learning components allows for an accurate detection of human movements. On the other hand, the execution of the retargeting element as a last step in the process increases efficiency as it is only necessary to use it when physical robot motion is required.

### 7.3 Performance and limitations of proposed vision-based perception system

As commented above, the proposed HMC system is one of the main contributions of the thesis. While it meets the requirements imposed to the system, after performed experimental results it seems to have reached its limits. Average position errors have the same magnitude than disparity and pixel errors. Thus, it may be very difficult to improve the perception of a human performer located at around 1.7 meters in RLbI scenarios, if the perception system itself is not modified.

On the other hand, in social interactions, both face expression and hand movements plays a very important role that is not considered in this thesis, due to the limitations in the used perceptual system. This problem may be solved using precise, narrow-angle cameras that could center attention in these body parts (Ude et al., 2003). Other option may be the use of different image resolutions (Arrebola et al., 1998).

The use of disparity information is a powerful mechanism in theory. In practice, disparity maps are noisy and unprecise. There are many areas in these maps in which disparity can not be computed due to lack of texture, shadows or intense lights. These problems even forced the use of a more restrictive filter for the depth coordinate, as detailed in chapter 3. Thus, if stereo disparity maps are used, it may be worthy to consider, at least, to ensure certain lighting conditions, e.g. mounting a headlamp in the social robot. On the other hand, while the use of disparity information is an interesting option as shown in this thesis, additional sensors should be investigated before implementing a social robot. For instance, the miniaturization of laser

range finders<sup>1</sup> makes them an interesting alternative to disparity computation, providing more robust and accurate measures. On the other hand, it may be difficult to perform complete explorations of the field of view using only one laser range finder, unless it is mounted in a mobile support.

## 7.4 Usage of global features

Most proposals in gesture representation and recognition focus on the local evaluation of perceived motion (Asfour et al., 2006a; Kojo et al., 2006; Calinon, 2007). However, global features are more robust against outliers and noise (Alajlan et al., 2007). They could be an interesting alternative to local features, but it is difficult to reach the discriminative levels required to achieve gesture recognition using only global features. Instead of substitute local features by global features, this thesis proposes to use global features to reinforce local comparisons. The results are promising and reveal that this combination of local and global features may significantly increase the obtained recognition rates.

The used global features are absolute and relative amplitudes. These are very simple measures, that can be easily obtained and compared and thus do not increase the system complexity. Given the obtained results, more complex global features could be considered, such as the ones used in Cooper and Bowden (2007). These features may increase the robustness and discriminative ability of the proposed system. In any case, it is important to consider how they affect to the response time and overall complexity, before including them in the system.

## 7.5 Evaluation of proposed learning system

In the proposed RLBI architecture the learning element modifies the contents of the knowledge database according to the results obtained in the comparison of perceived gesture against stored repertoire. This action is restricted to supervised inclusion of new gestures, or supervised modification of already known gestures. This approach is simpler than other proposed methods, but as commented above it is able to offer adequate functionality without increasing the complexity of the system.

The proposed learning mechanism may benefit from the use of additional sensory inputs. Speech may direct the learning process with effectiveness and accuracy. Tactile sensors and/or

---

<sup>1</sup>The Hokuyo URG-04LX-UG01 has a weight of only 160 grams, consumes 2.5 W, and provides measures with an accuracy of about 30 mm. at more than 5 meters.

kinesthetic teaching (Calinon, 2007) may also be useful to help the robot improving its knowledge database, by letting the human user directly control or correct robot movements.

## 7.6 Further work

During the description of the different parts of the system proposed in this thesis, different alternatives and topics that should be addressed in further work have been highlighted. In this section the main of these future research lines are presented. While some of them are punctual contributions to the complete system architecture, others will require deeper changes.

### 7.6.1 Increasing perceptual capabilities

As detailed above, while images provide a huge amount of information, there are important limitations in the data provided by a pair of stereo cameras. In order to provide *on-line* response, the resolution of provided images has to be constrained. Thus, although it is possible to follow upper-body gestures, it is difficult to perceive face expression or finger movements.

On the other hand, intense lights and shadows affect both colour and disparity. Plain areas in which it is difficult to detect textures are also common in RLbI scenarios, and they negatively affect the computed disparity values. Finally, calibration errors are also present. To summarize, stereo data, and specially disparity information, are usually noisy. Different methods to improve the quality of these data will be addressed, although it seems there are certain limits that will be difficult to overcome.

Thus, in order to provide the social robot with the ability to address more complex interactions its visual capabilities should be improved, respect to the data provided by a simple pair of stereo cameras. Thus, further work will consider using narrow angle cameras to detect details that are not perceived by the used wide angle cameras (Ude et al., 2003). Another option that will be considered is the inclusion of biologically-inspired foveal cameras, that can provide different resolutions depending on the requirements (Arrebola et al., 1998). Regardless the finally chosen option, the objective of this research line will be to provide more visual details about the face and hands of the person, that may be considered the most important body parts in social interactions (Breazeal et al., 2003).

Another important line of research will involve directional audio and speech recognition. The importance of speech in human society suggests to provide the social robot with a mechanism

to detect sound sources, and recognize speech. Social learning processes could then benefit from speech reinforcement. The robot could even access new learning methods, in which a certain task does not need to be demonstrated, but only described. Breazeal (2002) proposes even to provide the social robot with the ability to infer the human inner state, emotions and attitude, from its voice. This objective may also be addressed as further work.

Tactile sensors may be useful, specially for security reasons. It may be important for a social robot working in dynamic environments to detect collisions against objects or, specially, people. These collisions may be dangerous for the people in the robot vicinity. They can also damage the robot. On the other hand, NOMADA robot will incorporate a pair of grippers, or hands provided with multiple fingers, as its end-effectors. While this work has not yet been addressed, these end-effectors will most probably incorporate tactile sensors to control applied forces. These sensors could be incorporated to learning by imitation scenarios in which kinesthetic teaching is employed.

Finally, as commented above, there are certain sensors that are not necessarily inspired in human sensorial capabilities but may be very useful for the social robot (e.g. laser range finders). Future work will address their incorporation to the robot sensory input.

### 7.6.2 Considering additional global features for gesture recognition

The used global features have revealed to be an interesting addition to the system, even when these features are very simple ones. Future work will study the use of more complex global features, such as the ones proposed by Cooper and Bowden (2007). These features may provide the system with a higher degree of robustness and better recognition rates.

### 7.6.3 Using a more versatile learning module

The proposed RLbI system uses a learning component that is able to include new gestures in the stored repertoire, if human supervision is provided. This component is also able to modify stored representations of known gestures if instructed by the human user.

There are, however, other proposed learning algorithms that may automatically update stored gestures by evaluating the distances between different iterations of the same gesture, and different gestures. These systems may face important issues when maintaining the coherence of stored data (Calinon, 2007). The RLbI system proposed in this thesis avoids this problem by including the human supervisor in the process. There are, however, important advantages in the

automatization of the learning abilities of the robot that should be taken into account. A robot equipped with such a system can achieve indirect or vicarious learning, provides a more natural and intuitive agent to interact with, and can more easily learn from cognitive revision. Thus, the substitution of the proposed learning module by a more versatile and powerful mechanism will be evaluated in further work.

#### **7.6.4 Completing the construction of NOMADA and test the system in a working social robot**

HOAP-1 from Fujitsu is not an adequate platform to perceive human gestures in RLbI scenarios. NOMADA, a new social robot, is currently being developed in the ISIS research group, at University of Málaga. While a virtual realistic model of NOMADA is being used, the real robot will most probably provide new issues and objectives, specially if its perceptual abilities are improved as proposed. One of the main of the objectives defined for this robot is to achieve RLbI in real scenarios. Before addressing this objective, however, it will be necessary to consider safety, autonomy and robustness issues.

#### **7.6.5 Integrating the proposed RLbI system in a higher level architecture**

Vision-based gesture recognition is not the only capability that a social robot has to exhibit. There are many different behaviours, tasks and abilities that are also necessary for these robots. A social robot should be able to navigate through dynamic environments, avoiding collisions and achieving localization and mapping. It has to be safe for people in its environment. It should recognize already met people, and modify its behaviour according to its environment and interaction partners. As commented above, speech recognition is a key element in the social robot, that should attend to voice commands. It would be also desirable for the robot to generate speech responses that can provide valuable feedback for the people interacting with it, or to improve its interaction abilities in general.

A social robot should be able to decide *when*, *what* and *how* to imitate (Schaal, 1999). While the contents of this thesis partially answer the two last questions, the first one should be addressed at higher decision layers. These layers may also affect the latter questions. For instance, it may be important to decide whether certain gestures should be learnt (and imitated) or not, depending on different factors such as the performer identity, the environment or the inner state of the social robot.

The implementation of these higher level cognitive layers is currently being addressed in

the research group where this thesis has been elaborated. Other behaviours such as localization and mapping, or face recognition, are also being currently implemented. Future work will integrate the proposed vision-based RLBI system with them in a complete architecture, controlled by high level decision mechanisms. A prior implementation of this architecture has been already contributed ([Vázquez-Martín et al., 2006](#)). The complete architecture will be integrated and tested in the NOMADA robot when it is finished.

## Chapter 8

# Conclusion

This thesis proposes a novel RLbI system which main objective is to fit the demanding requirements of real human-robot interaction processes in real environments, a situation named in this thesis 'RLbI scenario'. More precisely, the presented system focus on the perception, representation and recognition of the upper-body gestures that a person performs while interacting with the robot. In order to implement this functionality, it has been necessary to achieve a set of steps that have been deeply explained in this thesis, and that can be summarize as follows: (a) perceive the human counterpart; (b) capture his/her motion; (c) divide motion into different gestures; (d) encode these gestures in an useful and efficient format; (e) recognize the gesture if it is already stored in the knowledge database of the robot; (f) learn, or modify the knowledge database according to the results of the recognition step; and (g) if required, perform physical imitation of perceived or stored gestures.

Chapter 1 describes the framework and motivation of the thesis, defines the concept of social robot, and provides a complete description of the requirements and characteristics of RLbI scenarios. The solution adopted in this thesis is also presented, along with its main contributions.

In Chapter 2, the proposed RLbI architecture is detailed. This architecture is inspired in concepts mainly taken from social learning and previous work in the field of RLbI. It has, however, certain particularities, and it proposes a series of modifications and novelties respect to previous works. The main of these contributions is the implementation of the knowledge database in the motion space of the human. Thus, a human model replaces the robot in the learning by imitation process. This can be described as 'use a human model to represent human perceived motion' and represents an important difference respect to most previous approaches. It allows not only to increase the discriminative ability of the gesture recognition system, but

also to increase efficiency as the retargeting module, that translates the motion from the human model to the robot, has only to be executed if physical imitation needs to be performed.

The requirements of RLbI scenarios concerning natural interaction and robot autonomy move the majority of social robots to rely on vision as their main sensory input. Chapter 3 details the vision-based HMC system proposed in this thesis. This system extracts human motion from images perceived by a pair of stereo cameras mounted on the head of a social robot. It incorporates a novel torso pose estimation algorithm. It also poses arms using an IK algorithm modified by a novel alternative pose evaluation system. An extensive quantitative evaluation of this complete HMC system is provided in this chapter. This evaluation remarks the advantages and drawbacks of the proposed system, and reveals its usefulness as a part of the proposed RLbI architecture.

Chapter 4 presents the gesture representation, recognition and learning systems that have been implemented in this thesis. After a description of the segmentation algorithm used to extract discrete gestures from perceived motion, the representation of these gestures is the first addressed issue. This thesis propose to encode gestures as two sets of features. The first set includes local features, that are basically dominant points extracted from adaptive 3D curvature functions, obtained from perceived trajectories of relevant body parts. This representation is slightly less efficient than PCA, but as it is demonstrated further in this chapter, it provides better recognition results. It also presents important advantages over other descriptors as CSS or fixed curvature functions. The second set of features that encode the gesture are simple global measures related to absolute and relative amplitudes of performed motion.

Once gestures have been encoded, it is necessary, in order to recognize them, to compare perceived gesture against the ones that compose the repertoire of the robot. The necessity of achieving recognition without extensive training phases was the main reason that prevented against the use of HMM-based solutions. Instead, different dynamic programming alignment techniques have been evaluated to perform comparison between local features. Experiments performed in controlled environments point towards the use of the DTW algorithm. Recognition, however, does not only employs local features, but also global ones. In this thesis an analytic method to compare global features is proposed. The final matching result, expressed in this thesis as a certain confidence value, is a combination of the local distance and the global similarity. The confidence values obtained when a perceived gesture is compared against the gestures in the repertoire of the robot are the inputs for the proposed learning system. Gestures that are not recognized may represent new gestures that should be incorporated to the repertoire. However, they may also represent incorrectly performed gestures, or noisy movements. The

learning system asks for a certain degree of human supervision in these confuse situations, that are detected using a novel, multiple threshold-based method.

The retargeting system, used to translate the perceived human movements to the motion space of the robot, represents the last step in the RLbI process, and it is presented in chapter 5. This retargeting module takes concepts from computer graphics animation to provide a flexible and efficient combined retargeting strategy, that is evaluated in this chapter.

Chapter 6 presents the results obtained when the complete RLbI architecture is used in real RLbI scenarios. Finally, chapter 7 discusses the results offered by the proposed RLbI architecture, and other issues and key topics that have appeared in the context of the thesis.

The main contribution of this thesis is the complete implementation of the RLbI system. This system has been validated through experiments executed in real RLbI scenarios. It includes different modules, from perception of human motion to learning processes, in which different more specific contributions have been proposed. The system architecture itself presents several contributions regarding its structure and components. The experimental results show that the main limitation of the proposal comes from the perception system, which is nevertheless able to track human movements in most situations, while imposing only soft initialization constraints. The gesture recognition and learning systems are also able to offer adequate results at human interaction rates, requiring only few, if any, training samples. The human is also allowed to participate in the learning procedure to solve uncertain situations. While further work has still to be conducted, the presented RLbI system, in conclusion, conforms a module that can be effectively integrated into a social robot, and combined with additional elements to achieve higher, more complex behaviours.



# Bibliography

- Agarwal, A. and Triggs, B. (2006). Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58.
- Aggarwal, J. K. and Cai, Q. (1999). Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440.
- Alajlan, N., Rube, I. E., Kamel, M., and Freeman, G. (2007). Shape retrieval using triangle-area representation and dynamic space warping. *Pattern Recognition*, 40:1911–1920.
- Aleotti, J. and Caselli, S. (2006). Robust trajectory learning and approximation for robot programming by demonstration. *Robotics and Autonomous Systems*, 54:409–413.
- Alissandrakis, A., Nehaniv, C., and Dautenhahn, K. (2007). Correspondence mapping induced state and action metrics for robotic imitation. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Special issue on robot learning by observation, demonstration and imitation*, 37(2):299–307.
- Ardizzone, E., Chella, A., and Pirrone, R. (2000). Pose classification using support vector machines. In *Proc. of the IEEE/INNS/ENNS Int. Joint Conf. on Neural Networks (IJCNN)*, pages 317–322.
- Arrebola, F., Urdiales, C., Camacho, P., and Sandoval, F. (1998). Vision system based on shifted fovea multiresolution retinotopologies. In *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (IECON '98)*, volume 3, pages 1357–1361.
- Asfour, T., Gyarfas, F., Azad, P., and Dillmann, R. (2006a). Imitation learning of dual-arm manipulation tasks in humanoid robots. In *Proceedings of the 6th IEEE RAS International Conference on Humanoid Robots*, pages 40–47.
- Asfour, T., Regenstein, K., Azad, P., Schröde, J., and Dillmann, R. (2006b). Armar-iii: A humanoid platform for perception-action integration. In *2nd International Workshop on Human-Centered Robotic Systems (HCRS'06)*.

- Asimov, I. (1942). *Astounding Science Fiction*, chapter Runaround. Street & Smith.
- Azad, P., Ude, A., Asfour, T., and Dillmann, R. (2007a). Stereo-based markerless human motion capture for humanoid robot systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2007)*, pages 3951–3956.
- Azad, P., Ude, A., Asfour, T., and Dillmann, R. (2007b). Stereo-based markerless human motion capture for humanoid robot systems. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pages 3951–3956, Roma, Italy.
- Aziz, M. and Mertsching, B. (2007). *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint*, chapter Color saliency and inhibition using static and dynamic scenes in region based visual attention, pages 234–250. Springer, Heidelberg.
- Bandera, A., Urdiales, C., Arrebola, F., and Sandoval, F. (2000). Corner detection by means of adaptively estimated curvature function. *Electronic Letters*, 36(2):124–126.
- Bandera, J. P., Marfil, R., Molina-Tanco, L., Bandera, A., Rodríguez, J. A., and Sandoval, F. (2008a). Visual tracking of human activity for a social robot working on real indoor scenarios. *International Journal of Factory Automation, Robotics and Soft Computing*, 3:120–128.
- Bandera, J. P., Marfil, R., Molina-Tanco, L., Rodríguez, J. A., Bandera, A., and Sandoval, F. (2006). Robot learning of upper-body human motion by active imitation. In *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, pages 314–320.
- Bandera, J. P., Marfil, R., Molina-Tanco, L., Rodríguez, J. A., Bandera, A., and Sandoval, F. (2007). *Robot Learning by Active Imitation*, pages 519–544. I-Tech Education and Publishing, Vienna, Austria.
- Bandera, J. P., Marfil, R., Rodríguez, J. A., Molina-Tanco, L., and Sandoval, F. (2008b). A novel hybrid approach to upper-body human motion capture. In *Proceedings of the 14th Mediterranean Electrotechnical Conference (MELECON 2008)*, pages 368–373, Ajaccio, France.
- Bandura, A. (1969). *Handbook of socialization theory and research*, chapter Social learning theory of identificatory processes, pages 213–262. Rand-McNally, Chicago, IL, USA.
- Barreto, J., Menezes, P., and Dias, J. (2004). Human-robot interaction based on haar-like features and eigenfaces. In *IEEE International Conference on Robotics and Automation (ICRA 2004)*, volume 2, pages 1888–1893.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA.

- Billard, A. and Dillmann, R. (2006). Special issue on the social mechanisms of robot programming by demonstration. *Robotics and Autonomous Systems*, 54(5):351–352.
- Boardman, T. (2005). *3ds max 7 Fundamentals*. New Riders Publishing, Thousand Oaks, CA, USA.
- Brand, M. and Hertzmann, A. (2000). Style machines. In *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 183–192.
- Breazeal, C. (2002). *Designing sociable robots*. Cambridge, MA, USA.
- Breazeal, C. (2003). Toward sociable robots. *Robotics and Autonomous Systems*, 42(3-4):167 – 175.
- Breazeal, C., Brooks, A., Gray, J., Hancher, M., McBean, J., Stiehl, D., and Strickon, J. (2003). Interactive robot theatre. *Communications of the ACM*, 46(7):76–84.
- Breazeal, C., Brooks, A., Gray, J., Hoffman, G., Kidd, C., Lee, H., Lieberman, J., Lockerd, A., and Mulanda, D. (2004). Humanoid robots as cooperative partners for people. *International Journal of Humanoid Robots*, 1(2):1–34.
- Breazeal, C., Buchsbaum, D., Gray, J., Gatenby, D., and Blumberg, B. (2005). Learning from and about others: Towards using imitation to bootstrap the social understanding of others. *Artificial Life*, 11(1-2):31–62.
- Breazeal, C., Edsinger, A., Fitzpatrick, P., and Scassellati, B. (2001). Active vision for sociable robots. *IEEE Trans. on Man, Cybernetics, and Systems, Part A: Systems and Humans*, 31(5):443–453.
- Calinon, S. (2007). *Continuous extraction of task constraints in a robot programming by demonstration framework*. Ph.d. dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, EPFL.
- Calinon, S. and Billard, A. (2005). Stochastic gesture production and recognition model for a humanoid robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pages 2769–2774.
- Canny, J. (1983). Finding edges and lines in images. Technical report, Cambridge, MA, USA.
- Carmona, A., Molina-Tanco, L., Azuaga, M., Rodríguez, J., and Sandoval, F. (2007). Online absorption of mediolateral balance disturbances for a small humanoid robot using accelerometer

- and force-sensor feedback. In *Proceedings of the 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1–6, Zurich, Switzerland.
- Chen, L., Tamer, M., and Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the Special Interest Group on Management of Data (SIGMOD 2005)*, pages 491–502.
- Cheng, G., Sang-Ho, H., Morimoto, J., Ude, A., Colvin, G., Scroggin, W., and Jacobsen, S. (2006). Cb: A humanoid research platform for exploring neuroscience. In *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, pages 182–187.
- Choi, K. and Ko, H. (2000). On-line motion retargetting. *J. Visual. Comput. Animation*, 11:223–243.
- Contini, R. (1972). Body segment parameters, part ii. *Artificial Limbs*, 16(1):1–19.
- Cooper, H. M. and Bowden, R. (2007). Sign language recognition using boosted volumetric features. In *Proceedings of the IAPR Conf. on Machine Vision Applications*, pages 359–362, Tokyo.
- Craig, J. (1986). *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, Boston, MA, USA.
- Craw, I., Tock, D., and Bennett, A. (1992). Finding face features. In *proceedings of the European Conference on Computer Vision*, pages 92–96.
- Croitoru, A., Agouris, P., and Stefanidis, A. (2005). 3d trajectory matching by pose normalization. In *Proceedings of the 13th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS'05)*, pages 153–162.
- Cypher, A. (1993). *Watch what I do: Programming by demonstration*. The MIT Press, Cambridge, MA, USA.
- Dai, Y. and Nakano, Y. (1996). Face-texture model based on sglcd and its application in face detection in a color scene. *Pattern Recognition*, 29(6):1007–1017.
- Dautenhahn, K. and Billard, A. (1999). Bringing up robots or -the psychology of socially intelligent robots: From theory to implementation. In *Proceedings of the third annual Conf. on Autonomous Agents*, pages 366–367, Seattle, Washington, United States.
- Dautenhahn, K. and Nehaniv, C. (2002). *Imitation in Animals and Artifacts*. MIT Press, Cambridge, MA, USA.

- Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press.
- Demirdjian, D., Ko, T., and Darrell, T. (2005). Untethered gesture acquisition and recognition for virtual world manipulation. *Virtual Reality*, 8:222–230.
- Demiris, J. and Hayes, G. (2002). *Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model*. MIT Press: Cambridge.
- Demiris, Y. and Khadhour, B. (2005). Hierarchical, attentive multiple models for execution and recognition (hammer). In *Proceedings of the International Conference on Robotics and Automation, Workshop on Robot Programming by Demonstration*, pages 38–41, Barcelona, Spain.
- Dimitrov, D., Wieber, P. B., Stasse, O., Ferreau, H. J., and Diedam, H. (2009). An optimized linear model predictive control solver for online walking motion generation. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 1171–1176, Kōbe, Japan.
- Donald, M. (1991). *Origins of the Modern Mind*. Harvard University Press, Cambridge, MA, USA.
- Downie, M. (2000). *Behavior, animation, and music: The music and movement of synthetic characters*. Master’s thesis, MIT Program in Media Arts and Sciences, Cambridge, MA, USA.
- Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. Wiley-Interscience, New York, NY, USA.
- Dufay, B. and Latombe, J. C. (1984). An approach to automatic robot programming based on inductive learning. *The International Journal of Robotics Research*, 3(4):3–20.
- Earthy, J., Jones, B. S., and Bevan, N. (2001). The improvement of human-centred processes—facing the challenge and reaping the benefit of iso 13407. *International Journal of Human-Computer Studies*, 55(4):553–585.
- Eriksen, C. and Yen, Y. (1985). Allocation of attention in the visual field. *Journal of Experimental Psychology: Human Perception and Performance*, 11(5):583–597.
- Faria, D., Aliakbarpour, H., and Dias, J. (2009). Grasping movements recognition in 3d space using a bayesian approach. In *Proceedings of the 14th International Conference on Advanced Robotics (ICAR 2009)*, Munich, Germany.

- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395.
- Folgheraiter, M., Bongardt, B., Schmidt, S., de Gea, J., Albiez, J., and Kirchner, F. (2009). Design of an arm exoskeleton using a hybrid model- and motion-capture-based technique. In *Workshop on Interfacing the Human and the Robot (IHR). Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, Kōbe, Japan.
- Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42:143–166.
- Fontmartry, M., Lerasle, F., and Dan’es, P. (2007). Data fusion within a modified annealed particle filter dedicated to human motion capture. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, pages 3391–3396.
- Forlizzi, J. and DiSalvo, C. (2006). Service robots in the domestic environment: a study of the roomba vacuum in the home. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction (HRI ’06)*, pages 258–265, New York, NY, USA. ACM.
- Freund, Y. and Schapire, R. (1995). A decision-theoretic generalization of on-line learning an application to boosting. In *Proceedings of the European Conference on Computational Learning Theory*, pages 119–139.
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA.
- Galef, B. G. J. (1988). *Social Learning: Psychological and Biological Perspectives*, chapter Imitation in animals: History, definition, and interpretation of data from the psychological laboratory, pages 3–28. Erlbaum, Hillsdale, N.J., USA.
- Gallese, V. and Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501.
- Gavrila, D. M. (1999). The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, 73(1):82–98.
- Ghahramani, Z. and Beal, M. (2000). *Variational inference for bayesian mixtures of factor analysers*, volume 12, pages 449–455. MIT Press.

- Gleicher, M. (1998). Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98)*, pages 33–42, New York, NY, USA. ACM.
- Gortler, S. J. and Cohen, M. F. (1995). Hierarchical and variational geometric modeling with wavelets. In *Proceedings of the Symposium on Interactive 3D Graphics*, pages 35–42.
- Gottschalk, S., Lin, M., and Manocha, D. (1996). Obb-tree: A hierarchical structure for rapid interference detection. *Computer Science*, 30:171–180.
- Hecht, F., Azad, P., and Dillmann, R. (2009). Markerless human motion tracking with a flexible model and appearance learning. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 3173–3179, Kōbe, Japan.
- Heinzmann, J. and Zelinsky, A. (1999). A safe-control paradigm for human robot interaction. *Journal of Intelligent and Robotic Systems*, 25(4):295–310.
- Heise, R. (1989). *Demonstration instead of programming: Focussing attention in robot task acquisition*. Ph.d. dissertation, University of Calgary, Alberta, Canada.
- Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., and Tanie, K. (2001). Planning walking patterns for a biped robot. *IEEE Transactions on Robotics and Automation*, 17:280–289.
- Huertas, E. (1992). *El aprendizaje no-verbal de los humanos*. Ediciones Pirámide, Madrid, 1st edition.
- Inamura, T., Toshima, I., Tanie, H., and Nakamura, Y. (2004). Embodied symbol emergence based on mimesis theory. *The International Journal of Robotics Research*, 23(4):363–377.
- Inoue, H., Tachi, S., Nakamura, Y., Hirai, K., Ohyu, N., Hirai, S., Tanie, K., Yokoi, K., and Hirukawa, H. (2001). Overview of humanoid robotics project of meti. In *Proceedings of the 32nd International Symposium on Robotics*, pages 1478–1482.
- Ito, M., Noda, K., Hoshino, Y., and Tani, J. (2006). Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, 19(3):323–337.
- Itti, L. (2002). Real-time high-performance attention focusing in outdoors color video streams. In Rogowitz, B. E. and Pappas, T. N., editors, *Proceedings of the SPIE Human Vision and Electronic Imaging (HVEI'02)*, pages 235–243.

- Jolliffe, I. (1986). *Principal Component Analysis*. Springer-Verlag, Heidelberg.
- Kaiser, M. and Dillmann, R. (1996). Building elementary robot skills from human demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1996)*, volume 16, pages 307–354, Minnesota, USA.
- Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., and Isozumi, T. (2004). Humanoid robot hrp-2. In *Proceedings of the IEEE International Conference of Robotics and Automation (ICRA 2004)*, pages 1083–1090, New Orleans, LA, USA.
- Kato, I. (1973). Development of wabot 1. *Biomechanism*, 2:173–214.
- Kehl, R. and Gool, L. V. (2006). Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, 104(2–3):190–209.
- Kerpa, O., Weiss, K., and Wörn, H. (2003). Development of a flexible tactile sensor for a humanoid robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6, Las Vegas, Nevada, USA.
- Klopčar, N., Tomšič, M., and Lenarčič, J. (2007). A kinematic model of the shoulder complex to evaluate the arm-reachable workspace. *Journal of Biomechanics*, 40:86–91.
- Koch, C. and Ullman, S. (1985). Shifts selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227.
- Kojo, N., Inamura, T., Okada, K., and Inaba, M. (2006). Gesture recognition for humanoids using proto-symbol space. In *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, pages 76–81.
- Krüger, V., Anderson, J., and Prehn, T. (2005). Probabilistic model-based background subtraction. In *Scandinavian Conference on Image Analysis*, pages 19–22, Joensuu, Finland.
- Kulic, D., Lee, D., and Nakamura, Y. (2009). Whole body motion primitive segmentation from monocular video. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 3166–3172, Kōbe, Japan.
- Kuniyoshi, Y., Yorozu, Y., Inaba, M., and Inoue, H. (2003). From visuo-motor self learning to early imitation - a neural architecture for humanoid learning. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA 2003)*, pages 3132–3139, Taipei, Taiwan.

- Lanitis, A., Taylor, C., and Cootes, T. (1995). An automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):393–401.
- Lee, C. and Xu, Y. (1996). Online, interactive learning of gestures for human/robot interfaces. In *IEEE International Conference on Robotics and Automation (ICRA 1996)*, volume 4, pages 2982–2987, Minnesota, USA.
- Lee, M. W. and Cohen, I. (2006). A model-based approach for estimating human 3d poses in static images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):905–916.
- Lopes, M. and Santos-Victor, J. (2005). Visual learning by imitation with motor representations. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 35(3):438–449.
- MacNeilage, P. F. (1998). The frame/content theory of evolution of speech production. *Behavioral and Brain Sciences*, 21:499–511.
- Maestri, G. (1996). *Digital Character Animation*. New Riders Publishing.
- Maini, E., Teti, G., Rubino, M., Laschi, C., and Dario, P. (2002). Bio-inspired control of eye-head coordination in a robotic anthropomorphic head. *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 549–554.
- Maki, A., Nordlund, P., and Eklundh, J. (2000). Attentional scene segmentation: integrating depth and motion. *Computer Vision and Image Understanding*, 78(3):351–373.
- Marfil, R. (2006). *Tracking Objects with the Bounded Irregular Pyramid*. Ph.d. dissertation, Dpto. Tecnología Electrónica, Universidad de Málaga, Spain, ISBN 84-689-9607-6, University of Málaga, Spain.
- Marfil, R., Bandera, A., Rodríguez, J., and Sandoval, F. (2004). Real-time template-based tracking of non-rigid objects using bounded irregular pyramids. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, volume 1, pages 301–306, Sendai, Japan.
- Marfil, R., Molina-Tanco, L., Bandera, A., Rodríguez, J., and Sandoval, F. (2006). Pyramid segmentation algorithms revisited. *Pattern Recognition*, 39(8):1430–1451.
- Marfil, R., Molina-Tanco, L., Bandera, A., and Sandoval, F. (2007). *6th IARP -TC-15 Workshop on Graphbased Representations in Pattern Recognition, LNCS 4538*, chapter The construction of bounded irregular pyramids using a union-find decimation process, pages 307–318. Springer, Heidelberg.

- Marji, M. and Siy, P. (2004). Polygonal representation of digital planar curves through dominant point detection - a nonparametric algorithm. *Pattern Recognition*, 34:2113–2130.
- Martz, P. (2007). *OpenSceneGraph Quick Start Guide*. Computer Graphics Systems Development Corporation, Mountain View, California.
- Maxwell, B. A., Meeden, L. A., Addo, N., Brown, L., Dickson, P., Ng, J., Olshfski, S., Silk, E., and Wales, J. (1999). Alfred: The Robot Waiter Who Remembers You. *AI Magazine*.
- Mayer, H., Nagy, I., Knoll, A., Braun, E., Large, R., and Bauernschmitt, R. (2007). Adaptive control for human-robot skilltransfer: Trajectory planning based on fluid dynamics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2007)*, pages 1800–1807.
- Mazur, J. E. (1986). *Learning and Behavior*. Prentice-Hall, Englewood Cliffs, Nueva Jersey.
- Meltzoff, A. and Moore, M. (1989). Imitation in newborn infants: Exploring the range of gestures imitated and the underlying mechanisms. *Developmental Psychology*, 25:954–962.
- Metta, G., Panerai, F., Manzotti, R., and Sandini, G. (2000). Babybot: an artificial developing robotic agent. In *Proceedings of the Sixth International Conference on the Simulation of Adaptive Behaviors (SAB 2000)*, pages 42–53, month = September,.
- Miller, N. E. and Dollard, J. (1941). *Social learning and imitation*. Yale University Press, New Haven, CT, USA.
- Mitchell, T., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. (1994). Experience with a learning personal assistant. *Communications of the ACM*, 37(7):80–91.
- Mitchelson, J. (2003). *Multiple-Camera Studio Methods for Automated Measurement of Human Motion*. Ph.d. dissertation, Centre for Vision, Speech and Signal Processing (CVSSP). School of Electronics and Physical Sciences, University of Surrey, UK.
- Moeslund, T., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126.
- Moeslund, T. B. and Granum, E. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268.
- Mohammad, Y. and Nishida, T. (2009). Interactive perception for amplification of intended behavior in complex noisy environments. *AI & Society*, 23(2):167–186.

- Mohan, R. E., Calderon, C. A. A., Zhou, C., and Yue, P. K. (2008). Evaluating virtual emotional expression systems for human robot interaction in rehabilitation domain. In *Proceedings of the 2008 International Conference on Cyberworlds (CW '08)*, pages 554–560, Washington, DC, USA. IEEE Computer Society.
- Mokhtarian, F. and Mackworth, A. (1986). Scale-based description and recognition of planar curves and two dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):34–43.
- Molina-Tanco, L., Bandera, J. P., Marfil, R., and Sandoval, F. (2005). Real-time human motion analysis for human-robot interaction. In *Proceedings of the IEEE/RSJ International Conference on Robotics and Intelligent Systems*, pages 1808–1813, Alberta, Canada.
- Mori, G. and Malik, J. (2002). Estimating human body configurations using shape context matching. *Lecture Notes in Computer Science*, 2352:666–674.
- Mosterín, J. (2005). *La naturaleza humana*. Espasa Calpe, Madrid.
- Muench, S., Kreuziger, J., Kaiser, M., and Dillmann, R. (1994). Robot programming by demonstration (rpd) - using machine learning and user interaction methods for the development of easy and comfortable robot programming systems. In *Proceedings of the International Symposium on Industrial Robots (ISIR)*, pages 685–693.
- Mülihg, M., Gienger, M., Hellbach, S., Steil, J. J., and Goerick, C. (2009). Task-level imitation learning using variance-based movement optimization. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 1177–1184, Kōbe, Japan.
- Nam, Y. and Wohn, K. (1996). Recognition of space-time hand-gestures using hidden markov model. In *ACM Symposium on Virtual Reality Software and Technology*, pages 51–58.
- Nehaniv, C. L. and Dautenhahn, K. (2002). The correspondence problem. pages 41–61.
- Nicolescu, M. and Matarić, M. (2003). Natural methods for robot task learning: Instructive demonstrations, generalizations and practice. In *Proceedings of the international joint conference on autonomous agents and multiagent systems (AAMAS)*, pages 241–248.
- Niyogi, S. A. and Adelson, E. H. (1994). Analyzing and recognizing walking figures in xyt. In *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1994)*, pages 469–474.

- Ogawara, K., Takamatsu, J., Kimura, H., and Ikeuchi, K. (2002). Modeling manipulation interactions by hidden markov models. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1096–1101, Lausanne, Switzerland.
- Ohya, A., Yuta, S., Yoshida, T., Koyanagi, E., Imai, T., Kitamura, S., Takeuchi, A., and Minamikawa, T. (2009). An optimized linear model predictive control solver for online walking motion generation. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 1429–1434, Kōbe, Japan.
- Orabona, F., Metta, G., and Sandini, G. (2007). *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint*, chapter A proto-object based visual attention model, pages 198–215. Springer, Heidelberg.
- Ozawa, S., Pang, S., and Kasabov, N. (2008). Incremental learning of chunk data for online pattern classification systems. *IEEE Transactions on Neural Networks*, 19(6):1061–1074.
- Pang, S., Ozawa, S., and Kasabov, N. (2005). Incremental linear discriminant analysis for classification of data streams. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 35(5):905–914.
- Park, H., Kim, E., Jang, S., Park, S., Park, M., and Kim, H. (2005). Hmm-based gesture recognition for robot control. *Pattern Recognition and Image Analysis*, pages 607–614.
- Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). Learning and generalization of motor skills by learning from demonstration. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 763–768, Kōbe, Japan.
- Peasant, S. (1986). *Bodyspace : anthropometry, ergonomics, and design*. Taylor & Francis, London, Philadelphia.
- Peirce, B. (1852). Criterion for the rejection of doubtful observations. *Astronomical Journal*, 2(21):161–163.
- Peshkin, M., Colgate, J., and Moore, C. (1996). Passive robots and haptic displays based on nonholonomic elements. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1996)*, volume 1, pages 551–556, Minnesota, USA.
- Rajko, S., Qian, G., Ingalls, T., and James, J. (2007). Real-time gesture recognition with minimal training requirements and on-line learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, pages 1–8.

- Reche, P., Urdiales, C., Bandera, A., Trazegnies, C., and Sandoval, F. (2002). Corner detection by means of contour local vectors. *Electronic Letters*, 38(14):699–701.
- Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141.
- Rodriguez, W., Last, M., Kandel, A., and Bunke, H. (2004). 3-dimensional curve similarity using string matching. *Robotics and Autonomous Systems*, 49:165–172.
- Rosenfeld, A. and Johnston, E. (1973). Angle detection on digital curves. *IEEE Transactions on Computers*, 22:875–878.
- Ryokai, K., Lee, M. J., and Breitbart, J. M. (2009). Multimodal programming environment for kids: a "thought bubble" interface for the pleo robotic character. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4483–4488, New York, NY, USA. ACM.
- Safonova, A., Pollard, N., and Hodgins, J. (2003). Optimizing human motion for the control of a humanoid robot. In *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines (AMAM2003)*, Kyoto, Japan.
- Safonova, A., Pollard, N. S., and Hodgins, J. K. (2002). Adapting human motion for the control of a humanoid robot. In *Proceedings of International Conference on Robotics and Automation*, pages 1390–1397.
- Sarkar, D. (1993). A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. *Pattern Recognition Letters*, 14(12):959–964.
- Saunders, J., Nehaniv, C., and Dautenhahn, K. (2006). Teaching robots by moulding behavior and scaffolding the environment. In *Proceedings of the ACM SIGCHI/SIGART conference on Human-Robot Interaction (HRI)*, pages 118–125.
- Sausser, E. and Billard, A. (2005). View sensitive cells as a neural basis for the representation of others in a self-centered frame of reference. In *Proceedings of the Third International Symposium on Imitation in Animals and Artifacts*, pages 119–127, Hatfield, UK.
- Scasselatti, B. (1999). Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot. *Lecture Notes in Computer Science*, 1562:176–195.
- Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242.

- Schaal, S. and Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084.
- Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational Approaches to Motor Learning by Imitation. *philosophical transactions: biological sciences*, 358(1431):537–547. philosophical transactions: biological sciences (The Royal Society).
- Scharstein, D., Szeliski, R., and Zabih, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42.
- Shin, H. J., Lee, J., Shin, S. Y., and Gleicher, M. (2001). Computer puppetry: An importance-based approach. *ACM Transactions on Graphics*, 20(2):67–94.
- Shreiner, D., Woo, M., Neider, J., and Davis, T. (2005). *OpenGL(R) Programming Guide : The Official Guide to Learning OpenGL(R), Version 2 (5th Edition)*. Addison-Wesley Professional.
- Simoncelli, M., Zunino, G., Christensen, H., and Lange, K. (2000). Autonomous pool cleaning: Self localization and autonomous navigation for cleaning. *Autonomous Robots*, 9(3):261–270.
- Smola, A. and Scholkopf, B. (1998). A tutorial on support vector regression. NEUROCOLT Technical Report NC-TR-98-030, Royal Holloway College, London.
- Smyth, M. M. and Pendleton, L. R. (1990). Space and movement in working memory. *The Quarterly Journal of Experimental Psychology Section A*, 42(2):291–304.
- Stanislavsky, K. (1936). *An actor prepares*. London: Methuen.
- Sugihara, T. (2009). Standing stabilizability and stepping maneuver in planar bipedalism based on the best com-zmp regulator. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 1966–1971, Kōbe, Japan.
- Sun, Y. and Fisher, R. (2003). Object-based visual attention for computer vision. *Artificial Intelligence*, 146(1):77–123.
- Tang, Z., Zhou, C., and Sun, Z. (2004). Gait planning for soccer playing humanoid robots. *Lecture Notes in Control and Information Sciences*, 299:241–262.
- Taylor, J. R. (1997). *An Introduction to Error Analysis*. University Science Books, Sausalito, California, 2nd edition.
- Terrillon, J. and Akamatsu, S. (1999). Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. In *Proceedings of the 12th Conference on Vision Interface*, volume 2, pages 180–187.

- Thorpe, W. (1963). *Learning and instinct in animals*. Harvard University Press, Cambridge, MA, USA.
- Thrun, S. and Mitchell, T. (1993). Integrating inductive neural network learning and explanation-based learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 930–936.
- Treisman, A. and Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, 12(1):97–136.
- Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.
- Ude, A. (1999). Robust estimation of human body kinematics from video. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '99)*, volume 3, pages 1489–1494.
- Ude, A., Atkeson, C. G., and Cheng, G. (2003). Combining peripheral and foveal humanoid vision to detect, pursue, recognize and act. *Proc. of the 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2173–2178.
- Ude, A., Atkeson, C. G., and Riley, M. (2004). Programming full-body movements for humanoid robots by observation. *Robotics and Autonomous Systems*, 47(2–3):93–108.
- Urdiales, C., Pérez, E., Vázquez-Salceda, J., Sánchez-Marre, M., and Sandoval, F. (2006). A purely reactive navigation scheme for dynamic environments using case-based reasoning. *Autonomous Robots*, 21:65–78.
- Vázquez-Martín, R., del Toro, J., Bandera, A., and Sandoval, F. (2005). Data- and model-driven attention mechanism for autonomous visual landmark acquisition. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, pages 3372–3377.
- Vázquez-Martín, R., Martínez, J., del Toro, J. C., Núñez, P., and Sandoval, F. (2006). A software control architecture based on active perception for mobile robotics. *WSEAS Transactions on Circuits and Systems*, 5(6):797–804.
- Čapek, K. (1920). *R.U.R. (Rossum's Universal Robots)*.
- Vijayakumar, S., D'souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634.

- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518, Kauai, USA.
- Viola, P. and Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.
- Vlachos, M., Gunopulos, D., and Das, G. (2004). Rotation invariant distance measures for trajectories. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 707–712.
- Vukobratovic, M. and Borovac, B. (2004). Zero-moment point- thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173.
- Wada, K. and Shibata, T. (2007). Social effects of robot therapy in a care house - change of social network of the residents for two months. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1250–1255.
- Walker, C. and Walker, E. (2001). *Game Modeling Using Low Polygon Techniques (Charles River Media Graphics)*. Charles River Media, 1st edition.
- Walter, W. G. (1950). An imitation of life. *Scientific American*, pages 42–45.
- Williams, C. and Rasmussen, C. (1996). *Gaussian processes for regression*, volume 8. MIT Press.
- Wilson, A. and Bobick, A. (1999). Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900.
- Wu, B. and Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detection. In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05)*, pages 15–21, Beijing, China.
- Yamane, K. and Nakamura, Y. (2003). Dynamics filter - concept and implementation of on-line motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432.
- Yang, J., Xu, Y., and Chen, C. (1994). Hidden markov model approach to skill learning and its application to telerobotics. *IEEE Transactions on Robotics and Automation*, 10(5):621–631.
- Yang, J., Xu, Y., and Chen, C. (1997). Human action learning via hidden markov model. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27(1):34–44.

- Yang, M. H., Kriegman, D., and Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):34–58.
- Yokoi, K., Nakashima, K., Kobayashi, M., Mihune, H., Hasunuma, H., and Yanagihara, Y. (2006). A tele-operated humanoid operator. *International Journal of Robotics Research*, 25(5-6):593–602.
- Yow, K. and Cipolla, R. (1997). Feature-based human face detection. *Image and Vision Computing*, 15(9):713–735.
- Yussof, H., Ohka, M., Takata, J., Yamano, M., and Nasu, Y. (2007). Application of contact-based sensors for self-localization and object recognition in humanoid robot navigation tasks. In *Proceedings of the 16th IEEE International Symposium on Robot and Human interactive Communication (RO-MAN 2007)*, pages 188–193.
- Zöllner, R., Rogalla, O., Dillman, R., and Zöllner, M. (2002). Understanding users intention: programming fine manipulation tasks by demonstration. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 1114–1119.



# Appendix A

## Vision-based perceptual systems

### A.1 Outline

Among the factors that influence the perceptual capabilities of a robot, one of the most important is the sensory equipment that it uses to collect data from the environment. In HRI scenarios, different sensory inputs have been explored in order to find the optimal way to transfer information from the human demonstrator to the robot. Following the classification adopted by [Calinon \(2007\)](#) for HMC systems, we divide the different perception systems into non-vision based systems and vision based systems.

### A.2 Non-vision based perception systems

Different types of sensors can be used to transfer information from the environment to the robot. Thus, [Breazeal et al. \(2003\)](#) use a microphone and a real-time, low level speech processing and recognition software to be able not only to transfer spoken commands to a social robot, but also to make it recognize the spoken affective intent of the human partner. Social robots that are equipped with more than one microphone, such as the one presented by [Metta et al. \(2000\)](#), may also be able to locate the sound source in space, under certain circumstances, by using directional tuning. This can help focusing attention or locating interaction partners.

A social robot should not only be able to perceive sounds. In human interaction scenarios body language and social signs give important information about intentions, orders or the inner state of the performer ([Breazeal et al., 2003](#)). In order to provide a social robot with the ability to interact with humans in HRI scenarios, the robot should be able to sense and interpret these movements.

Information concerning human body gestures can be captured by attaching different types of magnetic, inertial or mechanical sensors to the performer's body (Safonova et al., 2003; Ude et al., 2004; Calinon, 2007). The 3D positions and/or orientations of these sensors can be recorded for a certain gesture, thus providing a complete description of it as long as the markers have been meaningfully located. These motion capture systems do not suffer from occlusions, provide accurate information about the movement and usually are not computationally expensive thus very high capture rates can be achieved. But they present several drawbacks when applied to HRI scenarios: they require the human performer to wear special markers, they usually need to be calibrated before use and they may also impose restrictive constraints to the environment in which they are going to be used (i.e. magnetic motion sensors are sensitive to interferences produced by low-frequency current-generating devices such as CRT-displays). Thus, they are only suitable for RLBI scenarios that involve controlled environments and specific human teachers.

It is usually possible for a social robot to have proprioceptive knowledge about its own inner state. Thus, information about motor positions can be gathered and applied to a forward kinematics chain in order to obtain the robot pose with a high precision. A powerful method that uses this information to teach a social robot new skills or movements is the *kinesthetic teaching* (Calinon, 2007). This teaching process requires the human performer to move the robot articulations or, in general terms, to use an external robotic tracker instead of attaching markers to his/her body. The human then is able to teach the robot by controlling its movements. Different approaches set the robot motors to passive mode (Peshkin et al., 1996), implement a gravity compensated controller (Heinzmann and Zelinsky, 1999) or use haptic interfaces to provide feedback to the user while he/she is executing the movements (Yokoi et al., 2006). In any case, the idea is to teach the robot a gesture, or how to solve a certain task, by directly controlling it, thus in the future the robot will be able to perform similar gestures or tasks by considering how different human controllers solved them before (Urdiales et al., 2006). While this approach is useful when training can be *a priori* performed in a controlled environment, it may be difficult to use in real, uncontrolled HRI scenarios.

### A.3 Vision based perception systems

As detailed in chapter 1, a social robot should sense and interpret the same phenomena that humans observe (Breazeal, 2002). Visual perception represents the main system that people use to gather information about their environment. Thus, social robots are usually provided with

a vision based perception system in order to be able to address the previous requirement and success in social interaction scenarios and real human environments. This vision system can be used to navigate through the environment, detect landmarks (Vázquez-Martín et al., 2005), identify faces (Barreto et al., 2004), measure head and hands poses, capture human motion (Molina-Tanco et al., 2005; Asfour et al., 2006a), recognize gestures (Kojo et al., 2006; Bandera et al., 2006) and read facial expressions to emulate human social perception. These tasks require high resolution data, but these data must be processed online in order to allow the social robot to act at human interaction rates.

In this sense, from the perspective of social robots, one major drawback of many vision systems is that they are based on a passive reconstructive approach. This approach extracts physical scene parameters from the input images, segments it into meaningful parts and describes the visual input in such a way that the higher-level layers can act on the description to accomplish general tasks. But although progress has been done in this area, it appears that this reconstructive approach is to be nearing its limits without reaching its goals. The alternative to this paradigm is *animate vision* (Breazeal et al., 2001). This approach is based on the principles of active vision and task-oriented techniques (Marfil, 2006), and it is inspired by the foveal nature of the human eyes, in which the distribution of photoreceptors is not uniform in the retina, but is more dense in the center of the visual field (*fovea*). Thus, in the fovea the images are perceived with higher precision, while the resolution decreases as the distance to the fovea increases. This ocular structure is common to many biological entities, and it represents a very efficient system as it allows to analyze with high precision the details of the object in which attention is centered, but it also permits to detect relevant features in the surroundings, such as movements or potential objects of interest (Maini et al., 2002). The aim of animate vision is then to design a visual system that deliberately interacts with the environment by controlling the gaze and moving the focus of attention. This approach drastically reduces the data to be processed and allows real-time functionality, as only the relevant information is extracted from the huge amount of image input data. This selection process also implies the development of an attention mechanism and its integration in the vision system. A side effect of this attentional behaviour is that it also helps the robot to interact socially in a natural way, as attention is, together with expression, one of the capabilities that can make a human to consider that a robot is acting in a rational manner (Fong et al., 2003).

The previously commented advantages make many vision systems for social robots rely on attentional mechanisms and foveal vision to decrease the amount of image data to be processed (Breazeal, 2002; Ude et al., 2003). One of these artificial vision systems is the one used

by Metta et al. in his social robot, *Babybot* (Metta et al., 2000). This agent uses a stereo pair of log-polar foveated cameras. The structure of these cameras physically resembles the distribution of photoreceptors in the human eye, thus it is a good system to analyze human perception and attentional mechanism. The goal of this research is to design and understand complex adaptive systems by studying how biological systems solve the problem of learning and adaptation. *Babybot* achieves these goals and is able to detect relevant features in the environment, center its attention in them and track them as they move. But the complexity of the system, and the low resolution (64x32) of its log-polar cameras, make it unsuitable to perform more precise image processing algorithms such as face detection, gesture recognition or motion capture.

Most social robots use an easier to implement but more effective vision system composed by the two following elements (Ude et al., 2003):

- One or two wide-angle cameras for peripheral vision. These cameras provide low-resolution images that are used to detect relevant features of the environment such as moving objects or regions which a high color contrast respect to the background.
- Two narrow-angle cameras. These high-resolution cameras model the biological fovea and thus they are called foveal cameras. The precise image data they provide is used in complex image processing algorithms to obtain detailed descriptions of detected events and objects. This description can include face detection and recognition, object identification, gesture recognition, precise tracking and other features that require high definition images to be successfully processed.

Finally, the use of zoomed lenses is an alternative to foveated vision. The main problem of these systems is that they are not able to acquire simultaneously wide angle and high resolution images. This drawback strongly limits its use in social robotic applications (Ude et al., 2003).

All the structures commented above need to physically move the cameras -or even the head of the social robot- to place the object of interest in the fovea. Thus, when a relevant feature is detected by the peripheral vision, the system must perform a fast movement -a saccadic movement- to center this feature in the area of maximum resolution (Maini et al., 2002; Aziz and Mertsching, 2007). The set of movements of these vision systems also includes convergence movements and smooth tracking displacements to focus and track objects respectively. All these movements are inspired in biological systems and its use is not only related to efficiency, but also to *expression*. In HRI scenarios, these movements of the artificial vision system provide a natural and intuitive feedback to the human, as they show him where the social robot is centering its

attention (Breazeal, 2002). This expressive component is considered one of the main capabilities that a social robot exhibits (Fong et al., 2003), thus these movements have been kept in most social robot vision systems (Ude et al., 2003).

On the other hand, a disadvantage of these vision systems is their high complexity, as they require to perform many different types of mechanical movements to correctly center the fovea in the object of interest. As a result, most authors tend to give peripheral vision a dominant or even exclusive role in these vision systems. Although there are some exceptions like the work of Breazeal (2002), even systems that use foveated or zoomed lenses usually concentrate on problems that can essentially be solved by using only peripheral vision (Ude et al., 2003).

Another side effect of these vision systems is that they are designed to center its attention in only one relevant feature of the environment. This is a limitation of the biological systems that inspired these approaches, as they also have only one fovea per eye. But the typical saccadic movement of a human eye can reach velocities of 1000 radians per second. This speed allows the eye to perform between 5 and 50 saccadic movements per second. These gaze movements are very rough and unstable, and they require to be followed by a slow vestibular-driven stabilization phase. An artificial vision system requires a complex mechanical structure to perform similar movements, specially if we consider that saccadic movements must usually be followed by slower head movements. Besides, physical limitations of common DC motors and controllers constrain the maximum velocity of gaze shifts, thus it is a very difficult task to achieve results that can be compared with real eye responses (Maini et al., 2002). In this sense, what is a good solution in biological entities becomes strongly inefficient in artificial systems, in which focusing on only one item means that it would not be possible to move the attention to other relevant features at desired rates. A solution to this problem is the use of systems in which the fovea is able to move from one part of the Field Of View (FOV) to another, thus camera movements are not required unless the object of interest moves out the FOV (Arrebola et al., 1998). Another option is the use of multiple resolution structures to represent the perceived images. These structures allow to process image data at different resolution levels so that attention can be easily moved to one or more regions of interest (Marfil et al., 2006).

As commented above, a vision system that is able to focus on certain regions of interest does necessarily need to integrate an attention mechanism that locates, tracks and classifies these regions or objects of interest. Probably one of the most influential theoretical models of visual attention is the spotlight metaphor introduced by Eriksen and Yenb (1985), that has inspired many concrete computational models (Koch and Ullman, 1985; Itti, 2002). These approaches are related with the *feature integration theory*, a biologically plausible theory proposed to ex-

plain human visual search strategies (Treisman and Gelade, 1980). According to this model, these attention mechanisms are organized into two main stages. First, in a preattentive task-independent stage, a number of parallel channels compute image features. The extracted features are integrated into a single saliency map which codes the saliency of each image region. The most salient regions are selected from this map. Second, in an attentive task-dependent stage, the spotlight is moved to each salient region to analyze it in a sequential process. Analyzed regions are included in an *inhibition map* to avoid the spotlight moving to an already visited region. Thus, while the second stage must be redefined for different systems, the preattentive stage is general for any application. Although these models have good performance in static environments, they cannot in principle handle dynamic environments due to their impossibility to take into account the motion and the occlusions of the objects in the scene. In order to solve this problem, Maki et al. (2000) propose an attention mechanism which incorporates depth and motion as features for the computation of saliency.

The previously described methods deploy attention at the level of space locations (*space-based models of visual attention*). The models of space-based attention scan the scene by shifting attention from one location to the next to limit the processing to a variable size of space in the visual field. Therefore, they have some intrinsic disadvantages. In a normal scene, objects may overlap or share some common properties. Then, attention may need to work in several discontinuous spatial regions at the same time. Only if different visual features, which constitute the same object, come from the same region of space, an attention shift will not be required (Sun and Fisher, 2003). On the contrary, other approaches deploy attention at the level of objects instead to a generic region of space. *Object-based models of visual attention* provide a more efficient visual search than space-based attention. Besides, it is less likely to select an empty location. In the last few years, these models of visual attention have received an increasing interest in computational neuroscience and in computer vision. These models reflect the fact that the perception abilities must be optimized to interact with objects and not just with disembodied spatial locations. Thus, visual systems that follow this approach will segment complex scenes into objects which can be subsequently used for recognition and action, such as the *percepts* used by Breazeal et al. (2005).

Finally, space-based and object-based approaches are not mutually exclusive, and several researchers have proposed attentional models that integrate both approaches. Thus, Sun and Fisher (2003) propose to model visual attention by combining object-based and feature-based theories. In its current form, this model is able to replicate human viewing behaviour. However, it needs input images to be manually segmented. That is, it uses information that is not available

in a preattentive stage, before objects are recognized (Orabona et al., 2007).

In this thesis, a simplified object-based model of visual attention is employed, that searches for human faces in the perceived images. Once a face is detected, a new object-based search starts that detects skin color regions inside the human silhouette, extracted using the method described in chapter 3. A fast, hierarchical tracking approach is then employed to deal with the movements of these regions, and also with the variations in their shapes produced by motion itself and minor illumination differences between consecutive acquired images. This tracking approach is deeply explained in appendix D.



## Appendix B

# Face detection

The appearance of a face in the image is influenced by different factors, that should be considered when designing a face detector.

- Presence of structural components in the face as beard, mustache, hat or glasses.
- Pose and orientation.
- Facial expression.
- Variations in the lighting conditions.
- Occlusions.
- Noise.

Face detection methods can be classified into four categories (Yang et al., 2002):

1. Knowledge-based methods (Yang et al., 1994): this category includes any rule-based face detection approach. These rules usually describe the features of a face and their relationships. For example: a face often contains two eyes that are symmetric to each other, a nose and a mouth. The relationships between them are their relative distances and positions. The most important problem of this type of methods is the difficulty to extract the relevant human knowledge and represent it using rules.
2. Feature-based methods: these methods try to find invariant features of faces. The most used features are eyebrows, eyes, nose, mouth and hairline (Yow and Cipolla, 1997). A problem with these features is that they can be corrupted due to illumination, noise and

occlusions. Other used features are texture (Dai and Nakano, 1996) and skin colour (Terrillon and Akamatsu, 1999).

3. Template matching methods: a standard pattern of a face, or some patterns of face features as nose, eyes and mouth, are stored as template. This template can be a fixed template (Craw et al., 1992), which has problems with variations in scale, pose and shape, or a deformable template, that is more robust against scale and pose, but it still very dependant on shape (Lanitis et al., 1995).
4. Appearance-based methods (Turk and Pentland, 1991): in these methods, models are learned from a set of training images which should capture the representative variability of facial appearance. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face and non-face images. The learned characteristics are in the form of distribution models or discriminant functions.

In this thesis, the feature-based method for face detection proposed by Viola and Jones (2004) is employed. This method uses Haar-like features to detect faces. Haar-like features encode the existence of oriented contrasts in the input image. This method has proven to be fast (15 frames per second in a conventional desktop), thus it is adequate for RLbI scenarios. Two of the main characteristics of the method, that are exploited in this thesis, are the following:

- The use of a set of features which are reminiscent of Haar Basis functions. In order to compute these features very quickly at many scales, they introduce a new image representation called integral image. It can be computed from an image using a few operations per pixel. Once computed, any of the Haar-like features can be calculated at any scale or location in constant time.
- A simple and efficient classifier is used to select a small number of important features from the huge amount of potential ones. As proposed by Viola and Jones (2004), this classifier is built using the AdaBoost learning algorithm (Freund and Schapire, 1995), that selects a small set of critical face features from a large set of features.

The employed face detection algorithm is briefly detailed below. See Viola and Jones (2004) for further details.

## B.1 Features

The detector uses the four types of Haar-like features detailed below and depicted in Fig. B.1:

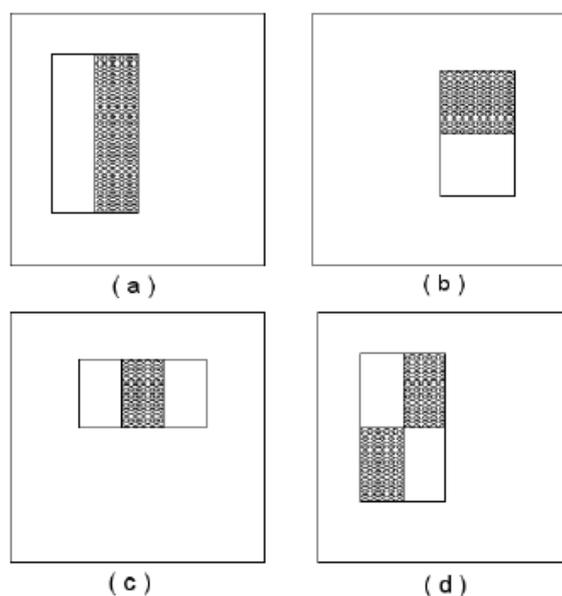


Figure B.1: Haar-like features used to detect faces: (a) Vertical two-rectangle; (b) horizontal two-rectangle; (c) three rectangle; and (d) four-rectangle.

- *Two-rectangle feature.* This is the difference between the sum of the pixels within two rectangular regions of the input image. The regions have the same size and shape and are horizontally or vertically adjacent. It is subdivided in two subclasses: (i) vertical two-rectangle, in which two rectangles vertically adjacent are considered; and (ii) horizontal two-rectangle, that uses two rectangles horizontally adjacent.
- *Three-rectangle feature.* It computes the sum within two outside rectangles subtracted from the sum in a center rectangle.
- *Four-rectangle feature,* that computes the difference between diagonal pairs of rectangles.

Fig. B.1 shows the used features. The sum of the pixels which are within the white rectangles are subtracted from the sum of the pixels in the grey rectangles.

## B.2 Integral image

In order to achieve fast computation of the previously explained features, [Viola and Jones \(2004\)](#) propose to use the integral image. The integral image is an intermediate representation for the image which at location  $(i, j)$  contains the sum of the pixels above and to the left of  $(i, j)$  inclusive.

It is possible, using the integral image, to compute the sum of the pixels within any image rectangle with only four memory accesses ([Viola and Jones, 2004](#)). A two-rectangle feature is computed with six memory accesses. A three rectangle feature needs eight memory accesses and nine memory accesses are needed to compute a four-rectangle feature (Fig. B.2).

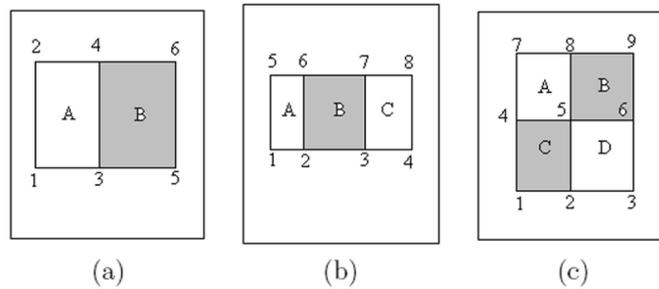


Figure B.2: Memory accesses needed to compute: (a) a two-rectangle feature; (b) a three-rectangle feature; (c) a four-rectangle feature. For example the two-rectangle feature is computed as:  $B - A = (5 - 6 - 3 + 4) - (3 - 4 - 1 + 2)$ .

## B.3 Classifier learning process: Adaboost algorithm

The Adaboost algorithm is used to select a small set of critical features ( $T$ ) from the huge set of previously computed features. Besides, it is used to train the classifier. The Adaboost learning algorithm consists of  $T$  weak classifiers (one for each feature) which are combined to form a strong classifier. Each weak classifier is designed to select the single rectangle feature which best separates the positive and negative examples. For each feature, the weak learning process determines the optimal threshold classification function, such that the minimum number of training images are misclassified. Therefore, a weak classifier ( $h(x, f, p, \theta)$ ) consists of a feature  $f$ , a threshold  $\theta$  and a polarity  $p$  indicating the direction of the inequality (Eq. B.1).

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.1})$$

Table B.1: Percentages of rightly classified faces, false positives and false negatives.

	Indoor images	Outdoor images
Rightly classified	86%	82%
False negatives	14%	18%
False positives	10%	8%

Given  $n$  example images  $(x_1, y_1), \dots, (x_n, y_n)$ , where  $y_i = 0$  for negative (non-face) examples and  $y_i = 1$  for positive (face) images, the following process is applied to select the optimum set of weak classifiers from the whole set of possible weak classifiers (Viola and Jones, 2004).

- Initialize weights  $\omega_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ , for  $y_i = 0, 1$  respectively, where  $m$  and  $l$  are the number of negatives and positives, respectively.
- For  $t = 1, \dots, T$ :

1. Normalize the weights,  $\omega_{t,i,norm} = \frac{\omega_{t,i}}{\sum_{j=1}^n \omega_{t,j}}$ .

2. Select the best weak classifier with respect to the weighted error  $\epsilon_t$ , computed using Eq. B.2 (see Viola and Jones (2004) for further details).

$$\epsilon_t = \min_{f,p,\theta} \sum_i \omega_i |h(x_i, f, p, \theta) - y_i| \quad (\text{B.2})$$

3. Define  $h_i(x) = h(x, f_t, p_t, \theta_t)$  where  $f_t, p_t$  and  $\theta_t$  are the minimizers of  $\epsilon_t$ .
4. Update the weights  $\omega_{t+1,i} = \omega_{t,i} \cdot \beta_t^{1-e_i}$ , where  $e_i = 0$  or  $1$  if the image  $x_i$  is correctly or incorrectly classified, respectively, and  $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ .
5. Compute the weights  $\alpha$  of the final strong classifier as  $\alpha_t = \log(1/\beta_t)$ .

The final strong classifier is a lineal combination of the previously selected weak classifiers. It is represented using Eq. B.3.

$$C(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T \alpha_t \cdot h_t(x) \geq 0.5 \cdot \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.3})$$

This classifier allows to decide whether a certain feature is a face ( $C(x) = 1$ ) or not ( $C(x) = 0$ ). Table B.1 shows results obtained when the number of training iterations and final features are set to  $T = 150$ .



## Appendix C

# Canny edge detector

In order to extract edges from input images, [Canny \(1983\)](#) proposes to firstly smooth these images using a Gaussian mask, and then apply a Sobel second order operator to detect significant changes in the pixel values. This second order operator is defined by the following equations:

$$G_x = \begin{pmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{pmatrix} \quad (\text{C.1})$$

$$G_y = \begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix} \quad (\text{C.2})$$

$$G = \sqrt{(G_x)^2 + (G_y)^2} \quad (\text{C.3})$$

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (\text{C.4})$$

where equations [C.1](#) and [C.2](#) defines the Sobel operator in each direction (horizontal and vertical). Eq. [C.3](#) represents the module of the Sobel operator and Eq. [C.4](#) its argument. This argument  $\Theta$  is grouped in only four directions to adapt to digitalized images (pixel maps), e.g. arguments in the interval  $[337.5^\circ, 22.5^\circ)$  are simplified as  $0^\circ$ . These four directions are shown in [Fig. C.1](#).

After the module  $G$  of the Sobel operator has been computed for all image pixels, a *non-maximum suppression* step is applied. This step marks as borders only those pixels which  $G$  meets certain criterion. This criterion is based on the use of two thresholds and hysteresis. Thus, pixels which  $G$  are over the higher threshold  $g_h$  are marked as borders. Pixels which  $G$

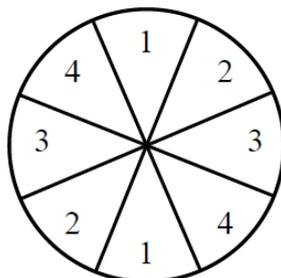


Figure C.1: Directions in which the Sobel argument is grouped.

are under the lower threshold  $g_l$  are marked as non-borders. Finally, pixels that are in the zone between both thresholds are marked as borders if and only if at least one of their 8 neighbors has been marked as a border. [Canny \(1983\)](#) suggests that these thresholds should meet Eq. [C.5](#).

$$(2 \cdot g_l) \leq g_h \leq (3 \cdot g_l) \tag{C.5}$$

Apart from this suggestion, thresholds values are a parameter of the method that can be set to very different values according of the task to accomplish. In this thesis silhouette borders should be detected, thus thresholds are set following these steps:

- Equations [C.1](#) and [C.2](#) are respectively applied over two regions in the disparity map. The first of these regions contains pixels located at the right of the face bounding box. The other region contains pixels located at the top of the face. The size of these regions is set to contain a transition from the face to the background if no occluding objects are present (Fig. [C.2](#)).
- The maximum  $G_x$  value obtained for the first region and the maximum  $G_y$  value obtained for the second region are selected to set the higher threshold,  $g_h$ , using Eq. [C.3](#).
- $g_l$  is set as  $g_h/2$ .

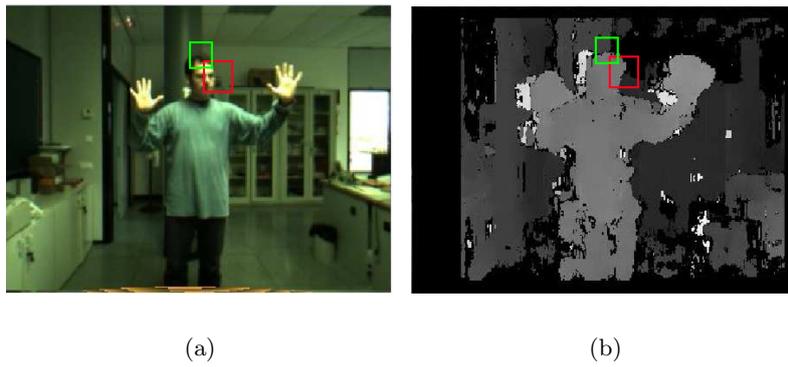


Figure C.2: Search regions used to obtain  $g_h$  value, depicted over: (a) left input image; and (b) disparity map.



## Appendix D

# Tracking using BIPs

Detected skin color regions representing face and hands are not searched in the complete image for each particular frame of the perceived sequence. Instead, they are locally tracked using a tracking algorithm that is detailed in this appendix.

The proposed tracking method uses a weighted template for each region to track which follows its viewpoint and appearance changes. These weighted templates and the way they are updated allow the algorithm to successfully handle partial occlusions. To reduce the computational cost, templates and targets are hierarchically modeled using Bounded Irregular Pyramid (BIP) that have been modified to deal with binary images (Marfil et al., 2004; Molina-Tanco et al., 2005). This BIP (Marfil et al., 2006, 2007) is a mixture of regular and irregular pyramids: a  $2 \times 2/4$  regular structure is used in the homogeneous regions of the input image and a simple graph structure in the non-homogeneous ones. The mixture of both structures generates an irregular configuration which is described as a graph hierarchy in which each level  $G_l = (N_l, E_l)$  consists of a set of nodes,  $N_l$ , linked by a set of intra-level edges  $E_l$ . Each graph  $G_{l+1}$  is built from  $G_l$  by computing the nodes of  $N_{l+1}$  from  $G_l$  and establishing the inter-level edges  $E_{l,l+1}$ . Therefore, each node  $n_i$  of  $G_{l+1}$  has associated a set of nodes of  $G_l$ , which is called the *reduction window* of  $n_i$ . This includes all nodes linked to  $n_i$  by an inter-level edge. The node  $n_i$  is called *parent* of the nodes in its reduction window, which are called *children*. The successive levels of the hierarchy are built using the following regular decimation process and union-find strategy (Marfil et al., 2007). Therefore, there are two types of nodes: nodes belonging to the  $2 \times 2/4$  structure, named regular nodes, and virtual nodes or nodes belonging to the irregular structure. In any case, two nodes  $n_i \in N_l$  and  $n_j \in N_l$  which are neighbors at level  $l$  are linked by an intra-level edge  $e_{ij} \in E_l$ .

The proposed approach uses a BIP structure to accomplish the detection of the tracked

skin color regions. Each node  $n$  in the binary BIP is identified by  $(i, j, l)$ , where  $l$  represents the level and  $(i, j)$  are the co-ordinates within the level. Besides, node  $n$  will have two associate parameters:

- Homogeneity,  $Hom(i, j, l)$ . This is set to 1 if the four nodes immediately underneath have homogeneity values equal to 1. Otherwise, it is set to 0. It must be noted that in the base of the structure (level 0) only nodes which correspond to skin color pixels of the tracked regions have homogeneity values equal to 1.
- Parent link,  $(X, Y)_{(i, j, l)}$ . If  $Hom(i, j, l)$  is equal to 1, the values of the parent link of the four nodes immediately underneath are set to  $(i, j)$ . Otherwise, these four parent links are set to a null value.

Fig. D.1 shows how the  $l + 1$  level of the binary BIP is constructed (green vertex correspond to vertex which homogeneity is equal to 1). This process is briefly explained below (see Marfil et al. (2007) for further details):

1. *Regular decimation process.* The first step is to generate nodes which homogeneity is equal to 1 (regular nodes). These nodes are linked to their sons. This step creates an incomplete  $l + 1$  level that will be completed further (Fig. D.1.a).
2. *Parent search.* After the previous step is executed, there are some regular *orphan* nodes (regular nodes without parent). From each of these nodes  $(i, j, l)$ , a search is made for a neighbour vertex with a parent in the level  $l + 1$ . If this neighbor node is found, the node  $(i, j, l)$  is linked to the parent of this neighbor node (Fig. D.1.b).
3. *Intra-level twinning.* Once the regular part of the BIP has been created, the irregular part is built. Thus, if for a certain node  $(i, j, l)$  a parent is not found, then a search is made for a neighbour orphan vertex at the same level. If this node is found, then both nodes are linked, or twinned, generating a virtual node at level  $l + 1$  (Fig. D.1.c). As the algorithm performs the steps depicted in Fig. D.1.b and Fig. D.1.c simultaneously, it is possible -and usual- that an orphan vertex is linked to the irregular parent of a neighbour.
4. *Intra-level edge definition.* Finally, intra-level edges are generated at level  $l + 1$  to define the bounds of the receptive fields (Fig. D.1.d), where a receptive field is defined as the set of sons of one vertex in the base level.

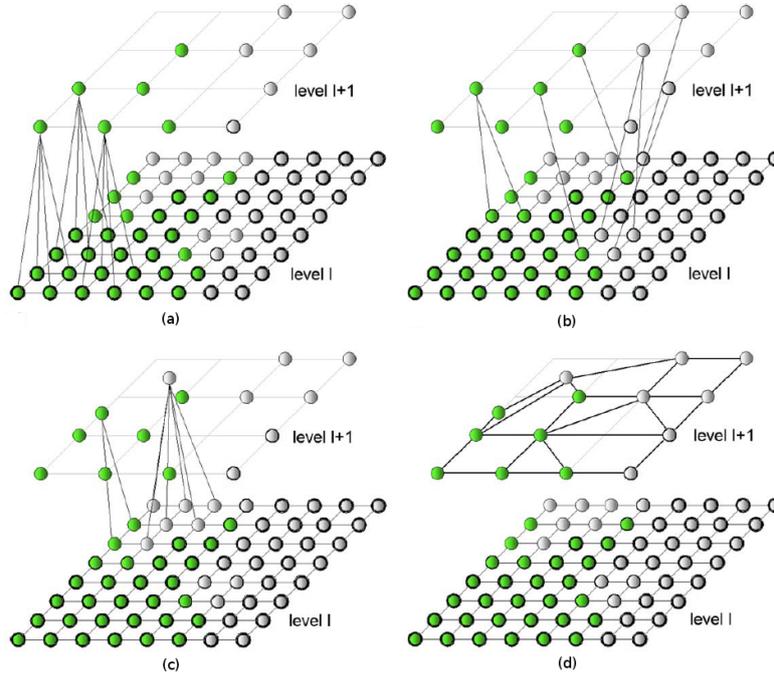


Figure D.1: Binary BIP level generation: (a) regular step (non-orphan vertices of level  $l$  have been marked); (b) parent search; (c) intra-level twining at level  $l$ ; and (d) intra-level edge definition at level  $l + 1$  (marked in black).

Each template  $M$  and target  $T$  is represented by using one of these binary BIP structures, as Eq. D.1 depicts.

$$\begin{aligned}
 M^{(t)}(l) &= \bigcup_{ij} m^{(t)}(i, j, l) \\
 T^{(t)}(l) &= \bigcup_{ij} q^{(t)}(i, j, l)
 \end{aligned}
 \tag{D.1}$$

$M^{(t)}(l)$  and  $T^{(t)}(l)$  being the level  $l$  of the pyramid structure corresponding to the template and target in frame  $t$ , respectively. Each level of the template and the target are made up of a set of homogeneous nodes. Fig. D.2 shows the different hierarchical levels of the BIP associated to one of these templates. In this case, the template corresponds to a tracked hand.

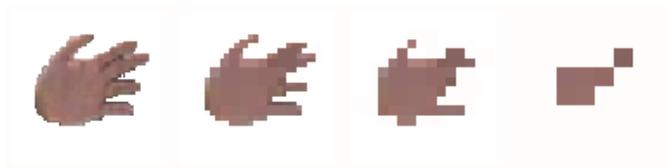


Figure D.2: Levels of a template, modeled as a BIP structure, that represents a tracked hand.

The tracking process is initialized by building hierarchical representations (binary BIPs)

for the three largest skin color regions inside the silhouette of the performer. These hierarchical structures are the first templates and their spatial positions are the first Region Of Interest (ROI), i.e. the portions of the current frame where each target is more likely located. Once initialized, the proposed tracking algorithm follows the data flow shown in Fig. D.3. It consists of four main steps which are described below.

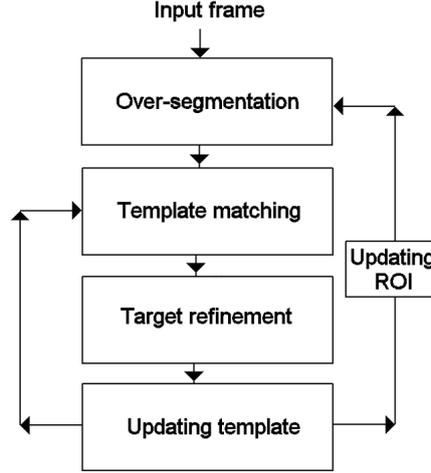


Figure D.3: Data flow of the tracking algorithm.

## D.1 Oversegmentation

The hierarchical representation in the current frame  $t$  of a ROI,  $ROI^{(t)}$ , depends on the target position in the previous frame, and is updated as described in Section D.4. The hierarchical structure can be represented in each level as:

$$ROI^{(t)}(l) = \bigcup_{ij} p^{(t)}(i, j, l) \quad (D.2)$$

being  $p$  a node of the bounded irregular pyramid built over the ROI.

## D.2 Template matching and target refinement

The process to localize the target in the current frame  $t$  is a top-down process which starts at a working level  $l_w^{(t)}$  and stops in the level where the target is found. In each level  $l$ , the template  $M_i^{(t)}(l)$  is placed and shifted in  $ROI^{(t)}(l)$  until the target is found or until  $ROI^{(t)}(l)$  is completely covered. If  $ROI^{(t)}(l)$  was completely covered and the target was not found, the

target localization would continue in the level below. The displacement of the template can be represented as  $d_k^{(t)} = (d_k^{(t)}(i), d_k^{(t)}(j))$ , being  $d_0^{(t)}$  the first displacement and  $d_f^{(t)}$  the final displacement.  $d_f^{(t)}$  is the displacement that situates the template in the position where the target is placed in the current frame. The algorithm chooses as initial displacement in the current frame  $d_0^{(t)} = d_f^{(t-1)}$ . In order to localize the target and obtain  $d_f^{(t)}$ , the overlap  $O_{d_k^{(t)}}^{(t)}$  between  $M^{(t)}(l)$  and  $ROI^{(t)}(l)$  in each template displacement  $k$  is calculated using Eq. D.3.

$$O_{d_k^{(t)}}^{(t)} = \sum_{ij \in \xi} w^{(t)}(m(i, j, l_w^{(t)})) \quad (\text{D.3})$$

being  $w^{(t)}(m(i, j, l))$  a weight associated to  $m^{(t)}(i, j, l)$  in the current frame  $t$ , as explained in section D.3.  $\xi$  is the subset of nodes that satisfy the following two conditions:

$$\begin{aligned} Hom(f(m^{(t)}(i, j, l_w^{(t)}), a(t))) &= 1 \\ Hom(p^{(t)}(i + d_k^{(t)}(i), j + d_k^{(t)}(j), l_w^{(t)})) &= 1 \end{aligned} \quad (\text{D.4})$$

being  $f(m^{(t)}(i, j, l_w^{(t)}), a(t))$  a coordinate transformation of  $m^{(t)}(i, j, l_w^{(t)})$  that establishes the right correspondence between  $m^{(t)}(i, j, l_w^{(t)})$  and  $p^{(t)}(i + d_k^{(t)}(i), j + d_k^{(t)}(j), l_w^{(t)})$ .  $a(t)$  denotes the parameter vector of the transformation, which is specific for the current frame. Equation D.4 is satisfied when a match occurs. All the ROI nodes that match with nodes of the template are marked as nodes of the target. Thus, the hierarchical representation of the target  $T^{(t)}$  is obtained.

In order to refine the target appearance, its hierarchical representation is rearranged level by level following a top-down scheme (Marfil et al., 2004). This process is applied to all homogeneous nodes of ROI which have not been marked as target nodes in the template matching process. If one of these nodes has a homogeneous neighbour node that belongs to the target, it is also marked as a target node.

## D.3 Template updating

In order to update the template, a new parameter is included in the template model:

- $w^{(t)}(m(i, j, l))$ . It is a weight associated to each node  $m^{(t)}(i, j, l)$  of the template  $M^{(t)}$  in the current frame  $t$ .

The whole template is updated at each sequence frame:

$$\begin{aligned}
 m^{(t+1)}(i, j, l) &= \begin{cases} m^{(t)}(i, j, l) & \text{if no match} \\ f^{-1}(q^{(t)}(i, j, l), a^{(t)}) & \text{if match} \end{cases} \\
 w^{(t+1)}(m(i, j, l)) &= \begin{cases} w^{(t)}(m(i, j, l)) - \alpha & \text{if no match} \\ 1 & \text{if match} \end{cases}
 \end{aligned} \tag{D.5}$$

where the superscript  $(t)$  denotes the current frame and the forgetting constant,  $a$ , is a predefined coefficient that belongs to the interval  $[0,1]$ . This constant dictates how fast the forgetting action will be.

## D.4 Region of interest updating

This process has two main steps:

1. *ROI<sup>(t+1)</sup>(0) selection*: Level 0 of the new region of interest is obtained by taking into account the position where the target is placed in the original image of the frame  $t$ . Firstly, the bounding-box of  $T^{(t)}(0)$ ,  $(BB(T^{(t)}(0)))$ , is computed. Then,  $ROI^{(t+1)}(0)$  will be made up of the pixels of the next frame  $p^{(t+1)}(i, j, l)$  which are included in the bounding box plus the pixels included in an extra border  $\varepsilon$  of the bounding box. This extra border tries that the target in the next frame will be placed in the new ROI.

$$ROI^{(t+1)}(0) = \bigcup_{ij \in (BB(T^{(t)}(0)) + \varepsilon)} p^{(t+1)}(i, j, 0) \tag{D.6}$$

2. *Over-segmentation of ROI<sup>(t+1)</sup>(0)*: The hierarchical structure  $ROI^{(t+1)}$  is built. This step is performed at the beginning of the tracking process  $t + 1$  (Section D.1).

## Appendix E

# FIR filter employed to smooth perceived hand motion

Perceived hand trajectories are smoothed by applying a FIR filter to X,Y,Z hand coordinates. The impulse response of these FIR filters,  $\vec{h}$ , is obtained using Eq. E.1. It can be seen that these filters can be easily tuned by modifying only two parameters: (i) the number of coefficients of the filter,  $N$ ; and (ii) the slope of the response,  $p$ .

$$\begin{aligned} h_0 &= \frac{2 \cdot e^{\ln(p)}}{2 \cdot e^{\ln(p)} + \sum_{j=1}^{(N-1)} e^{\ln\left(\frac{p}{N} \cdot (N-j)\right)}} \quad j = 0 \\ h_j &= \frac{e^{\ln\left(\frac{p}{N} \cdot (N-j)\right)}}{2 \cdot e^{\ln(p)} + \sum_{j=1}^{(N-1)} e^{\ln\left(\frac{p}{N} \cdot (N-j)\right)}} \quad \forall j \in [1 \dots (N-1)], j \in \mathbb{N} \end{aligned} \tag{E.1}$$

Fig. E.1 shows the impulse responses obtained when  $N = 11$  and  $p$  equals 100 and 5. It can be seen that higher  $p$  values produce faster responses, while smaller  $p$  values correspond to low-pass filters that have a stronger dependence on past samples. The option to provide a different  $p$  value for each coordinate is interesting since depth information tends to be noisier in stereo systems. Thus, this coordinate of hand trajectories should be filtered using lower  $p$  values.

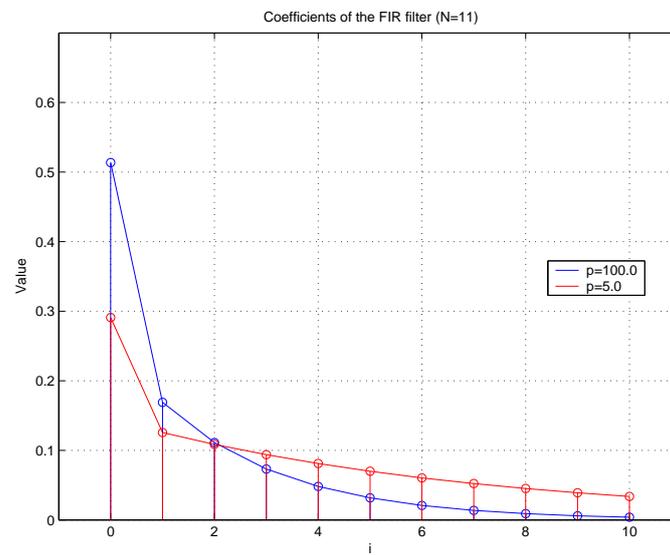


Figure E.1: Coefficients of the FIR filter used to smooth one of the coordinates of a hand trajectory.

# Appendix F

## Inverse kinematics algorithm

In this thesis, an analytic IK algorithm is used to pose model arms. This algorithm considers a simplified model of an arm, that consists on a two-bone kinematic chain (Fig. F.1). The parent bone corresponds to the upper arm and is allowed to rotate around three perpendicular axes. This provides a simplified model of the shoulder joint. The child bone corresponds to the forearm and is allowed only to flex, thus only one DOF is located in the elbow.

$T({}_1^w R)$  is the local transformation between the upper-arm reference frame  $O_1$  and a coordinate frame attached to the torso and centered at the shoulder joint  $w$ . The bone representing the lower arm is allowed to rotate around a single axis, corresponding to the elbow joint.  $T({}_1^2 R, {}^1 \vec{l}_1)$  denotes the local transformation between the upper-arm reference frame  $O_1$  and the lower-arm reference frame  $O_2$ , where  ${}^1 \vec{l}_1 = (0, 0, l_1)^T$ , being  $l_1$  the length of the upper-arm, and  ${}^2_1 R$  corresponds to the rotation  $\theta_e$  about the elbow axis.

Given a desired position for the end-point of the arm at time instant  $t + 1$ ,  ${}^w \vec{p}_d^{(t+1)}$ , and given the rotation matrices  ${}^w_1 R^{(t)}$  and  ${}^2_1 R^{(t)}$  at the previous time instant  $t$ , the problem is then to find the updated matrices  ${}^w_1 R^{(t+1)}$  and  ${}^2_1 R^{(t+1)}$ . A simple geometric method is summarized here that can solve such problem. See [Mitchelson \(2003\)](#) for further details.

1. Bring  ${}^w \vec{p}_d^{(t+1)}$  within reach of the arm:

$$\text{if } |{}^w \vec{p}_d^{(t+1)}| > (l_1 + l_2) \quad \text{then } {}^w \vec{p}_d^{(t+1)} \leftarrow {}^w \vec{p}_d^{(t+1)} \frac{l_1 + l_2}{|{}^w \vec{p}_d^{(t+1)}|}$$

2. Compute elbow circle: . Posing the model arms is an under-constrained problem, as four degrees of freedom must be specified from only three constraints, corresponding to the coordinates of the desired end-point position  ${}^w \vec{p}_d^{(t+1)}$ . The elbow circle is defined as the set of positions that the elbow is free to adopt when the end-point of the arm reaches  ${}^w \vec{p}_d^{(t+1)}$ .

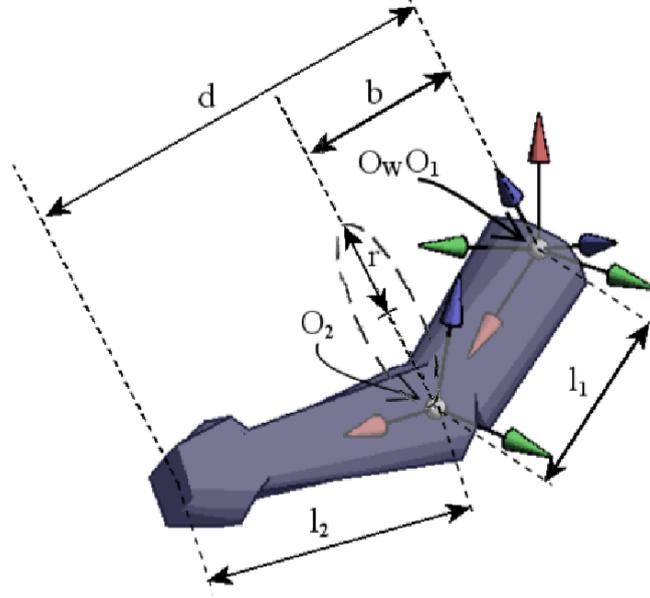


Figure F.1: Kinematic model of the human arm showing local coordinate frames and elbow circle.

It has a radius  $r$  and it is contained in a plane perpendicular to the vector  $w_{\vec{p}_d}^{(t+1)}$  at a distance  $b$  to the shoulder joint.

$$r^2 = \frac{(d + l_1 + l_2)(-d + l_1 + l_2)(d - l_1 + l_2)(d + l_1 - l_2)}{2d}$$

$$b = \sqrt{l_1^2 - r^2}$$

where  $d = |w_{\vec{p}_d}^{(t+1)}|$

3. Choose updated elbow axis  $w_{\vec{x}_2}^{(t+1)}$  and location  $w_{\vec{l}_1}^{(t+1)}$ : We chose the elbow axis at time instant  $t+1$  to be the closest to the one at the previous time instant,  $w_{\vec{x}_2}^{(t)}$ :

$$w_{\vec{x}_2}^{(t+1)} = (w_{\vec{p}_d}^{(t+1)} \wedge w_{\vec{x}_2}^{(t)}) \wedge w_{\vec{p}_d}^{(t+1)}$$

$$w_{\vec{l}_1} = b \frac{w_{\vec{p}_d}^{(t+1)}}{|w_{\vec{p}_d}^{(t+1)}|} + r \frac{w_{\vec{x}_2}^{(t+1)} \wedge w_{\vec{p}_d}^{(t+1)}}{|w_{\vec{x}_2}^{(t+1)} \wedge w_{\vec{p}_d}^{(t+1)}|}$$

4. Fill updated rotation matrices  ${}^w_1R^{(t+1)} = ({}^w\vec{x}_1 \ {}^w\vec{y}_1 \ {}^w\vec{z}_1)$  and  ${}^1_2R^{(t+1)} = ({}^1\vec{x}_2 \ {}^1\vec{y}_2 \ {}^1\vec{z}_2)$  with:

$$\begin{aligned} {}^w\vec{x}_1 &= {}^w\vec{x}_2 & {}^1\vec{x}_2 &= (1, 0, 0) \\ {}^w\vec{z}_1 &= w_{\vec{l}_1}/|w_{\vec{l}_1}| & {}^1\vec{z}_2 &= {}^wR_1(w_{\vec{p}_d} - w_{\vec{l}_1}) \\ {}^w\vec{y}_1 &= {}^w\vec{z}_1 \wedge {}^w\vec{x}_1 & {}^1\vec{y}_2 &= {}^1\vec{z}_2 \wedge {}^1\vec{x}_2 \end{aligned}$$

These obtained rotation matrices set the arm joint angles to make the arm end-effector reach the desired 3D position. It can be seen that the whole method uses only analytic relations.

## Appendix G

# Publications of the author

This appendix lists the publications of the author, and provides a brief description of their contents and relation with this thesis. The publications are listed according to the date in which they were published.

### G.1 Publications covered in this thesis

**Bandera, J.P., Molina-Tanco, L., Marfil, R., and Sandoval, F. (2004).** A Model-based Humanoid Perception System for Real-time Human Motion Imitation. In *Proceedings of the 2004 IEEE Conference on Robotics, Automation and Mechatronics (RAM 2004)*.

The main subject of this article is the IK algorithm described in section 3.5.4.2. In this paper, although incorrect poses and collisions are detected to ensure the integrity of the robot, no alternative poses are searched for. The human virtual model is not used in this contribution. Following previous approaches [Demiris and Hayes \(2002\)](#), perceived motion is directly mapped to robot motion space. Experimental results show that the proposed IK algorithm, while incomplete, is a promising alternative to probabilistic approaches.

**Molina-Tanco, L., Bandera, J.P., Marfil, R., and Sandoval, F. (2005).** Real-time Human Motion Analysis for Human-Robot Interaction. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*.

This paper represents a first step towards implementing a mechanism that allows a robot to imitate human motion, in a RLbI framework. Stereo images are obtained by the cameras

mounted on the head of the HOAP-1, described in section 3.6. HMC is achieved using the IK algorithm detailed in section 3.5.4.2. Incorrect poses are not checked in this contribution. Captured motion is translated to HOAP-1 robot, but retargeting consists only on a normalization of the perceived 3D relative positions of hands, respect to the head. This contribution sets some of the main basis of the system and reveals some of the most important issues that have to be addressed. Thus, the analysis of its results directed research towards using different stereo systems, more robust and complex HMC and retargeting algorithms, and in the end, a different robotic platform.

**Bandera, J.P., Marfil, R., Molina-Tanco, L., Bandera, A., and Sandoval, F. (2005). Model-based Pose Estimator for Real-time Human-Robot Interaction. In *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005)*.**

The complete model-based arm pose generator detailed in section 3.5.4 is described in this paper. Thus, the IK algorithm is finally extended with the alternative pose evaluation system. The approach detailed in this paper does not use the human virtual model. It uses the difference between human and robot heights to normalize perceived movements to the robot motion space. Experimental results provide a qualitative evaluation of the complete arm pose generator and reveal its benefits against previously mentioned contributions.

**Molina-Tanco, L., Bandera, J.P., Rodríguez, J.A., Marfil, R., Bandera, A., and Sandoval, F. (2006). A Grid-based Approach to the Body Correspondence Problem in Robot Learning by Imitation. In *Proceedings of the European Robotic Symposium (EUROS 2006), Workshop on Vision Based Human-Robot Interaction*.**

Previous contributions revealed the necessity of replacing the direct mapping approach by more complex retargeting strategies. As commented in this thesis, the reachable spaces may be very different from the human to the robot. These differences difficult obtaining satisfactory imitation results using a simple scale transformation. This paper represents a first attempt to implement more complex retargeting strategies in the considered RLbI scenarios. The proposed approach relies on a discretization of the motion spaces for both the human and the robot, and the use of certain transformation rules that establish a correspondence (usually, a many-to-one correspondence) between these discrete spaces. The results were interesting, but the difficulties of finding transformation rules, the necessity of using a training or body-babbling stage, and the additional errors introduced by the discretization operation, pointed towards the use of a different retargeting method.

**Bandera, J.P., Marfil, R., Molina-Tanco, L., Rodríguez, J.A., Bandera, A., and Sandoval, F. (2006).** Robot learning of upper-body human motion by active imitation. In *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS'06)*.

This paper represents an important milestone in the research process in which this thesis is included. The first approach to a RLbI architecture, presented in section 2.5, was presented in this paper. Its contents were extended in the posterior contribution entitled "Robot learning by active imitation".

**Sandoval, F., Bandera, J.P., and Molina-Tanco, L. (2007).** Perception and Sensor-motor learning system for Humanoid Platforms: TIN2005-01359. In *Jornada de Seguimiento de Proyectos de Tecnología Informática*. MED edition, Zaragoza (Spain), pages 25-32.

This contribution details the general framework in which the subjects covered by this thesis are included. This description of its context helps understanding the requirements imposed to the proposed RLbI system. It also reveals the practical necessities that arises regarding its integration with other systems. Finally, it provides useful information about the limitations of the proposed system, and the direction that will take future work.

**Bandera, J.P., Marfil, R., Molina-Tanco, L., Rodríguez, J.A., Bandera, A., and Sandoval, F. (2007).** Robot learning by active imitation. In *Humanoid Robots: Human-like Machines*, Matthias Hackel edition, Vienna (Austria), pages 519-544.

This paper provides a complete description of the firstly implemented RLbI architecture, presented in section 2.5, although the elements of the perceptual components are notoriously simpler. The main advantages and limitations of this contribution are deeply explained in section 2.5. Besides, this work revealed the necessity of substitute the qualitative evaluations that had been previously used by a more adequate, precise and complete evaluation. Thus, it was after this paper that a new RLbI architecture began to be implemented, considering not only changes in the relations between different elements, but also deep changes in the implementation of these elements themselves. On the other hand, different methods to quantitative evaluate the results obtained by the HMC system began to be studied.

**Bandera, J.P., Marfil, R., Rodríguez, J.A., Molina-Tanco, L., and Sandoval, F. (2008).** A novel hybrid approach to upper-body Human Motion Capture. In *Proceedings of the 14th IEEE Mediterranean Electrotechnical Conference (MELECON)*

2008).

Quantitative evaluation of the proposed HMC system was finally achieved thanks to the experiments conducted at the Centre for Vision, Speech and Signal Processing (CVSSP), at the University of Surrey. In parallel to these experiments, different options to extract torso pose from stereo images were considered. This paper presents a HMC system that uses an EMD based method to estimate human torso pose, and the algorithm detailed in section 3.5.4 to obtain arms pose. Results are quantitatively described using information provided by Codamotion system as ground-truth. As detailed in the paper, torso rotations were successfully detected using this proposed method. However, results regarding torso flexion were more limited. This limitation, the training requirements, and the computational complexity of the EMD algorithm, finally moved research towards searching different solutions.

**Bandera, J.P., Marfil, R., Molina-Tanco, L., Bandera, A., Rodríguez, J.A., and Sandoval, F. (2008). Visual Tracking of Human Activity for a Social Robot Working on Real Indoor Scenarios. *International Journal of Factory Automation, Robotics and Soft Computing*, 3:120-128.**

This paper deeply explains the perceptual component of the prior architecture detailed in section 2.5. This component substituted the perceptual elements used in previous contributions. The description provided in this paper includes an extensive review of the attentional mechanisms that inspired this approach. It is interesting also in this paper the use of a simplified version of the HMC system which dataflow is presented in section .

**Bandera, J.P., Bandera, A., Rodríguez, J.A., Molina-Tanco, L., and Sandoval, F. (2008). Evaluating the performance of a vision-based human motion capture system mounted on a humanoid robot. *Motion Times - Journal for the motion-capture community*, 2(08):1-2.**

This short contribution details the process followed to obtain a quantitative evaluation of the proposed vision-based HMC system. In this thesis this process and the obtained results are described in chapter 3.

**Bandera, J.P., Marfil, R., López, R., del Toro, J.C., Palomino, A., and Sandoval, F. (2008). Retargeting System for a Social Robot Imitation Interface. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2008)*.**

The combined retargeting strategy used in this thesis, described in section 5.3, is deeply

detailed in this paper. Experimental results showed that the method can easily include new retargeting criteria if required. These experiments were also useful to detect certain issues in the design of the arms of the NOMADA, that will be corrected in the definitive implementation.

**Bandera, J.P., Bandera, A., and Sandoval, F. (2009). Human Gesture Recognition based on Adaptive Curvature Functions. In *Workshop on Interfacing the Human and the Robot (IHR). 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*.**

This thesis presents a gesture representation and recognition system based on local and global features. Local features are the subject of this paper. In the proposed system, these local features are stored as a set of dominant points. These sets are compared using DTW to achieve recognition. While this approach provided promising results, it was necessary to compare it against other classification methods in order to validate it. Thus, a local gesture representation that reduces dimensionality of curvature functions using PCA was implemented. The obtained representations were then classified using PCA or LDA. This paper confronts the results obtained by these two approaches. This comparison is included in section 4.9 in this thesis. As detailed there, while PCA-LDA encodes gestures in a more reduced feature vector, DPD-DTW achieves higher recognition rates mainly due to its ability to deal with temporal shifting.

**Cruz, A., Bandera, J.P., and Sandoval, F. (2009). Torso pose estimator for a robot imitation framework. In *Proceedings of the 12th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2009)*.**

This paper deeply explains the silhouette extraction method described in section 3.4, and the global torso pose estimator detailed in section 3.5. The contents of this paper and the arm pose generator, previously contributed, constitute the final upper-body HMC system proposed in this thesis.

**Bandera, J.P., Marfil, R., Bandera, A., Rodríguez, J.A., Molina-Tanco, L., and Sandoval, F. (2009). Fast gesture recognition based on a two-level representation. *Pattern Recognition Letters*, 30(13):1181-1189.**

This paper represents one of the main contributions related to this thesis. It details the gesture representation and recognition systems that are described in chapter 4. However, in this paper the concept of a set of global features that reinforce local ones is replaced by the idea

of representing gestures using both local and global features, that are equally important. The experimental results provided in this paper are presented in chapters 4 and 6 of this thesis. The discussion about these results sets some of the main research lines that will be addressed as future work for this thesis.

## G.2 Publications not covered in this thesis

Results that are only briefly touched upon in this thesis, but which concern related topics.

**Pérez, J.M., Bandera, J.P., Urdiales, C., and Sandoval, F. (2003). Agente autónomo de bajo coste para la exploración de conductos teleoperado mediante realidad virtual. In *I Seminario Nacional Hispabot (HISPABOT'03)*.**

In this early contribution, a wheeled small robot was used to explore pipes and other conducts. The robot was partially teleoperated and used virtual reality to create a model of the explored environment. These models were built from the information captured by sonar sensors. Obtained results suggested the use of a different sensory input for the social robot, due to the problems associated to sonar sensors.

**de Trazegnies, C., Bandera, J.P., Urdiales, C., and Sandoval, F. (2003). A real 3D object recognition algorithm based on virtual training. In *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA 2003)*.**

This paper details an object recognition algorithm based on the use of a database of 3D views of different types of objects. While object recognition has not finally been addressed in this thesis, the method proposed in this paper and the further work of the first author may be used in the future to integrate an object recognizer in the social robot architecture.

**Pérez, J.M., Bandera, J.P., Bandera, A., and Sandoval, F. (2003). Algoritmo de agrupación de segmentos en mapas métricos basado en fusión de clases en el espacio de Hough. In *Unión Científica Internacional de Radio, XVIII Simposium Nacional URSI'2003*.**

In this contribution a method to extract segments from maps obtained using sonar or laser sensors is presented. The method is based on the use of the Hough transform.

**Bandera, J.P., Urdiales, C., and Sandoval, F. (2004). Selective Video Transmission**

by means of Virtual Reality Based Object Extraction. In *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (MELECON 2004)*..

This contribution presents some partial results that are extended in the paper "Video Object Transmission by means of Virtual Reality Based Background Subtraction", detailed below.

**Bandera, J.P., Zhou, C., and Sandoval, F. (2005). Vision Based Walking Parameter Estimation for Biped Locomotion Imitation. *Lecture Notes in Computer Science*:582-589.**

While this thesis deals only with upper-body motion imitation, some efforts have also been done in achieving imitation of leg movements. Leg motion capture presents an important issue: it is difficult to find natural marks. Thus, in this paper color patches are attached to relevant leg locations, i.e. hip, knee and ankle. Then, observed trajectories for these marks are analyzed in order to extract the concrete walking gait that the human is performing. Parameters of this gait are translated to a robot imitator that mimics the observed walking pattern.

**Núñez, P., Bandera, J.P, Pérez-Lorenzo, J.M., and Sandoval, F. (2006). A Human-Robot Interaction System for Navigation Supervision based on Augmented Reality. In *Proceedings of the 13th IEEE Mediterranean Electrotechnical Conference (MELECON 2006)*.**

This paper proposes a human-robot interaction mechanism that permits users to interact intuitively with an autonomous mobile robot which localisation problem is solved using a new fast feature extraction method. To allow that human-robot interaction, an Augmented Reality display is used. These displays may be useful to supervise social robot behaviours, thus their inclusion in the proposed architecture will be addressed as future work.

**Bandera, A., Pérez-Lorenzo, J.M., Bandera, J.P., and Sandoval, F. (2006). Mean shift based clustering of Hough domain for fast line segment detection. *Pattern Recognition Letters*, 27:578-586.**

This paper presents an algorithm to extract line segments from edge images using the Random Window Randomized Hough Transform in a first stage. In a second stage a Variable Bandwidth Mean Shift algorithm is employed to cluster previous items. While this method could have been used to extract silhouette borders (section 3.4), the Canny edge detection method has been finally chosen due to its simplicity and shorter execution times.

**Bandera, J.P., Urdiales, C., Segura, B., and Sandoval, F. (2006). Video Object Transmission by means of Virtual Reality Based Background Subtraction. *Telecommunication Systems Journal*, 32:165-180.**

This paper presents a novel background subtraction technique based on virtual reality. In this method, a virtual model of the background is built to allow camera movements during the transmission of a video sequence. Experimental results show that this approach significantly increases the compression rates obtained when these sequences are processed using MPEG4. In the RLbI scenarios considered in this thesis, however, it is not possible to create these background virtual models due to the dynamic nature of the environment.