UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

TESIS DOCTORAL

TRACKING OBJECTS WITH THE BOUNDED IRREGULAR PYRAMID

AUTOR: Rebeca Marfil Robles
Ingeniera de Telecomunicación

2006

D. JUAN ANTONIO RODRÍGUEZ FERNÁNDEZ, PROFESOR DEL DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA DE LA UNIVERSIDAD DE MÁLAGA

y

D. LUIS MOLINA TANCO, PROFESOR DEL DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA DE LA UNIVERSIDAD DE MÁLAGA

CERTIFICAMOS:

Que Dª. Rebeca Marfil Robles, Ingeniera 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:

"TRACKING OBJECTS WITH THE BOUNDED IRREGULAR PYRAMID"

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, 13 de febrero de 2006

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

TRACKING OBJECTS WITH THE BOUNDED IRREGULAR PYRAMID

AUTOR: Rebeca Marfil Robles

Ingeniera de Telecomunicación

DIRECTORES:

Juan Antonio Rodríguez Fernández
Dr. Ingeniero de Telecomunicación

Luis Molina Tanco
Dr. Ingeniero de Telecomunicación

A mis padres

Abstract

Target representation and localization is a central component in visual object tracking. In this Thesis, a tracking algorithm based on a novel approach for target representation and localization is presented. The goal is to track, in real time, rigid and non-rigid objects in cluttered environments, under severe changes of viewpoint and deformations and in the absence of an a priori model. To achieve this goal, a novel template-based appearance model of the tracked object is proposed. This appearance model uses a new pyramidal estructure, the Bounded Irregular Pyramid, to represent the target and the template as well as to perform the template matching process in a hierarchical way. This allows to reduce the computational cost associated with the template matching procedure.

The Bounded Irregular Pyramid (BIP) is a mixture of regular and irregular pyramids whose goal is to combine their advantages: low computational cost and accurate results. The key idea is to use a regular approach in the homogeneous regions of the input image and an irregular approach in the rest of regions. The BIP's data structure is a combination of a 2x2/4 regular structure with a simple graph. Thus, while in the regular part of the BIP a regular decimation process is used, in the irregular part a union-find decimation approach is employed. The irregular part of the BIP allows to solve the three main problems of regular structures: non-connectivity preserving, non-adaptability to the image layout and shift-variance. On the other hand, the BIP is computationally efficient because its regular part prevents a big increase of height. The use of the BIP as target representation tool allows to perform the tracking in real-time as it can be rapidly built and traverse. At the same time, it has demonstrated to represent the target and the template accurately.

The proposed tracking approach allows to track rigid and non-rigid objects by employing a weighted template which is dynamically updated. This template includes information of previous templates, addressing two of the most important causes of failure in object tracking: changes of object appearance and occlusions. In addition, the proposed hierarchical tracker allows tracking of multiple objects with low increase of computational time.

The previously mentioned characteristics of the proposed algorithm makes it very suitable for more complex visual applications which require real time response. This algorithm has been included in two applications: a human motion capture system and an attentional mechanism. In the first application the tracking algorithm is used to follow the movements of the hands and the head of the human whose movements are being captured. In the attentional mechanism, the proposed tracking approach is used to track the movements of the salient objects presented in the scene. This allows implementing an inhibition of return mechanism for moving objects.

Resumen

En procesos de seguimiento de objetos usando visión artificial, tanto la forma de representar y modelar el objeto a seguir u objetivo (*target*) como el proceso de localización de dicho objeto en cada fotograma de la secuencia son procesos centrales. En la literatura, ambos procesos suelen agruparse en uno solo, denominado Representación y localización del objetivo. En esta Tesis se propone un sistema de seguimiento de objetos basado en un nuevo método de Representación y localización del objetivo. Se trata de realizar el seguimiento de objetos no rígidos en tiempo real, sin utilizar ningún modelo previo de los objetos a seguir. Para conseguir esto, se propone un nuevo modelo para caracterizar la apariencia del objeto basado en una máscara o *template*. Este modelo utiliza una nueva estructura piramidal, denominada *Bounded Irregular Pyramid* (BIP), para representar el target y el template, así como para realizar el proceso de localización del objeto o *template matching* de forma jerárquica, reduciendo su coste computacional.

La BIP es una combinación de una estructura regular y una irregular, cuyo objetivo es aprovechar las ventajas de ambas: bajo coste computacional y resultados precisos. La idea principal de la BIP es utilizar un enfoque regular en las zonas homogéneas de la imagen y un enfoque irregular en el resto de regiones. La estructura de la BIP es una combinación de una estructura regular 2x2/4 con un grafo simple. Así, en la parte regular de la BIP se utiliza un diezmado regular y en su parte irregular se utiliza un proceso de diezmado denominado *union-find*. La parte irregular de la BIP permite solucionar los tres problemas principales de las estructuras regulares: no conectividad de las regiones resultantes, no adaptabilidad a la estructura de la imagen de entrada y obtención de diferentes resultados para pequeños desplazamientos de la imagen (*shift-variance*). Por otro lado, la BIP es computacionalmente eficiente porque su parte regular evita que la estructura crezca demasiado. Por ello, su uso permite llevar a cabo el proceso de seguimiento en tiempo real, ya que esta estructura se construye y se recorre muy rápidamente. Además, lo resultados obtenidos con esta estructura son muy precisos.

El sistema de seguimiento propuesto permite realizar el seguimiento de objetos rígidos y no rígidos utilizando una máscara que se actualiza de forma dinámica. Esta máscara incluye información de las máscaras previas, solucionando dos de los causas de fallo más importantes de los sistemas de seguimiento: cambios en la apariencia del objetivo y oclusiones del mismo. Además, el sistema permite seguir varios objetos simultáneamente sin un incremento excesivo del coste computacional.

Las características previamente comentadas del sistema propuesto lo hacen muy adecuado para su utilización en aplicaciones visuales más complejas, que requieren una respuesta en tiempo real. Este algoritmo ha sido incluido en dos aplicaciones de este tipo: un sistema de captura del movimiento humano y un mecanismo atencional. En la primera de estas aplicaciones, el sistema de seguimiento propuesto es utilizado para seguir los movimientos de las manos y la cabeza de la persona. En el mecanismo atencional, el sistema de seguimiento es usado para seguir los movimientos de los objetos relevantes de la escena, implementando un mecanismo dinámico de inhibición de retorno.

Contents

Acronyms

AAM	Active Appearance Model.
APF	Auxiliary Particle Filter.
ASIR	Auxiliary Sampling Importance Resampling filter.
BIP	Bounded Irregular Pyramid.
ClIP	Classical RAG hierarchy proposed by Bertolino and Montanvert [10].
CMM	Combined Motion Model.
CoIP	Segmentation based on combinatorial pyramids and union-find algorithm.
D3P	Data driven decimation process.
EKF	Extended Kalman Filter.
EM	Expectation Maximization algorithm.
GMM	Global Motion Model.
HIP	Hierarchy of image partitions by internal and external contrast measures.
HSI	Hue Saturation Intensity colour space.
HSV	Hue-Saturation-Value colour space.
IMM	Individual Motion Model.

JPDAF	Joint Probabilistic Data Association Filter.
KF	Kalman Filter.
KLT	Kanade-Lucas-Tomasi algorithm.
LIP	Localized Irregular Pyramid.
LRP	Linked Regular Pyramid proposed by Burt *et al.* [24].
MCG	Mean Colour Gradient.
MHF	Multiple Hypothesis Filter.
MHT	Multiple Hypothesis Tracking algorithm.
MIDES	Maximal Independent Directed Edge Set algorithm.
MIES	Maximal Independent Edge Set algorithm.
MIG	Mean Intensity Gradient.
MIP	Adaptive Irregular Pyramid with Moran's test proposed by Lallich *et al.* [86].
MIS	Maximal Independent Set.
MMF	Multiple Model Filtering.
MPLA	Modified Pyramid Linking Approach.
MRMTPF	Multireceiver Multitarget Particle Filter.
MST	Minimum weight Spanning Tree.
MTPF	Multitarget Particle Filter.

PCA	Principal Component Analysis.
PDAF	Probabilistic Data Association Filter.
PLA	Pyramid Linking Approach.
RAG	Region Adjacency Graph.
RGB	Red Green Blue colour space.
RMSD	Root Mean Square Difference.
ROI	Region Of Interest.
RPF	Regularised Particle Filter.
SIS	Sequential Importance Sampling algorithm.
SIR	Sampling Importance Resampling filter.
SSDA	Sequential Similarity Detection Algorithm.
SV	Shift Variance.
TSL	Tint Saturation Luminance colour space.
UKF	Unscented Kalman Filter.
WRP	Weighted linked pyramid with possibilistic linking.

List of Figures

List of Tables

Chapter 1

Introduction

Vision is the most important of the five human senses, since it provides over 90% of the information our brain receives from the external world. Its main goal is to interpret and to interact with the environments we are living in. In everyday life, humans are capable of perceiving thousands of objects, identifying hundreds of faces, recognizing numerous traffic signs, or appreciating beauty almost effortlessly. The ease with which humans achieve these tasks is in no way due to the simplicity of the tasks but is a proof of the high degree of development of our vision system.

Computer vision is an applied science whose allow computers to extract relevant specific information from the input image.

Typical goals of computer vision include:

- The detection, segmentation, location, and recognition of objects in images (e.g., human faces).
- The search for digital images by their contents (content-base image retrieval).
- The help in robot localization and navigation (e.g. visual landmark adquisition, building a 3D model of the scene).
- The tracking of objects in image sequences.
- The estimation of the three-dimensional poses of humans and their limbs.

In computer and human vision, tracking means maintaining correspondence of a representation of the projected object in the images, over multiple frames. The projection of the object in the images is called target. The goal of the tracking process is to recognize or estimate the motion

and the position of the tracked object. The results of the tracking have a variety of applications [45]:

- Motion capture: if the movement of a person can be tracked, then it can be used in tasks like, for example, cartoon animation, human-robot interaction, virtual avatar animation, perceptual user interfaces, smart rooms, etc. Besides, the movement can be modified to obtain slightly different motions.

- Recognition from motion: The motion of objects is quite characteristic. In some cases, it is possible to determine the identity of an object, and what it is doing, from its motion.

- Surveillance: In traffic surveillance, for example, is very useful to follow the movement of the different vehicles in order to give a warning if a problem is detected.

1.1 Motivation

When an object moves relative to an observer, the projected images of the object on the retina or in the camera change. Not just the position but also the appearance of the tracked target is likely to change over time for a number of reasons. Changing lighting conditions, the 3D structure of the object combined with the relative motion of the object with respect to the observer, camera noise and various occlusions will all cause changes in appearance. This makes the object tracking a challenging task in which the computer vision community has been putting effort since the seventies.

In a typical visual tracker, two major components can be distinguished [34]:

- *Target Representation and Localization*: it copes with the appearance changes of the tracked object. Specifically, target representation is the way that the information about the desired tracked object is manipulated and stored. Localization is the process applied to search the object in each image. Target representation determines the process to localize the target.

- *Filtering and Data Association*: filtering deals with the dynamics of the tracked object, estimating the present and the future of the target kinematics quantities such as position, velocity, and acceleration. Data association techniques try to solve the problem of measurement association when there are several objects to track.

The way the two components are combined and weighted depends on the application and plays a decisive role in the robustness and efficiency of the tracker. For example, tracking a face in a crowded scene relies more on target representation than on target dynamics [38], while in aerial video surveillance, e.g., [162], the target motion and the ego-motion of the camera are the more important components. In this Thesis the emphasis is put in the Target Representation and Localization as the responsable of dealing with the most important causes of failure in object tracking: change of object appearance and occlusions. A good selection of the target representation has important consequences in the behaviour of the whole system. The reasons for this are manifold:

- The chosen target representation determines the target localization procedure, for instance, the similarity measurements employed in the target searching procedure.
- The target representation encodes object information resulting in a data reduction. Therefore, the target representation determines which object information is relevant, i.e. which information is encoded, and which is not.
- Some desirable invariance properties with respect to perceived object sizes, deformations, occlusions and illumination, are directly related with the model used to represent the target.
- A further aspect is the efficiency of the representation. Low reaction time is of vital importance to many systems. However, it depends on the data that needs to be evaluated, so depends on the target representation.

1.2 Goals

This Thesis is concerned with tracking rigid and non-rigid objects in cluttered image sequences. The aim is to develop a target representation approach which can perform, in real time, robust object tracking under severe changes of viewpoint and deformations in the absence of a priori model. This approach is based on a novel hierarchical template scheme.

The classical idea behind template tracking is that an object is tracked through a video sequence by extracting an example image of the object in the first frame -a template- and then finding the region which matches the template as closely as possible in the remaining frames. The underlying assumption is that the appearance of the object remains the same throughout the entire video. This assumption is generally reasonable for rigid objects during a certain period

of time, but breaks in the case of non-rigid objects which modify their appearance with time. A naive solution to this problem is to update the template every frame (or every n frames) with a new template extracted from the current image at the current location of the template. The problem with this approach is occlusions. What happens if the template is updated in a frame where the object is occluded?. This work will address the two main drawbacks of classical template matching approaches to tracking, namely:

- mismatches between template and object appearance,
- partial and total occlusions of the object.

To do that, the tracker should: i) update the template to accommodate the changed object appearance and, ii) detect the occlusion and recapture the object when the occlusion ends. In order to acquire a template that can satisfy these conditions, the entire sequence up to the current frame must be used. For example, the template could be computed as a weighted average between the previous template and the current localized target [140].

The main goal of this Thesis can be resumed as the development of a novel template-based target representation scheme which is robust and, at the same time, has low computational cost. Robustness implies the ability of the algorithm to track objects under difficult conditions which include:

- severe occlusions and lighting changes,
- changing of object orientation or viewpoint,
- deformations of non-rigid objects,
- background clutter and the presence of other moving objects in the scene,
- a moving camera, and
- non-translational object motion like zooms and rotations.

In order to achieve low computational cost, a pyramid is used to represent both the template and the scene image and to perform the matching method in a hierarchical way. In this Thesis the most important types of pyramidal structures have been studied in order to select the most adequate for the proposed tracking system. Mainly, there are two kinds of pyramids: regular pyramids and irregular ones. After comparing their performance, it is possible to claim

that the studied pyramids are inadequate for the purposes of this Thesis: the regular structures due to their rigidity and the irregular ones because of their high computational complexity. To solve these problems, a new pyramidal structure is proposed in this Thesis: the Bounded Irregular Pyramid (BIP). The goal of the BIP is to achieve a more computational efficient framework for target representation as well as a hierarchical support for the tracking process. It is a mixture of both regular and irregular pyramids whose goal is to combine their advantages: low computational cost and accurate segmentation results.

1.3 Main contributions

The main contributions of this Thesis are summarized as follows:

- The implementation and detailed analysis of a new pyramidal structure for image processing: the Bounded Irregular Pyramid (BIP). The key idea of this pyramid is to combine the advantages of regular and irregular pyramids within the same structure. To do that regular and irregular data structures as well as regular and irregular decimation processes are mixed in a novel way to build the BIP. This pyramid allows to process images ten times quicker than the existing irregular pyramids with similar accuracy. This reduction of the computational time makes it possible to use the BIP in real-time applications such as the tracking algorithm proposed in this Thesis.

- The development of a new template-based target representation scheme using the Bounded Irregular Pyramid. This template combines colour and spatial information. The way in which this template is updated allows to include information of previous templates in order to avoid tracking errors due to appearance changes of the object or occlusions.

- The implementation of a tracking algorithm based on template matching. This algorithm takes advantage of the hierarchical structure of the template representation to perform the template matching in a hierarchical way. This approach makes possible to simultaneously track several objects without a high increase of the computational cost.

- The experimental validation of the accuracy of the proposed tracking system in several situations as:

  - partial and total occlusions of the tracked object,
  - illumination changes,
  - appearance changes of the object due to deformations, zooms or rotations,

  - moving camera,
  - the presence of other moving objects in the scene, and
  - the tracking of several objects at the same time.

- The study of the behavior of the tracking system in real time applications as human motion capture and an attentional mechanism. The proposed tracking approach has been included in both previously mentioned applications, demonstrating its suitability to work in more complex systems which require a fast response.

1.4 List of publications

Early versions and partial results of this work appear in several publications. Specifically, previous work related to pyramidal structures appears in:

- R. Marfil, C. Urdiales, J. A. Rodríguez and F. Sandoval, Automatic Vergence Control Based on Hierarchical Segmentation of Stereo Pairs, International Journal of Imaging Systems and Technology, 13(4), 224-233, 2003.
- R. Marfil, A. Bandera and F. Sandoval, Colour image segmentation based on irregular pyramids, IASTED conference on Visualization, Imaging and Image Processing (VIIP 2003), Benalmádena (Spain), September 2003.
- R. Marfil, A. Bandera, J. A. Rodriguez and F. Sandoval, Region based stereo matching through bounded irregular pyramids, International Workshop in Colour Science on Computer Vision and Image Processing, London (UK), November 2003.
- R. Marfil, F. Jiménez, A. Bandera and F. Sandoval, Análisis de imagen basado en textura mediante transformada wavelet, XIX Simposium Nacional de la Unión Científica Internacional de Radio URSI'2004, Barcelona (Spain), September 2004.
- R. Marfil, J.A. Rodriguez, A. Bandera and F. Sandoval, Bounded irregular pyramid: a new structure for colour image segmentation. Pattern Recognition, 37(3), pp. 623-626, 2004.
- R. Marfil, L. Molina-Tanco, A. Bandera, J.A. Rodríguez and F. Sandoval, Pyramid segmentation algorithms revisited, accepted to Pattern Recognition.

Previous work related to the proposed tracking approach appears in:

- R. Marfil, A. Bandera, J. A. Rodríguez and F. Sandoval, Real-time Template-based Tracking of Non-rigid Objects using Bounded Irregular Pyramids, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1, pp. 301-306, Sendai (Japan), September 2004.

- R. Marfil, L. Molina-Tanco, J.A. Rodríguez and F. Sandoval, Real-Time Object tracking using Bounded Irregular Pyramids, in second revision process in Pattern Recognition Letters with minor revisions.

Applications of the proposed tracking algorithm have been published in:

- J.P. Bandera, L. Molina-Tanco, R. Marfil and F. Sandoval, A Model-based Humanoid Perception System for Real-time Human Motion Imitation, Proc. of the IEEE Conference on Robotics, Automation and Mechatronics, pp. 324-329, Singapore (Singapore), December 2004.

- J.P. Bandera, R. Marfil, L. Molina-Tanco, A. Bandera y F. Sandoval, Model-based Pose Estimator for Real-time Human-Robot Interaction, aceptado en: Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005), Singapore (Singapore), December 2005.

- L. Molina-Tanco, J.P. Bandera, R. Marfil and F. Sandoval, Real-time Human Motion Analisys for Human-Robot Interaction, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1808-1813, Alberta (Canada), August 2005.

- R. Marfil, R. Vázquez-Martín, L. Molina-Tanco, A. Bandera and F. Sandoval, Fast attentional mechanism for a social robot, European Robotic Symposium (EUROS-06) (Workshop on Vision Based Human-Robot Interaction), Palermo (Italy), March 2006.

1.5 Thesis outline

This Thesis is divided in five main chapters. The first chapter makes a review of the literature in object tracking and piramidal structures. The second and third chapters explain the proposed Bounded Irregular Pyramid and the tracking process, respectively. The fourth chapter studies the use of the proposed tracking approach in two real applications: human motion capture and attentional control. The final chapter makes a brief summary of the main conclusions extracted from the previous chapters. This Thesis also includes two appendices which explain the

HSV colour model employed to build the BIP, and the main piramidal segmentation algorithms present in the literature.

The content of each chapter is briefly summarized below:

- Chapter 2: Literature review.

  The main types of filtering and data association methods as well as the main target representation approaches, are briefly described in the first and second sections of this chapter. The third section is dedicated to detailed explain the main regular and irregular pyramids. This revision is detailed in order to make easier to understand the pyramidal structure presented in Chapter 3 of this Thesis and its advantages.

- Chapter 3: Bounded Irregular Pyramid.

  In this chapter the Bounded Irregular Pyramid is presented, analyzing its regular and irregular data structures and decimation processes. Besides, it is compared with the main pyramidal approaches previously described in Chapter 2, pointing out its better suitability for the proposed tracking system.

- Chapter 4: Tracking algorithm.

  This chapter presents the different modules of the proposed tracking algorithm and its application to track a single object and multiple objects. A study of the behaviour of the tracking in different situations is also presented, demonstrating the accuracy of the proposed method. Besides, an analysis of the different parameters of the algorithm is shown.

- Chapter 5: Applications.

  The tracking proposed in this Thesis has been used in two real applications, which are explained in this chapter: a human motion capture application and an attentional mechanism. The results of these applications and the advantages of use the proposed tracking are showed.

- Chapter 6: Conclusions and future work.

  This chapter summarizes the main conclusions extracted from the development of the different parts of this Thesis. It also includes several improvements that can be performed over the proposed tracking approach.

Chapter 2

Literature review

Two major components can be distinguished in a typical visual tracker: i) Target Representation and Localization and ii) Filtering and Data Association. Target Representation and Localization is mostly a bottom-up process, which must be capable of dealing with changes in appearance and partial occlusions of the target, while Filtering and Data Association is usually a top-down process dealing with the dynamics of the objects and the evaluation of different assumptions. Therefore, Target Representation corresponds with the way that the information about the desired tracked object is manipulated and stored. The used Target Representation approach determines the process to localize the target. Filtering is the process to predict the position of the tracked object in the current frame taking into account the past behaviours of the object and the system.

This chapter is organized according to this subdivision. The first section is dedicated to the main contributions in Filtering and Data Association that, although is not the focus of this Thesis, is revised to give a whole vision of tracking systems. The second section reviews the Target Representation and Localization techniques. Finally, a third section makes a detailed revision of the main regular and irregular pyramids. This section has been included in order to make easier to understand the pyramidal structure proposed in Chapter 3 of this Thesis and its better suitability for the proposed tracking approach.

2.1 Filtering and Data Association

A filter is a procedure that looks at a collection or stream of data taken from a system. The system is described by a state equation. The filter estimates parameters or system state variables. System parameters are usually taken to be time-invariant or slowly-varying properties of the

system. In the case of object tracking, the state variables could be, for instance, the position, velocity and acceleration of the tracked object. The state equation might be that of a point moving under constant acceleration. The available set of data is a stream of noisy position measurements.

The most abstract formulation of the filtering and data association process is through the *state space approach* for modeling discrete-time dynamic systems [7]:

> "The information characterizing the target is defined by the state variables $\{x_k\}_{k=0,1,\ldots}$, whose evolution in time is specified by the state dynamic equation $x_k = f(x_{k-1}, v_k)$. The available measurements $\{z_k\}_{k=1,\ldots}$ are related to the corresponding states through the measurement equation $z_k = h(x_k, n_k)$. In general, both f_k and h_k are vector-valued, nonlinear, and time-varying functions. Each of the noise sequences, $\{v_k\}_{k=1,\ldots}$ and $\{n_k\}_{k=1,\ldots}$ is assumed to be independent and identically distributed (i.i.d.). The objective of tracking is to estimate the state x_k given all the measurements $z_{1:k}$ up that moment, or equivalently to construct the probability density function (*pdf*) $p(x_k|z_{1:k})$. The theoretically optimal solution is provided by the recursive Bayesian filter which solves the problem in two steps. The prediction step uses the dynamic equation and the already computed *pdf* of the state at time $t = k-1$, $p(x_{k-1}|z_{1:k-1})$, to derive the prior *pdf* of the current state, $p(x_k|z_{1:k-1})$. Then, the update step employs the likelihood function $p(z_k|x_k)$ of the current measurement to compute the posterior *pdf* $p(x_k|z_{1:k})$".

Depending on the characteristics of the tracking system the optimal solution to the previous problem is provided by different techniques. When the noise sequences are Gaussians and f_k and h_k are linear functions, the optimal solution is provided by the Kalman Filter (KF) [7]. Boykov and Huttenloncher [16] have used this filter combined with a Bayesian Recognition technique to track rigid objects in an adaptive framework. Girondel *et al.* [49] track multiple people in real time. To do that, they use a Kalman filter for each person to predict the bounding boxes and velocity for the person and his face. They use a partial Kalman filter in the case of incomplete measurements (for instance, when a partial occlusion occurs). A simplification of the Kalman filter is the $\alpha - \beta$ or $\alpha - \beta - \gamma$ filter [7]. This filter is a time-invariant filter in which system variations through time are accommodated by modeling them as noise. It can be the most realistic assumption if the variations are unknown. The $\alpha - \beta$ filter used in tracking assumes a constant velocity model, while the accelerations are modeled as process noise. Besides, it assumes that only position measurements are available. The $\alpha - \beta - \gamma$ filter is the extension

to the $\alpha - \beta$ filter. It includes an estimate for the acceleration and it can be used with the assumption of uniform acceleration.

When f_k and h_k are not linear there are two possibilities: the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) [7]. In the case of tracking objects in images, the measurement model is often nonlinear due to clutter in images. Traditional visual trackers based on Kalman filters that employ simple linear measurement models often collapse during the tracking process. The EKF is obtained by applying KF techniques locally to the data by linearizing f_k and h_k around the current estimation. The UKF was developed by Julier and Uhlmann [73] as an alternative to the EKF, because EKF is only reliable for systems which are almost linear on the time scale of the update intervals. This filter is based in the unscented transformation which uses a set of discretely sampled points to parameterize the means and covariances of probability distributions. The UKF is easier to implement than the EKF and it predicts the state of the system more accurately because it uses a second order approximation versus the first order of the EKF. Masoud and Papanikolopoulos [95] track pedestrian in two steps: blob tracking and pedestrian tracking. The blobs are extracted using background subtraction and they are tracking regardless of what they represent. In the pedestrian tracking step the blobs are associated to pedestrians which are modeled like rectangular patches with movement. This movement is assumed to have constant velocity. Each pedestrian has a EKF to track its parameters (position and velocity). Gao *et al.* [46] use a multi-Kalman filtering approach to track objects. A set of features are extracted from the object to track which is represented by a movement vector and a shape vector. The features are grouped taking into account the frame in which they appears the first time. An EKF is used to update the uncertainty of the motion vector of each set of features. Chen *et al.* [30] use an Unscented Kalman Filter to track contours. In a first step they model the contour as a parametric shape (i.e., an ellipse) and they use a Hidden Markov Model to detect it. In a second step an UKF is used to estimate the shape parameters (i.e, the center, the length of the minor and major axes and the orientation of the ellipse). Li *et al.* [88] use the Unscented Kalman Filter to track contours which are modeled as B-splines and they show that this filter has a better performance than the classical KF.

All the types of Kalman filters described above are based on the assumption that all distributions remain Gaussian. In practice, for many linear or linearizable systems this assumption is often reasonable as far as the noise in the system dynamics is concerned. The main problem is the data likelihood function which can easily be non-Gaussian and multimodal in cluttered scenes [16]. To deal with non-Gaussian, multimodal and non-linear functions, a Particle Filter can be used, which is based on Monte Carlo integration methods. The basis of most particle

filters that have been developed is the Sequential Importance Sampling Algorithm (SIS). The various versions of particle filters proposed in the literature can be regarded as special cases of this general SIS algorithm. The key idea is to represent the required posterior density function by a set of random samples with associated weights (*particles*) and to compute estimates based on these samples and weights. These samples are chosen from a function named *importance density function*. A problem with the SIS is the degeneracy problem, where after a few iterations, all but one particle will have negligible weight. This problem is avoided in two ways: using an adequate importance density and using a resampling process. The selection of the importance density function is the most critical step in the design of a particle filter for a particular application. Depending on the chosen importance density and/ or in the resampling algorithm, new particle filters can be derived from the SIS. Some of them are the Sampling Importance Resampling filter (SIR), the Auxiliary Sampling Importance Resampling filter (ASIR), the Likelihood Particle filter and the Regularised Particle filter (RPF) [5]. The particle filter was introduced to the computer vision comunity by Isard and Blake as the CONDENSATION Algorithm [69]. This condensation algorithm was developed to track curves in visual clutter. Bruno [20] uses particle filters to simultaneously include in the tracker the statistical models for the background clutter, target motion and target aspect change. Specifically, he proposes two particle filters which are based, respectively, on the SIR filter and on the alternative auxiliary particle filter (APF). Kang and Kim [75] proposed a new competitive CONDENSATION algorithm to achieve robust and real-time tracking of near or partially occluded multiple people.

The above mentioned methods are all estimation methods, that is, given a sequence of images containing the object that is going to be represented with a parametric model, an estimator is a procedure for finding the parameters of the model which best fit the data. In order to do that, the object must be discriminated from the rest of the image. Under real world conditions, it can be difficult to accurately identify an object's image projection because visual phenomena such as agile motion, distractions, and occlusions interfere with estimation [116]. In these situations it is possible to have no measurements or multiple measurements of the object due to noise. These problems can be tackled using *data association techniques*. The simplest approach is the Nearest Neighbour [7] which selects the closest measurement to what is expected in order to update the state. Another technique is the Probabilistic Data Association Filter (PDAF) [7] which can be applied when only one object is being tracked. It is an extension of the Kalman filter that uses a Bayesian approach to update the state when there is a single target and possibly no measurements or multiple measurements due to noise. Although the previous approach is only for tracking a single object, there are data association approaches to deal with the problem of the association of measurements when there are several targets to track.

In that case, the association must be done by considering all the targets simultaneously. One of these techniques is an extension to the PDAF called the Joint Probabilistic Data Association Filter (JPDAF) [7]. It enforces a kind of exclusion principle that prevents two or more trackers from latching onto the same target by calculating target-measurement association probabilities jointly. The JPDAF is only appropriate if the number of tracks is known a priori and remains fixed throughout the motion sequence [36]. A different strategy is represented by the Multiple Hypothesis Filter (MHF) [7]. While the JPDAF is a target-oriented approach, that is, the probability that each measurement belongs to an established target is evaluated, the MHF approach is measurement oriented in the sense that the probability that each established target or a new target gave rise to a certain measurement sequence is obtained. The JPDAF filter was initially proposed by Bar-Shalom [8]. Zhou and Bose [167] presented a revision of the origins and problems of the JPDAF approach and they proposed three different alternatives to approximate it. Gennari *et al.* [48] proposed a new derivation of the JPDAF based on the theory of evidence which permits to include new information (i.e. shape constrains). The multiple hypothesis tracking algorithm was originally developed by Reid [118] in the context of multi-target tracking. This classical MHT technique by itself is computationally exponential both in time and memory but it has been efficiently implemented by Cox and Hingorani [36]. Tissainayagam and Suter [144] use the Cox and Hingorani's approximation to the MHF filter to develop a novel technique of efficiently and reliably tracking corner features in a sequence of images. This method couples the MHT technique with a multiple model Filtering (MMF) algorithm. In [146] they extend the corner feature tracking to object tracking.

Unfortunately, the above mentioned algorithms of data association in multi-target tracking do not cope with nonlinear models and non-Gaussian noises [25]. Under such assumptions (stochastic state equation and nonlinear state or measurement equation and non-Gaussian noises), an adaptation of the particle filters to track multiple objects is an appropriate solution. Hue and Le Cadre [25] made a revision of the origins of multi-target tracking using particle filters and they present two major extensions of the classical particle filter in order to deal first with multiple targets (MTPF) and with multiple receivers (MRMTPF). A recent work to track multiple objects using basic particle filter in conjunction with the JPDAF algorithm was presented by Arj and Vahdati-khajeh [4].

2.2 Target representation and Localization

If Filtering and Data Association techniques deal with the dynamics of the tracked objects, Target Representation and Localization approaches cope with their appearance changes. Depending on their target representation, object tracking methods can be classified into five groups [27]: model-based, appearance-based, contour- and mesh-based, feature-based and hybrid methods.

Model-based tracking methods exploit the a priori knowledge of the shape of typical objects in a given scene. The definition of parameterized object models makes it possible to solve the problem of tracking partially occluded objects. This approach has three main drawbacks: i) it is computationally expensive, ii) it needs an object model with detailed geometry for each object that could be found in the scene, and iii) it is not possible to generalize to any object. This last drawback prevents the system from detecting objects that are not in the database.

Appearance-based methods track connected regions that roughly correspond to the 2D shapes of video objects based on their dynamic model. The tracking strategy relies on information provided by the entire region. Examples of such information are motion, colour and texture. These methods cannot usually cope with complex deformations of the tracked object.

Instead of tracking the whole set of pixels comprising an object, contour-based methods track only the contour of the object. Usually the contour-based methods use active contour models like snakes, B-splines or geodesic active contours. 2D meshes are a target representation which allows to simultaneously represent motion and shape. They are based on the assumption that the initial appearance of the object can be specified, and the object motion can be modeled by a piecewise affine transformation.

The fourth group of tracking methods uses features of an object to track parts of the object. The key idea of feature-based tracking is that computational complexity is reduced when tracking prominent features of the object, instead of the whole region of the object or its contours. Besides, tracking parts of objects results in stable tracks for the features under analysis even in case of partial occlusion of the object. The problem of grouping the features to determine which of them belong to the same object is its current major drawback.

The last group of tracking approaches is designed as a hybrid between an appearance-based and a feature-based technique. They exploit the advantages of the two approaches by considering first the object as an entity and then by tracking its parts. The main drawback of these approaches is their high computational complexity.

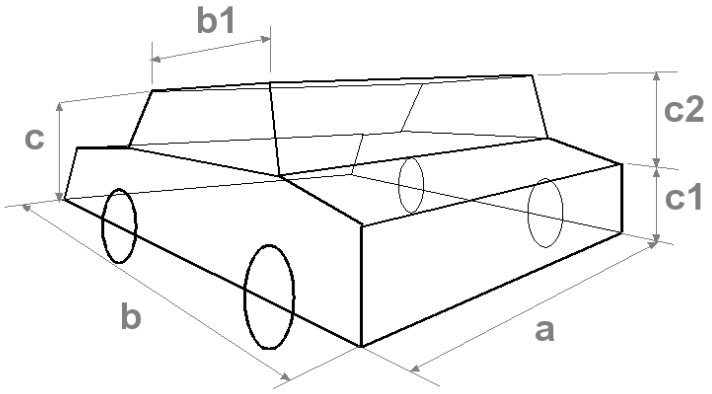

Figure 2.1: Wire-frame model of a car.

In the remaining of this section the most relevant work on Target Representation is reviewed.

2.2.1 Model-based target representation

Model-based tracking approaches employ the a priori knowledge of object shapes in a given scene. Depending on the nature of the object to track, the most widely used model-based representations can be roughly divided in rigid object models and articulated object models. Rigid object models are used to track rigid objects, while articulated models are capable to track more complex objects as hands or human bodies.

2.2.1.1 Rigid object models

The most widely used model to represent rigid objects is the wire-frame model. A wire-frame model is a model that only contains vertex and edge information (see Fig. 2.1). Using this model, object tracking can be performed by tracking object transformations in 3D pose space. That means there is a geometric transformation mapping model features onto their corresponding ones in the image. This transformation minimizes some error model.

A commonly used approach for this kind of methods uses numerous points of the wire-frame model of the target as model features. Algorithms in this approach minimize the squared sum of distances from the projection of these points to the matching scene features. Examples of this approach are the work of Koller *et al*. [78] to track moving vehicles, and the work of Martin and Horaud [93] for tracking rigid objects using multiple cameras. The main drawback of these

algorithms is that they work only when the model features are close to the true matching scene features, which is the case when the initial pose is fairly close to the true pose and the motion of the target is very smooth. To solve this problem, voting-based schemes for estimating the pose parameters have been proposed [138]. The voting based algorithms are more robust than the previous algorithms, but they are time-consuming and not suitable for real-time applications. Other approaches use appearance image databases of the target combined with a 3D wire-frame target model. An example of this approach is the work of Vacchetti *et al.* [154]. In this approach, the image database is acquired in a learning phase. During testing, for a given scene the target pose is first roughly estimated by registering the input scene to the closest image in the appearance image database. Then, the target pose is refined further by projecting the model features into the image plane using the rough pose.

The previously mentioned methods use numerous features. An alternative is to use a wire-frame model that consists of a small number of features and search and select a matching scene feature for each model feature individually. By reducing the number of feature matchings, it becomes feasible to assess the validity of an individual match and use backtracking if the current matching leads to a large error. The work of Yoon *et al.* [165] follows this approach.

The previous approaches only used edges and vertex information. But the wire-frame model can be extended by including texture information. In the work of Vacchetti *et al.* [153] the texture points are handled as the rest of interest points of the model.

2.2.1.2 Articulated models

Among the first to address the problem of tracking articulated objects in a sequence of images, O'Rourke and Badler [107] used a realistic 3D model of a person made of about 600 overlapping spheres and 25 joints. They added constraints on acceleration limits, on distances, on joint angle limits, and on collision avoidance. They defined the tracking process as a loop with four steps: synthesis of the model in images, analysis of images (low level), estimation of pose of the model (parsing), and prediction of the next pose (high level). Each step uses the 3D model.

Rohr [120] proposed a 3D model of a pedestrian made of cylinders, and with only four parameters: three for the general pose and one that indexes the current state of the walk. For both initialization and tracking, the differences between the projection of the 3D model and segments extracted from images are minimized.

Rehg and Kanade [117] built a 3D articulated model of a hand with truncated cones.

They predict occlusions between its rigid parts to increase the robustness of the tracking. They minimize the difference between each image and the appearance (layers of templates) of the 3D model.

Kakadiaris and Metaxas [74] proposed a method for automatically building the 3D model of an articulated object from several cameras. They create physical forces between the model and the images to not only update the shape of the 3D model but also to track it along the sequence.

Gavrila and Davis [47] used a 3D articulated model of a human with 17 degrees of freedom made of tapered super-quadrics estimated from images. Super-quadrics include such diverse shapes as cylinders, spheres, ellipsoids and hyper-rectangles. The tracking algorithm is performed with a generate and test strategy in a discretized and hierarchical state space by minimizing a Chamfer distance between the 3D model and four different views.

Bregler and Malik [17] projected orthographically in images their 3D articulated model. Each rigid part projection of the model is an ellipse with a support map that memorizes the probability that each point of the ellipse is a point of the filmed person. Initialization is manual and tracking is modeled by twists and products of exponential maps.

Wachter and Nagel [160] proposed a 3D human model with 28 degrees of freedom made of truncated elliptic cones. The initialization is manual and the tracking consists of detecting contours (maxima of image gradient) and detecting moving regions between images (with optical flow).

Delamarre *et al.* [39] proposed a human model with 22 degrees of freedom made of truncated cones (arms and legs), spheres (neck, joints and head), and right parallelepipeds (hands, feet and body). They also proposed a 3D model of a hand similar to the human model but with 27 degrees of freedom. They created physical forces between the projections of the model and the contours of the silhouettes of the human body or a 3D reconstruction of the hand, and solve the dynamical equations of motion with a fast recursive algorithm.

Mikic *et al.* [99] used a human body model formed by ellipsoids and cylinders which is described using the twist framework. They presented a fully automated system for human body model acquisition and tracking using multiple cameras. The system does not perform the tracking directly on the image data, but on the 3D voxel reconstructions computed from the 2D foreground silhouettes. This approach removes all the computations related to the transition between the image planes and the 3D space from the tracking and model acquisition algorithms.

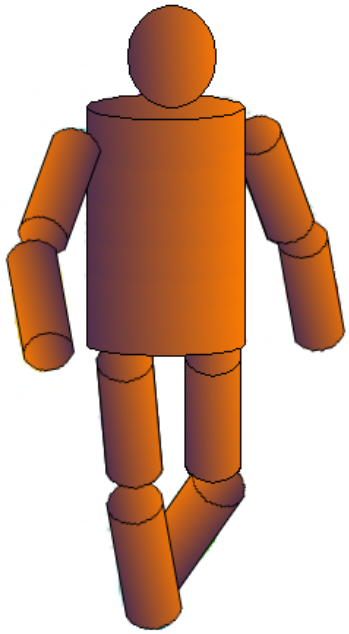

Figure 2.2: Articulated human model.

An example of articulated model made with cylinders is shown in Fig. 2.2.

2.2.2 Appearance-based target representation

Appearance-based approaches track connected regions of the input image that roughly correspond to the planar shape of objects. These models can be divided in [70]:

- template-based, being a template a sample image of the tracked region,
- view-based methods, which use subspace models of appearance obtained from a set of training images which represent different views of the object,
- global statistic based methods, which use local and global image statistics, and
- motion-based methods, that integrate motion estimates through time.

2.2.2.1 Template-based appearance models

The simplest template representation is to use a fixed template of the target to track. This approach can be reliable over short periods of time, but it copes poorly with the appearance changes that occur in most applications. In general it is preferable to have a template that

is updated over time. A fast updating scheme that acquires the template from the preceding frame [132] will fail at the presence of occlusions or abrupt changes in lighting conditions. To make the tracking robust to these factors, an appropriate temporal update of the template which uses the entire sequence up to the current frame is needed. In the work of Tao *et al.* [140] the template is updated using a weighted sum between the old template and the current data. Nguyen *et al.* [105] tracked rigid objects using a template matching approach where the intensities in the template are estimated by robust and adaptive Kalman filters. They used a Kalman filter for each pixel of the template. Using this template, the algorithm can find the object position accurately. Besides, it is robust against occlusions. The main problem of this approach is that it employs intensity as feature space and, therefore, it is not robust against strong and abrupt illumination changes. This drawback is solved in their more recent work [103, 104], where photometric invariant colour features are used. Nevertheless, these approaches are pixel-based, and they do not take into account the colour of neighbouring pixels.

Another problem to be solved in template-based target representations is the high computational cost derived from the matching process which involves cross-correlating the template with the scene image and computing a measure of similarity between them to determine the displacement. To solve this problem, Rucklidge [124] proposed a template matching approach based on a novel multi-resolution search strategy. This method divides the search space into rectilinear cells and determines which cells could contain a good match. The cells that pass the test are divided into subcells, which are examined recursively. The rest are pruned.

Another approach to reduce the computational cost associated with the template matching process is to use an image pyramid for both the template and the scene image, and to perform the matching by a top-down search. Most work presents in the literature is related to image registration and not with template-based tracking, but in both cases the interest area is represented as a template. First attempts to use pyramids in template matching were done back in 1977 by Vanderbrug and Rosenfeld [156]. They used a subwindow first to find probable candidates of the corresponding window in the reference image and then the full-size window was applied. They discussed the appropriate choice of the subwindow size to minimize the expected computational cost. In other work, Rosenfeld and Vanderbrug [123] proposed to use first both the sensed and the reference images at a coarser resolution and then, on locations with small error measure, to match higher resolution images. Althof *et al.* [2] proposed to decrease the necessary computational load by taking just a sparse regular grid of windows for which the cross correlation matching is performed. These techniques are simple examples of pyramidal methods. The linked pyramid is used in [37]. Wong and Hall [164] combined the sequential

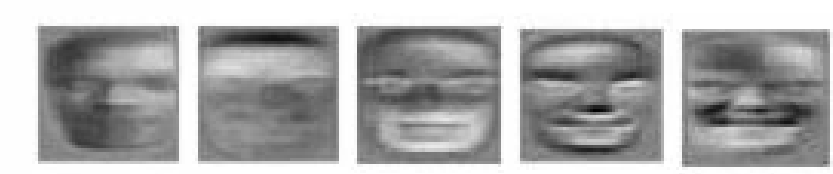

Figure 2.3: Images obtained using eigenspace representations.

similarity detection algorithm (SSDA) with pyramidal speed-up. Thévenaz et al. [143] applied a cubic spline based pyramid along with the minimization of the mean square intensity difference between the images. Kumar et al. [114] combined different types of pyramids (Laplacian, Gaussian) with different similarity measures (cross correlation, sum of squared differences) to register aerial video sequences. Non-linear min-max filters applied in a pyramidal scheme were used by Shinagawa and Kunii in [130].

Krüger *et al.* [85, 44] have proposed Wavelet networks as an efficient representation of object templates. In this approach a face template (or image template, in general) is represented by a very small set of weighted wavelets.

2.2.2.2 View-based appearance models

Robustness can be further enhanced with the use of subspace models of appearance. Such view-based models, usually learned with Principal Component Analysis, have the advantage of modeling variations in pose and lighting. However, they have the disadvantage that they are object-specific and they require training prior to tracking in order to learn the subspace basis.

Black and Jepson [13], proposed an appearance model based on eigenspace representations. Given a set of samples images, eigenspace approaches construct a small set of basis images that characterize the majority of the variation in the training set and can be used to approximate any of the training images. In [13], only a small number of samples views are represented from only a few orientations and objects are recognized in other orientations by recovering a parameterized transformation (or warp) between the image and the eigenspace.

Fig. 2.3 shows some examples of basis images of a face obtained using eigenspace representations.

Hager and Belhumeur [54] explicitly modelled the motion of the target pixels, the illumination changes and the occlusions. They model image variation due to changing illumination by low-dimensional linear subspaces. The image motion of points (geometric distortion) within

a target region are modeled using low-order parametric models. These models are incorporated into an efficient estimation algorithm which establishes temporal correspondence of the target region by simultaneously determining both motion and illumination parameters. Finally, in the case of partial occlusion, they apply results from robust statistics to develop automatic methods of rejecting occluded pixels.

Buenaposada *et al.* [21] modelled changes in appearance with a linear subspace model of gray-level texture which is computed using Principal Component Analysis. The motion parameters are estimated using a set of motion templates computed during the training step.

In the above mentioned methods the subspace model is calculated during the training process and it is not updated during the tracking. Ho *et al.* [60] extended the previous work by incorporating the capability of updating the eigen-model.

A different approach to view-based models are the Active Appearance Models (AAMs), which were introduced by Cootes and co-workers in [35] [41] [161]. An AAM contains a statistical model of the shape and gray level variability in the appearance of the object in a training set of images. The training set consists of labelled images, where key landmark points are marked on each example object. Given such a set they generate the statistical model of shape and grey variation by using Principal Component Analysis (PCA). Dornaika and Ahlberg [40] extend the AAM to deal with the 3D geometry of faces using independent 3-D shape and appearance models in contrast with Cootes and his group which use combined 2-D shape and appearance.

2.2.2.3 Global statistic based methods

The use of local and global image statistics, such as color histograms, have been popular for tracking. These methods offer robustness under image distortions and occlusions. Moreover, the models are fast to learn and can be used for searching as well as tracking. Their primary disadvantage is their lack of expressiveness which limits their ability to accurately register the model to the image in many cases. Moreover, these appearance models can also fail to accurately track regions that share similar statistics with other nearby regions.

Birchfield [11] combined two image statistics to track heads: the intensity gradient in the boundary of the object and the colour histogram of the object's interior. The colour histogram model is built in a training process and it is not updated during the tracking process, so changes in lighting conditions cause errors in the system.

Comaniciu *et al.* [34] represented the target using its color histogram and an isotropic

kernel that spatially masks the target assigning smaller weights to pixels farther from the center of the target.

Perez *et al.* [109] introduced an approach which uses color histogram and a particle filter framework. Nummiaro [106] proposed an adaptive color-based particle filter tracker which introduce the colour histogram information in the particle filter. Li and Zheng [88] extended the adaptive color-based particle filter by introducing two auxiliary variables in the particle state-space which control the speed of the color histogram forgetting process.

2.2.2.4 Motion-based models

Motion-based trackers integrate motion estimates through time. With two-frame motion estimation, the appearance model is, implicitly, just the most recently observed image. This has the advantage of adapting rapidly to appearance changes. However, models often drift away from the target. This is especially problematic when the motions of the target and background are similar. Motion estimation can be improved significantly by accumulating an appearance model through time. Irani *et al.* [68] track objects using temporal integration. For each tracked object a dynamic internal representation image is constructed. This image is constructed by taking a weighted average of recent frames, registered with respect to the tracked motion (to cancel its motion). This image contains, after a few frames, a sharp image of the tracked object, and a blurred image of all the other objects. Each new frame in the sequence is compared to the internal representation image of the tracked object rather than to the previous frame.

Optimal motion estimation can be alternatively formulated as the estimation of both motion and appearance simultaneously. Jepson *et al.* [70] proposed an appearance model for motion-based tracking that combines predictive density models of appearance with components that adapt over long and short time courses. Their tracking algorithm uses this appearance model to simultaneously estimate both motion and appearance. While their adaptive model is able to handle appearance and lighting change, the authors pointed out that it is possible for their model to learn the stable structure of the background if the background moves consistently with the foreground object over a period of time. Consequently, their model may drift from the target object and lose track of it. Zhou *et al.* [168] modified the Jepson's adaptive appearance model to integrate it in a particle filter.

2.2.3 Contour- and mesh-based target representation

2.2.3.1 Contour-based representation

B-spline-based curve representation methods have attracted considerable attention since the research work carried out by Blake *et al.* [14]. B-splines are an efficient representation of curves with limited degrees of freedom. Large image features may be represented by a B-spline using a few control points, rather than as a list of pixels, and this reduction in degrees of freedom enables real-time (or near real-time) implementation of tracking algorithms feasible. Blake *et al.* [14] proposed a new method to learn the dynamics of a B-spline from training motion sequences. The disadvantage of the method is that the tracker is effective only for the relatively narrow class of shapes and motions on which it was trained. The advantage is that performance is enhanced compared with an un-trained tracker. Blake's proposal assumes that the changes of shape of objects, between frames, are very small. It also assumes that the object of interest is moving with a constant motion model. Tissainayagam and Suter [145] developed a tracker which does not need these assumptions. They applied it to track walking/running people in real time. To do that, they introduced a new shape space decomposition technique which permits fast processing, and they couple the tracker with an automatic motion-model switching algorithm in order to track complex movements.

Snakes were proposed by Kass *et al.* [76] for object segmentation and have received a great deal of attention since then. The classical snakes approach is based on deforming an initial contour toward the boundary of the object to be detected. This deformation is obtained by minimizing a global energy designed such that its (local) minimum is obtained at the boundary of the object. A detailed analysis of the Kass's snake model, emphasizing its limitations and shortcomings, is presented in [87]. This traditional snake approach has two major problems: i) it is unable to track objects that are partially occluded, and ii) when the change between two consecutive frames is large, due to, for example, a fast movement of the object or the camera, tracking of the snake cannot be guaranteed. To solve the last problem Kim *et al.* [77] used an optical flow algorithm to estimate the object's motion. However, the computation of the optical flow field for the entire area of interest lead to a considerable computational complexity. Peterfreund [110] presented a new class of active contour models, named Velocity Snake, which results from applying a velocity control to the snakes. This work is extended in [111] by including a Kalman filter in the velocity snake model. This tracking scheme is robust to partial occlusions, large variance between frames and to image clutter. Sun *et al.* [137] have recently proposed a new snake approach, called VSnake, which solves the problems of the classical one. The active

contour energy is defined so as to reflect the energy difference between two contours instead of the energy of a single contour. The methods described above use only intensity information, so it is difficult to separate an object from its complex background. Seo *et al.* [128] recently proposed a new approach which uses color information in the snake model.

As it was previously mentioned, snakes are deformable models that are based on energy minimization along a curve. The snake model is a linear model and thus an efficient and powerful tool for object segmentation and edge integration, especially when there is a rough approximation of the boundary location. There is however an undesirable property that characterizes this model. It depends on the parameterization. The model is not geometric, and thus the solution space is constrained to have a predefined shape.

Geodesic active contours were introduced by Caselles *et al.* [26] as a geometric alternative for snakes. The geodesic active contour model is both a geometric model as well as energy functional minimization. Although the geodesic active contour model has many advantages over the snake, its main drawback is its nonlinearity that results in inefficient implementations. Goldenberg *et al.* [50] introduced a new method that maintains the numerical consistency and makes the geodesic active contour model computationally efficient. The efficiency is achieved by limiting the computations to a narrow band around the active contour and by applying an efficient re-initialization technique.

2.2.3.2 Mesh-based representation

Meshes or dynamic meshes to track deformable objects with deformable boundaries were introduced by Toklu *et al.* [147]. Although there was prior work in mesh-based motion estimation and compensation, these methods did not address tracking of an arbitrary object in the scene, since they treated the whole frame as the object of interest. A dynamic 2-D mesh consists of geometry (a set of node points), connectivity (a set of triangular elements) and motion (each node is attributed a sequence of motion vectors describing its temporal trajectory) information. Van Beek *et al.* [155] proposed a hierarchical representation of meshes and a new method to track these hierarchical meshes (see Fig. 2.4). The proposed representation consists of a hierarchy of Delaunay meshes, which models object geometry and motion at various levels of detail. The 2D dynamic mesh representation used in this work allows modeling of mildly deformable object surfaces, without occlusions. The occlusions were handled by Celasun *et al.* [64]. Griffin and Kittler [52] introduced a new mesh-based approach that improves the quality of feature matches in cases when the video sequence is rich in perspective effects and 3D camera induced

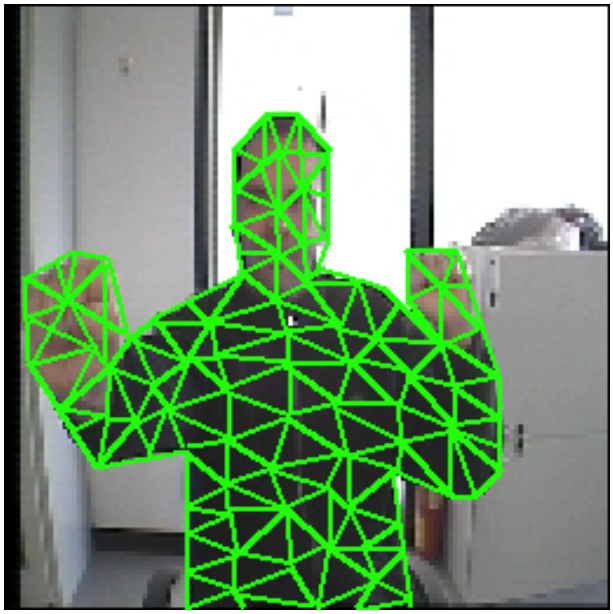

Figure 2.4: Hierarchical mesh geometry.

motion from static scenes. This is achieved by fitting an active mesh to the sequence and then matching not features, but-mesh induced planar patches. They group together single features and treat them not as isolated 2D points, but as part of 3D entities each undergoing a single motion. Sclaroff and Isidoro [126] combined a mesh and a color texture map to represent the shape of the tracked object.

2.2.4 Feature-based target representation

Feature-based tracking involves feature extraction and feature matching. Parameters such as corners and edges have been used as features for the purpose of tracking. There are two broad approaches to feature-based tracking:

- Static feature tracking. Feature tracking is termed static when features are extracted in each frame a priori and the algorithm computes the optimal correspondence between them.

- Dynamic feature tracking. In dynamic feature-based tracking the features are determined and tracked over consecutive frames dynamically, estimating motion of a feature and searching for it in the next frames.

2.2.4.1 Dynamic feature tracking

Lucas and Kanade [91] proposed a dynamic image registration technique which makes use of the spatial intensity gradient of the images to iteratively find a good match between frames. The method defines the measure of match between fixed-size feature windows in the past and current frame as the sum of squared intensity differences over the windows. The displacement is then defined as the one that minimizes this sum. Tomasi and Kanade [148] extended the previous approach to track the motion of features in an image stream by including a method to automatically select the appropriate image windows. Later, Shi and Tomasi [149] extended this technique by incorporating an affine transform to handle rotation, scaling, and shearing of objects. This algorithm is popularly known as the *Kanade-Lucas-Tomasi (KLT)* algorithm and has been widely used since its inception. Gonzalez *et al*. [51] modified the KLT for robustness by clustering of feature points undergoing the same motion. Singh *et al*. [133] included in the KLT two weight functions that reduce the matching error in noisy sequences.

2.2.4.2 Static feature tracking

Static algorithms are preferred when tracking a dense field of similar objects. They use cost functions and optimization strategies to find the matches between features. Researchers attempt to apply one-to-one mapping constraints to resolve motion correspondence. Static algorithms can be further classified, according to the method used to resolve correspondence, as:

- statistical methods,
- heuristic methods, and
- qualitative methods.

Statistical methods

Statistical methods of feature tracking represent the location of feature points as probability density functions and not as specific locations. These methods rely more on filtering and data association than in target representation. Specifically, they are based on data association

techniques as, for example, the Multiple Hypothesis Filter (MHF) [48] and the Joint Probabilistic Data-Association Filter (JPDAF) [144].

Heuristic methods

Other researchers have attempted to solve the motion correspondence problem with deterministic solutions. The most common approach in these methods is the use of a greedy exchange algorithm (an algorithm that always takes the best immediate, or local, solution while finding an answer). Greedy algorithms find the overall, or globally, optimal solution for some optimization problems, but may find less-than-optimal solutions for some instances of other problems. Not only are the heuristic methods computationally simpler, but they also have a smaller set of parameters to be investigated. It is easy to incorporate additional constraints such as motion velocity and smoothing cost functions to the heuristics. Smith and Brady [134] employed global correspondence using an iterative greedy algorithm to track corners. Shafique and Shah [129] presented a non-iterative greedy algorithm for multiframe point correspondence.

Qualitative methods

Veenman *et al.* [157] incorporated available motion knowledge to build motion models, which resolve the motion correspondence to a certain degree. They proposed three motion models: individual motion models (IMM), combined motion models (CMM) and global motion models (GMM). They also discussed different strategies to satisfy these models. IMMs represent the motion of individual features. Properties such as inertia and rigidity were incorporated in the individual models. A motion smoothness constraint was imposed on a set of points to develop the CMMs, and was extended over the whole sequence to develop the GMMs. These models made it easier to find specific strategies for optimal solutions among the large number of candidate solutions. The optimization algorithm optimizes the framework greedily by only considering two consecutive frames at the same time. Veenman *et al.* in [158] reported an improved optimization scheme which establishes the correspondence decisions using an extended temporal scope. This scheme has indeed improved the tracking performance at a limited computational cost. The limitation of this framework is that it allows for the tracking of a fixed number of points. In a subsequent work, Veenman *et al.* [159] generalized the problem by lifting this restriction, so that the number of tracked points may vary over time.

2.2.5 Hybrid target representation

Marques and Llach [92] and Tsaig and Averbuch [151] proposed a similar hybrid approach to track moving objects. These algorithms exploit an image representation as a partition hierarchy and track video objects based on interactions between different levels of the hierarchy. The hierarchy is composed of an object level and a region level. The object level defines the topology of the video objects. The region level defines the topology of homogeneous areas constituting the objects. This characteristic allows the tracking system to deal with the deformation of objects. This flexibility is obtained at the cost of a higher computational complexity. Such complexity is due to the use of complex motion models to project and adapt the regions from one frame to another.

In order to overcome such limitations, Cavallaro *et al.* [27] proposed a tracking algorithm which computes the temporal evolution of the object partition through interactions with the region partition. These interactions exploit the tracking of the region partition to associate the data from two successive object partitions, thus resulting in a multilevel tracking algorithm. A distinctive feature of the proposed algorithm is to operate on region descriptors instead of regions themselves. Projecting a region descriptor instead of the entire region is a simple and effective strategy. The simplicity comes from the fact that instead of projecting the entire region into the next frame, only the region descriptor needs to be processed. Therefore, there is no need for computationally expensive motion models. In addition, region descriptor projection is effective, since it can cope with deformation and complex motion, when updating the feature values in the region descriptor by refining the predicted region partition.

2.3 Pyramids as Target Representation tools

Pyramids have been widely used as image processing structures due to their capability to represent an input image at different resolution levels. This capability allows to reduce the computational load associated with such image processing tasks. Although the main application of these structures is image segmentation, they were developed as general purpose structures because of their hierarchical way of representing the information. This hierarchical representation of the information can be useful for many image processing tasks, such as feature extraction, registration or tracking. In fact, the building process of a pyramid involves the representation of the contents of an image at multiple levels of abstraction. Each level of abstraction corresponds to a physical level of the structure. This coarse-to-fine representation of the information can be

exploited in template-based target representation to hierarchically represent the tracked object and to reduce the high computational time associated with template matching approaches.

In this section a revision of the main pyramids present in the literature is made. This revision is detailed in order to make easier to understand the pyramidal structure proposed in Chapter 3 of this Thesis and its advantages.

2.3.1 General structure of a pyramid

Jolion and Montanvert [72] described the principle of the pyramidal approach: "a global interpretation is obtained by a local evidence accumulation". In order to accumulate this local evidence, a pyramid represents the contents of an image at multiple levels of abstraction. Each level of this hierarchy is at least defined by a set of vertices V_l connected by a set of edges E_l. These edges define the horizontal relationships of the pyramid and represent the neighbourhood of each vertex at the same level (intra-level edges). Another set of edges define the vertical relationships by connecting vertices between adjacent pyramid levels (inter-level edges). These inter-level edges establish a dependency relationship between each vertex of level l+1 and a set of vertices at level l (reduction window). The vertices belonging to one reduction window are the sons of the vertex which defines it. The value of each parent is computed from the set of values of its sons using a reduction function. The ratio between the number of vertices at level l and the number of vertices at level l+1 is the reduction factor. Using this general framework, the local evidence accumulation is achieved by the successive building of level $G_{l+1} = (V_{l+1}, E_{l+1})$ from level $G_l = (V_l, E_l)$. This procedure consists of three steps:

1. Selection of the vertices of G_{l+1} among V_l: This selection step is a decimation procedure and selected vertices V_{l+1} are called the surviving vertices.

2. Inter-level edges definition: Each vertex of G_l is linked to its parent vertex in G_{l+1}. This step defines a partition of V_l.

3. Intra-level edges definition: The set of edges E_{l+1} is obtained by defining the adjacency relationships between the vertices V_{l+1}.

The parent-son relationship defined by the reduction window may be extended by transitivity down to the base level. The set of sons of one vertex in the base level is named its receptive field. The receptive field defines the embedding of this vertex in the original image. Global properties of a receptive field with a diameter d can be computed in $O(log(d))$ parallel processing steps

using this parent-son relationship. In a general view of the pyramid hierarchy, the vertices of the bottom pyramidal level (level 0) can be anything from an original image pixel via some general numeric property to symbolic information, e.g. a vertex can represent an image pixel grey level or an image edge. Corresponding to the generalization of the vertex contents, the intra-level and inter-level relations of the vertices are also generalized.

The efficiency of a pyramid to represent the information is strongly influenced by two related features that define the intra-level and inter-level relationships. These features are the data structure used within the pyramid and the decimation scheme used to build one graph from the graph below [19]. The choice of a data structure determines the information that may be encoded at each level of the pyramid. It defines the way in which edges E_{l+1} are obtained. Thus, it roughly corresponds to setting the horizontal properties of the pyramid. On the other hand, the reduction scheme used to build the pyramid determines the dynamics of the pyramid (height, preservation of details ...). It determines the surviving vertices of a level and the inter-level edges between levels. It corresponds to the vertical properties of the pyramid. Taking into account these features, pyramids have been roughly classified as regular and irregular pyramids. A regular pyramid has a rigid structure where the intra-level relationships and the reduction factor are constant. In these pyramids, the inter-level edges are the only relationships that can be changed to adapt the pyramid to the image layout. The inflexibility of these structures has the advantage that the size and the layout of the structure are always fixed and well-known. However, regular pyramids can suffer several problems [3, 12]: non-connectivity of the obtained receptive fields, shift variance, or incapability to represent elongated objects. In order to avoid these problems, irregular pyramids were introduced. In the irregular pyramid framework, the spatial relationships and the reduction factor are not constant. Original irregular pyramids presented a serious drawback with respect to computational efficiency because they gave up the well-defined neighbourhood structure of regular pyramids. Thus, the pyramid size cannot be bounded and hence neither can the time to execute local operations at each level [163]. This problem has been resolved by recently proposed strategies [19, 56, 63, 86].

2.3.2 Regular pyramids

Regular pyramids can be explained as a graph hierarchy. However, it is more usual to represent them as a hierarchy of image arrays due to their rigid structure (see Fig. 2.5). Thus, a node of a regular pyramid can be defined by its position (i, j, l) in the hierarchy, being l the level of the pyramid and (i, j) its (x, y) coordinates within the level. In each of these arrays two nodes are neighbours if they are placed in adjacent positions of the array. The possibility to express the

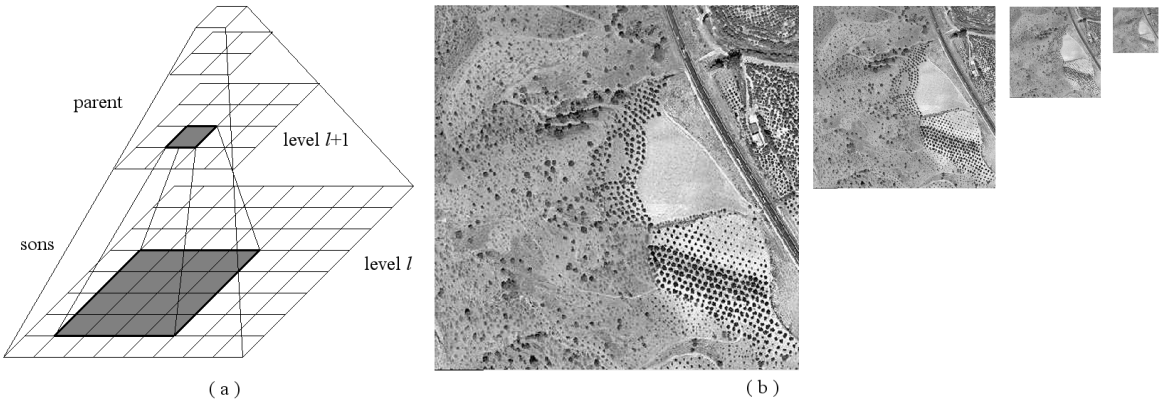

Figure 2.5: Regular pyramids: a) A 4x4/4 regular pyramid; and b) different levels of a 2x2/4 pyramid.

regular pyramids as a hierarchy of image arrays with well-defined neighbourhood relationships is the main advantage of these kind of pyramids, because it allows to build and traverse them with a low computational cost.

2.3.2.1 Regular pyramid data structure

The usefulness of pyramidal structures in image processing was firstly pointed out in [119, 139]. In these pyramids, inter- and intra-level relationships are fixed, so the structure only reduces the resolution of the image in successive levels. On the base level of the pyramid, the vertices represent single pixels and the neighbourhood of the vertices is defined by the 4- or 8-connectivity of the pixels (Fig. 2.5.a). Each pyramid level is recursively obtained by processing its underlying level. Fig. 2.5.b) shows that these pyramids generate a set of bandpass-filtered versions of an image, and they do not exploit their intrinsic capability to reliable delineate the significant features in an image [122]. The son-parent relationships are fixed and for each vertex in level l+1, there is a NxN reduction window of sons at level l. A regular pyramid is thus defined by the ratio NxN/q, where NxN is the size of the reduction window and q the reduction factor or fixed ratio between the sizes of two consecutive levels of the pyramid [18]. When the ratio NxN/q is greater than one, reduction windows are overlapped, and the parent selection scheme can be easily modified: each vertex v_i at level l could now be linked to any of its potential parents, which are the set of vertices at level l+1 whose reduction window include v_i. Therefore, in a regular pyramidal structure, inter-level relationships could adapt itself to the image layout.

2.3.3 Irregular pyramids

Irregular pyramids were introduced in order to solve the problems of the regular pyramids derived from their lack of flexibility. In contrast to regular pyramids, irregular ones have variable data structures and decimation processes which dynamically adapt to the image layout. Thus, the reduction factor between adjacent levels is not fixed; the size of each level and the height of the structure are unknown. Consequently, the well-defined and easy to compute neighbourhood relationships among nodes of the regular structures are lost in the irregular ones.

Initial attempts to develop adaptive hierarchical structures were done in the eighties (i.e. custom-made pyramids [108] and Voroni tesselation based approaches [121, 28]). The first irregular pyramid to be applied in image analysis was proposed by Montanvert et *al*. [100]. They employed a stochastic decimation algorithm [97] to construct irregular tessellations and generate a hierarchical representation of the input image. This representation was built bottom-up and adapted to the content of the input image.

Irregular pyramids allow coarse-to-fine strategies by encoding a hierarchy of successively reduced graphs. Level l is represented by a graph $G_l = (V_l, E_l)$ consisting of vertices $v \in V_l$ and edges $e \in E_l$. In this hierarchy, each graph G_{l+1} is built from G_l by selecting a subset of V_l. The selected vertices are called surviving vertices. Non-surviving vertices of V_l are linked to surviving ones. Thus, each vertex v of G_{l+1} has associated a set of vertices of G_l, the reduction window of v, which includes itself and all non-surviving vertices linked to it [100]. This is a decimation process which requires rules for:

- The selection of the vertices V_{l+1} among V_l. These vertices are the surviving vertices of the decimation process.
- The allocation of each non-surviving vertex of level l to a survivor, which generates the son-parent edges.
- The creation of edges E_{l+1} by defining the adjacency relationships among the surviving vertices of level l.

The receptive field of one surviving vertex is defined by the transitive closure of the parent-son relationship and must be a connected set of vertices in the base level. Rules for the definition of the set of surviving vertices and the set of edges connecting each non-surviving vertex to its parent vary according to the considered decimation algorithm used within the irregular pyramid [80]. Therefore, the reduction procedure used to build one graph from the one below strongly

influences the efficiency of the pyramid. On the other hand, each level of the hierarchy is encoded by a graph and, since many graph algorithms suffer from a high computational complexity, the efficiency of the irregular pyramid is also influenced by the selected graph encoding. Next subsections present different graph encodings and decimation algorithms used within the irregular pyramid framework.

2.3.3.1 Irregular pyramid data structures

Irregular pyramid data structures can be classified as:

- Simple Graphs [100]. This is the simplest data structure where the pyramid is defined as a stack of successively reduced simple graphs. This type of structures have two main drawbacks for image processing tasks: i) they do not allow to know if two adjacent receptive fields have one or more common boundaries, and ii) they do not allow to differentiate an adjacency relationship between two receptive fields from an inclusion relationship.
- Dual Graphs [163]. This structure solves the drawbacks of the simple graph approach representing each level of the pyramid as a dual pair of simple graphs and computing contraction and removal operations within them. The problem of this structure is the high increase of memory requirements and execution times since two data structures needs to be stored and processed.
- Combinatorial Maps [18]. The combinatorial map is an efficient implementation of the dual graph approach which solves its aforementioned drawbacks. To do that, the combinatorial map approach uses an only planar graph to represent each level of the pyramid, which encodes explicitly the orientation of edges around the vertices instead of a pair of dual graph. In this planar graph it is possible to perform the contractions and removals operation using a set of permutations within the graph.

Simple Graph

A simple graph is a non-weighted and undirected graph containing no self-loops. In this hierarchy, a pyramidal level l is defined by a graph $G_l = (V_l, E_l)$, where the set of vertices V_l represents a partition of the image into connected subsets of pixels. The graph edges E_l represent adjacency relationships among pyramidal vertices of the level l. Two vertices are connected

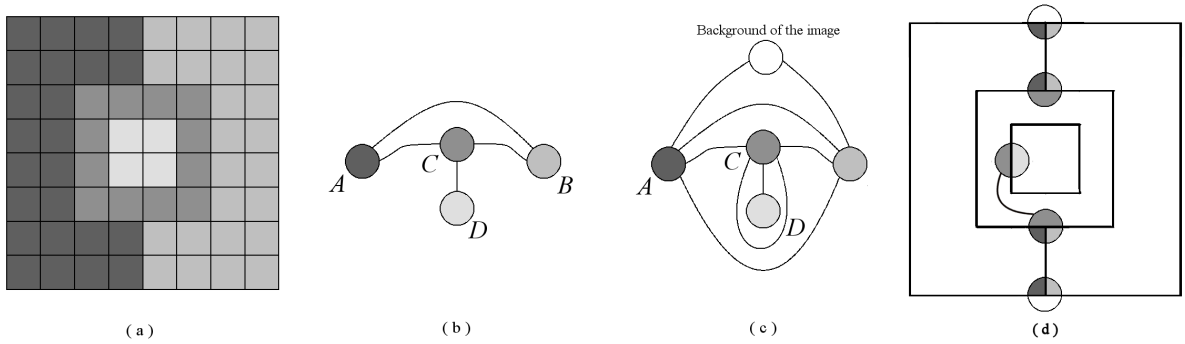

Figure 2.6: Codification of connected components by several irregular pyramid data structures: a) 8x8 image layout; b) encoding by a simple graph pyramid; c-d) encoding by a dual graph or combinatorial pyramids.

if there exists a connecting path in level l-1 that joins them. A path in G_{l-1} is a connecting path of two surviving vertices $v, v' \in V_l$ if it satisfies one of the following conditions [80]:

- v and v' are connected by an edge $e \in E_{l-1}$.
- v and v' are connected by a path (e_1, v_i, e_2), where v_i is a non-surviving vertex connected to v or v'.
- v and v' are connected by a path $(e_1, v_i, e_i, v_j, e_2)$, where v_i and v_j are two non-surviving vertex connected to v and v', respectively.

Simple graphs encode the adjacency between two vertices by only one edge, although their receptive fields may share several boundary segments. Therefore, a graph edge may thus encode a non-connected set of boundaries between the associated receptive fields. Moreover, the lack of self-loops in simple graphs does not allow to differentiate an adjacency relationship between two receptive fields from an inclusion relationship. These facts are shown in Fig. 2.6.b), which represents the top of a simple graph pyramid encoding the connected components of Fig. 2.6.a).

Dual Graph

In a dual graph pyramid, a level consists of a dual pair $(G_l, \bar{G}_l)$ of planar graphs G_l and $\bar{G}_l$. The vertices of G_l represent the cells on level l and the edges of G_l represent the neighbourhood relationships of the cells on level l. The edges of $\bar{G}_l$ represent the boundaries of the cells in level

l and the vertices of $\bar{G}_l$ define meeting points of boundary segments of $\bar{G}_l$. Fig. 2.6.c) represents the top of a dual graph pyramid encoding the connected components of Fig. 2.6.a). Fig. 2.6.d) shows the dual graph corresponding to 2.6.c).

Within the dual graph pyramid framework, the set of edges that defines the adjacency relationships among pyramidal vertices of the level $l+1$ is generated in two steps. First, the set of edges that connects each non-surviving vertex to its parent is contracted using a contraction kernel. A contraction kernel of a level l is the set of surviving vertices of l and the edges that connect each non-surviving vertex with its parent. The edge contraction operation collapses two adjacent vertices into one vertex, removing the edge between them. This operation may create redundant edges such as empty self-loops or double edges. The removal of these redundant edges constitutes the second step of the creation of the set of edges E_{l+1}. These redundant edges are characterized in the dual graph and removed by a set of edge removal kernels [81]. The key idea of the dual graphs is that a contraction in a graph implies a removal in its dual, and viceversa, in order to maintain the duality between the newly generated graphs. Thus, the generation of the edges in level $l+1$ can be resumed as follows:

1. Contraction of edges in G_l which connect non-surviving vertices with their parents. Removal of their corresponding edges in $\bar{G}_l$. Fig. 2.7.b) shows the reduction performed by the contraction kernel in Fig. 2.7.a).

2. Contraction of redundant edges in $\bar{G}_l$ and removal of their corresponding edges in G_l. In Fig. 2.7.c), the dual vertex a has a face defined by vertices **A** and **B**. The boundary between the regions defined by these vertices is artificially split by this dual vertex. Then, the two dual edges incident to this dual vertex (e'_1 and e'_2) can be contracted. The contraction of these dual edges has to be followed by the removal of one associated edge (e_1 or e_2) in order to maintain the duality between both graphs. In the same way, the dual vertex b encodes an adjacency relationship between two vertices contracted in the same vertex. This relationship can be removed by eliminating this direct self-loop and contracting the associated dual edge.

Using such a reduction scheme each edge in the reduced graph corresponds to one boundary between two regions. Moreover, inclusion relationships may be differentiated from adjacency ones in the dual graph.

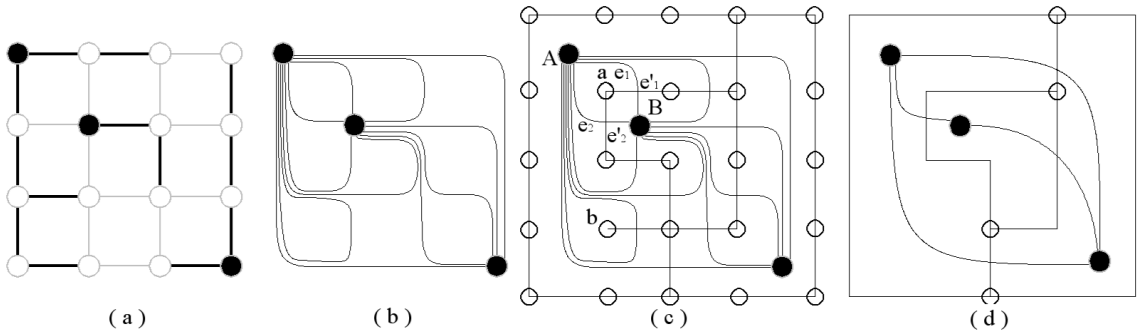

Figure 2.7: Contraction and removal kernels: a) contraction kernel composed of three vertices (surviving vertices are marked in black); b) graph G obtained after contractions of the trees defined in a); c) redundant edges characterisation; and d) dual graph pair $(G, \bar{G})$ after dual decimation step.

Combinatorial Map

A combinatorial map may be defined as a planar graph encoding explicitly the orientation of edges around a given vertex [18]. Fig. 2.8 illustrates the derivation of a combinatorial map from a plane graph. Firstly, edges are split where their dual edges cross (see Fig. 2.8.b)). These half-edges are called darts and have the origin at the vertex they are attached to. A combinatorial pyramid is defined by an initial combinatorial map successively reduced by a sequence of contraction or removal operations [19].

A combinatorial map can be expressed as $G = (\mathcal{D}, \sigma, \alpha)$, where $\mathcal{D}$ is the set of darts and σ and α are two operations defined on $\mathcal{D}$. α allows to know which two darts stem from the same edge and is called "reverse permutation". σ is used to compute which darts are around a given vertex and it is named "successor permutation". Another important operation φ is defined over the combinatorial map which allows to know which darts are around a given face of G. This operation is the same permutation than σ but calculated in the dual graph $\bar{G}$. The advantage of this representation of graphs using α, σ and φ is that φ can be also computed over G as a combination of α and σ: $\varphi = \sigma \circ \alpha$. Thus, the dual graph is implicitly encoded in the combinatorial map G.

The advantage of the combinatorial map based pyramid representation is that the contraction and removal operations can be performed knowing only the combinatorial map G and the permutation operations α, σ and φ. $\bar{G}$ is not needed. The advantages of the dual graph representation are kept without a high increase of the computational cost.

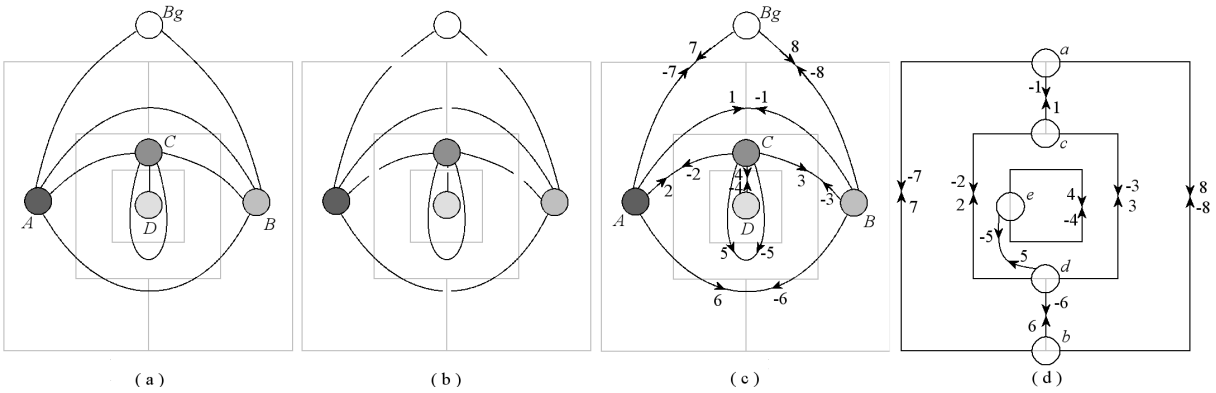

Figure 2.8: Combinatorial map: a) a plane graph; b) edges splitting; c) combinatorial map G; d) dual map of G.

2.3.3.2 Irregular pyramid decimation schemes

Although original irregular pyramids overcome the drawbacks of regular ones, their main drawback is that they only grow to a reasonable height as long as the base level is small. If the base level size gets larger, the reduction factor cannot be bound because the progressive deviation from the regular base favours configurations that slow down the contraction process [57]. This height increasing degrades the efficiency of irregular pyramids. Recent work has resolved this problem by new selection mechanisms which guarantee logarithmic heights [83]. Next subsections deal with different reduction schemes used to build the irregular pyramid. These schemes determine the height of the pyramid and the properties that arise from the decimation process.

Stochastic decimation process

If $G_l = (V_l, E_l)$ represents the level l of the hierarchy, where V_l defines the set of vertices of the graph and E_l the set of edges, the stochastic decimation process introduced by Meer [97] imposes two constraints on the set of surviving vertices, V_{l+1}:

1. Any non-surviving vertex v of level l has at least one surviving vertex in its neighbourhood, v'.

2. Two neighbour vertices v and v' at level l cannot both survive.

These rules define a maximal independent set (MIS). In order to build this MIS the decimation

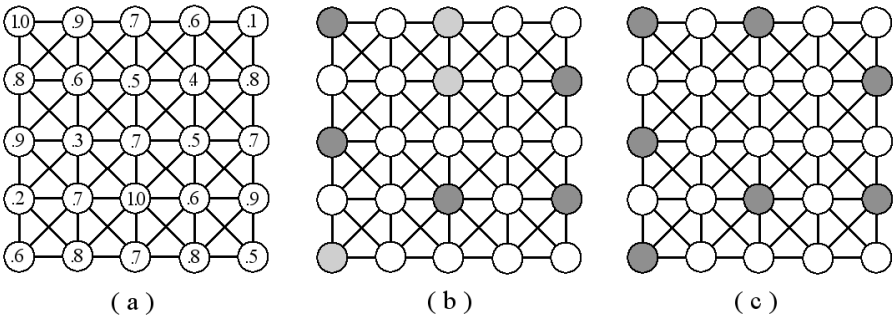

Figure 2.9: Stochastic decimation procedure: a) 8-connected valuated graph; b) extraction of local maxima (dark grey vertices) and their neighbours (white vertices); and c) complete specification of the set of surviving vertices (light grey vertices).

algorithm uses three variables for each vertex v_i: two binary-state variables p_i and q_i, and a random variable x_i uniformly distributed between [0,1]. The surviving vertices are chosen by an iterative local process. A vertex v_i in V_l survives if, at the end of the algorithm -iteration k-, its $p_i(k)$ state value is true. In the first iteration:

- $p_i^{l+1}(1)$ of a vertex v_i is set to 1 (true) if its x_i value is the maximum x value in its neighbourhood (local maximum). It must be noted that the local maximum nodes are selected as surviving vertices in the first iteration.
- $q_i^{l+1}(1)$ is set to 1 if v_i is not a local maximum and there is not a local maximum (node with $p^{l+1}(1) = 1$) in its vicinity.

In the rest of iterations the nodes with $q_i^{l+1}(n-1) = 1$ are studied. Thus, a node with $q_i^{l+1}(n-1) = 1$ is set to $p_i^{l+1}(n) = 1$ and $q_i^{l+1}(n) = 0$ if it is the local maximum among its neighbours with $q_i^{l+1}(n-1) = 1$. This process is iterated until $q_i^{l+1}(n)$ is false for all vertex v_i. The set of sons are defined in G_l only after the vertices of G_{l+1} (their parents) have been chosen.

In Fig. 2.9 the stochastic decimation process is shown. In Fig. 2.9.a) the x value of each vertex is represented. The first iteration of the stochastic decimation procedure is shown in Fig. 2.9.b). The dark vertices are the vertices with $p(2) = 1$ (surviving vertices) and the white vertices are their neighbours. Grey vertices are the vertices with $q(2) = 1$. In Fig. 2.9.c) the second iteration is presented. This iteration is the last one in this case because all the vertices have a surviving vertex in their vicinity.

Connectivity preserving relinking approach

Nacken [102] describes a decimation process that adapts the classical relinking rules proposed by Burt et *al*. [24] for an irregular data structure. In order to create V_{l+1}, the set of vertices of the lower level in the hierarchy, V_l, is partitioned into a number of connected reduction windows. Reduction windows are computed by applying the following iterative process:

1. Every vertex v_i which does not belong to any reduction window is given a label γ_i.
2. Every vertex whose label is larger than that of all of its neighbours is selected as a surviving vertex (centre of a new reduction window).
3. For each newly selected surviving vertex v, a maximal subset of the neighbours of v, containing no dissimilar pairs, is added to complete the reduction window. Dissimilarity of adjacent vertices must be defined using an edge strength measure.

The label γ_i can be a random number, although some image dependent value can also be employed. The difference with the stochastic decimation procedure is in the order of the steps. In stochastic decimation, the computation of a maximal independent set by repeated selection of local maxima is completed before the reduction windows are computed by assignments of neighbours; in this approach, a number of reduction windows are computed in each selection of local maxima.

The parent-son edges created in this step have the same role as the regular structure in the classical relinking scheme [24]: they serve as an initial configuration which is adapted by relinking. Then, the algorithm performs an iterative relinking process that preserves the connectivity. This process is applied vertex by vertex. For each vertex v, a set of allowed candidate parents is computed, depending on the actual structure of the hierarchy. This set plays the same role as the fixed set of candidate parents in the classical relinking scheme, with the particularity that linking v to any of this allowed parent assures that connectivity is preserved [102]. Then, a new parent is chosen from the set of allowed candidate parents. The vertex is relinked to the new parent and the graph structure and attributes of vertices are updated accordingly. This process is repeated until a stable configuration is reached. When the relinking process finishes, the next level of the hierarchy can be built.

Dual graph contraction

In [80], the building of irregular pyramids by dual-graph contraction is described. In this work, a contraction kernel is defined on a graph $G_l = (V_l, E_l)$ by a set of surviving vertices V_{l+1} and a set of non-surviving edges N_l such that:

- (V_l, N_l) is a spanning forest of G_l. A spanning forest of G_l is a subgraph that contains all the vertices of G_l and that contains no cycles. Fig. 2.7.a) shows a spanning forest of a graph.
- Each tree of (V_l, N_l) is rooted by a vertex of V_{l+1}.

Therefore, the decimation of a graph by contraction kernels differs from the stochastic decimation process in that two surviving vertices may be adjacent in the contracted graph. Also a non-surviving vertex may be connected to its parent by a branch of a tree.

Specified rate and prioritized sampling approaches

In the stochastic pyramid framework, the ratio between the number of surviving vertices and the total number of vertices (the sampling rate, which is the inverse of the reduction factor) may be different in different parts of the graph. This is because different parts of the graph consist of vertices having different numbers of neighbours. Hence, parts of the graph where vertices have a smaller number of neighbours on average can accommodate more surviving vertices than other parts of the graph. The specified rate sampling approach [67] replaces the iterative decimation process of the stochastic approach by a single step process. The decimation algorithm uses two variables for each vertex v_i: a binary-state variable p_i and a random variable x_i uniformly distributed between [0,1]. Initially, the state variable p_i of all vertices is set to 0. Then

$$p_i^{l+1} \Leftrightarrow x_i < \omega \tag{2.1}$$

That is, a vertex is selected as a surviving vertex based on a fixed probability. The variable ω determines the sampling rate and can be specified by the user. The constraints imposed by the stochastic decimation process [97] may be violated. Thus, any non-surviving vertex can have no surviving vertex in its neighbourhood, and two neighbouring vertices can be selected as surviving vertices.

Another way to increase the reduction factor is to allow some vertices to have a higher priority over others in being selected as surviving vertices. If these prioritized vertices have

larger numbers of neighbours (the number of neighbours of a node is called "degree" of the node), a larger number of vertices will become non-surviving vertices. Then, the number of selected surviving vertices is reduced. To introduce priority in the decimation scheme, a ranking approach is proposed by Ip and Lam [67]. The range [0,1] is divided into n sub-ranges. The value n is an estimated constant denoting the maximum degree of a vertex. A random variable x' is generated in the range $[0,1/n]$. The random variable x associated with a vertex is then set by

$$x = x' + (r-1)/n \quad if\ r < n \tag{2.2}$$

$$x = x' + (n-1)/n \quad if\ r \geq n \tag{2.3}$$

where r is the degree of the node. Thus, for any two neighbouring vertices with different degree, the one which has a larger degree will be usually assigned a higher priority in being chosen as a surviving vertex.

Data driven decimation scheme

One of the disadvantages of the stochastic decimation process is that vertices extracted as local maxima in the first iteration must wait until the graph is complete in successive iterations [71]. These iterations are used only to complete the maximal independent set. In the data driven decimation process (D3P), a vertex v_i of G_l survives if and only if it is a local maximum (p_i^{l+1}=true) or does not have yet any surviving vertex in its neighbourhood (q_i^{l+1}=true). Therefore, it is assumed that being a local maximum is of importance and no correction is performed in subsequent iterations. In areas where there is no real maxima, the process still tries to extract sub-maxima but without slowing down the decimation procedure in other areas of the graph. The procedure is not iteratively run.

The graph G_{l+1} defined by the D3P is slightly different to the one defined by the stochastic pyramid because two neighbours in V_l can both survive in V_{l+1}. Thus, D3P achieves faster convergence and better fits the distribution of the values associated with the vertices of the initial graph [71]. However, as for the stochastic decimation process, the D3P cannot guarantee a constant reduction factor between pyramid levels [57].

MIES and MIDES algorithms

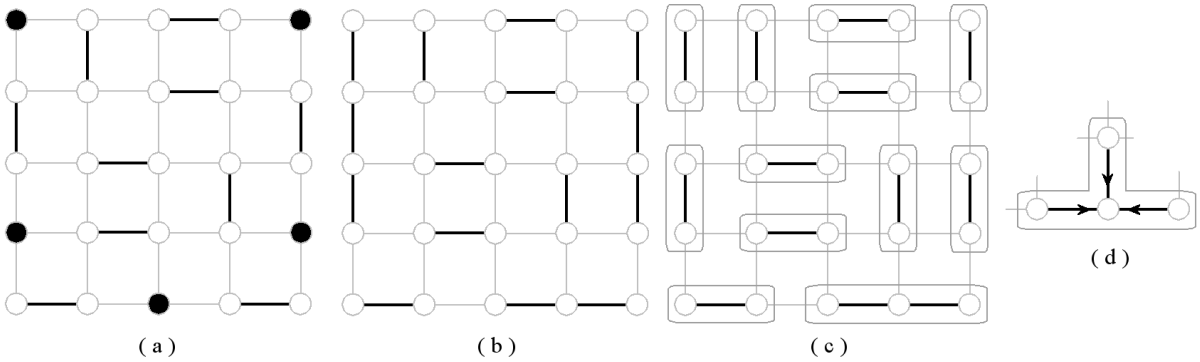

Figure 2.10: MIES algorithm: a) maximal matching M (isolated vertices are black coloured); b) enlarged matching $M+$; c) reduced matching $M+$ and contraction kernels; and d) restriction to choose the surviving vertex and direction of contraction of a contraction kernel.

Although stochastic pyramids overcome the drawbacks of regular ones, they grow higher than the base diameter for large input images. As a consequence of the greater height the efficiency of pyramids degrades. This problem has been resolved in dual graph pyramids by selection mechanisms which guarantee logarithmic heights by replacing the selection method proposed in [97] by two new iteratively local methods: Maximal Independent Edge Set algorithm (MIES) [57] and Maximal Independent Directed Edge Set (MIDES) [56].

The MIES algorithm has been developed to be applied in the dual graph framework. Its goal is to find a set of contraction kernels in a plane graph G_l such that each vertex of G_l is contained in exactly one contraction kernel, and each contraction kernel contains at least two vertices. Thus, the number of vertices between consecutive graph levels is reduced to half or less and a reduction factor of at least 2 can be guaranteed. The MIES algorithm consists of three steps [57, 83]:

1. Find a maximal independent matching M from G_l. An independent matching is a set of edges in which no pair of edges has a common end vertex (Fig. 2.10.a)).

2. Enlarge M to a matching $M+$ by connecting isolated vertices of G_l to the maximal matching M (Fig. 2.10.b)).

3. $M+$ is reduced by breaking up trees of diameter three into trees of depth one. A tree is a set of edges connected at their ends containing no closed loops (cycles) (Fig. 2.10.c)).

A maximal matching of G_l is equivalent to a maximal independent vertex set on $\bar{G}_l$ [57]. Therefore, the maximal matching can be obtained by applying the MIS algorithm in $\bar{G}_l$. The second

and three steps of the MIES algorithm allow to obtain a set of contraction kernels where each vertex belongs to a tree of depth one.

The MIES algorithm can be used either in a dual graph framework or for connected component analysis [84]. However, its main disadvantage is that it is only applicable where there are no constraints on direction of contraction [56, 83]. As it is shown in Fig. 2.10.d), there are certain contraction kernels that impose the only possible surviving vertex and, therefore, the direction of contraction.

Maximal independent directed edge set (MIDES) algorithm can be applied in oriented graphs, such as the graph applied to line image analysis [22]. In an oriented graph the relations between pairs of vertices are not symmetric, so that each edge has a directional character. Besides, this edge direction is unique (i.e., edges cannot be bi-directed). In these graphs, an edge e with source s_e and target t_e, $e = (s_e, t_e)$, must be contracted from s_e to t_e, only if the attributes of the edge e and of its source and target vertices fulfil a certain rule. The set of edges that fulfils the rule are called pre-selected edges [56]. Only these pre-selected edges are considered as candidates for contraction and the goal is to build contraction kernels with a high reduction factor. In order to perform the contractions in parallel, a vertex disjoint union of contraction kernels is needed [83]. The MIDES algorithm defines such a union in terms of independent directed edges. Two directed edges are independent if they do not belong to the same neighbourhood. The neighbourhood of a directed edge e, N_e, is defined by all directed edges with the same source s_e, targeting the source s_e or emanating from t_e [83]. Then, the contraction kernels can be found as in MIS, but dealing with edges instead of vertices. This algorithm shows better reduction factor than MIS or MIES [56, 83].

Union-find techniques

The union-find algorithm was proposed by Tarjan [141] as a general method for keeping track of disjoint sets. Basically, it allows performing of set-union operations on sets which are in some way equivalent, while ensuring that the end product of such a union is disjoint from any other set.

Brun and Kropatsch [19] propose to use the union-find algorithm to design a contraction kernel in the combinatorial pyramid framework. Union-find algorithms use tree structures to represent sets. Each non-root vertex in the tree points to its parent, while the root is flagged in some way. Therefore, each tree of a contraction kernel is encoded by storing in each vertex

a reference to its parent. Initially, the parent of each vertex v is itself. Then, the union-find algorithm performs the following operations over any dart d in $\mathcal{D}_l$:

- A find operation is applied on the origin vertices of the two darts d and $-d$ defined over the same edge. These operations return the roots, r_d and r_{-d}, of the trees containing these two vertices.
- If r_d and r_{-d} are different and they must be merged, a union operation merges the corresponding two trees into one. This union is performed by setting one of the roots to be the parent of the other root. The edge which contain d and $-d$ is included in the contraction kernel.

The union-find algorithm has proven to be very efficient, especially when it is run on sequential machines.

Chapter 3

Bounded Irregular Pyramid

In Chapter 2, a general review of computer vision approaches to tracking was attempted. Thus, as was explained in that chapter, two main parts can be distinguished in a general tracking framework: filtering and data association and target representation and localization. More importance is given to one part or to the other depending on the final application of the tracking. This Thesis is focused in the development of a tracking approach which relies in target representation and localization as the responsible of solve problems derived from changes in the appearance of tracked objects.

Target representation approaches present in the literature can be roughly divided in: model-based, appearance-based, contour-based, feature-based and hybrid methods. Each of these approaches has advantages and disadvantages which depend mainly on the necessary prior knowledge and on the requirements of the final application. This Thesis focus on the development of a tracking system which does not require any prior knowledge about the object to track and which should run in real time ($\simeq 25Hz$). In addition, the method should cope with appearance changes of the object, occlusions, changes in the environment conditions and tracking of multiple objects. To achieve these goals, this Thesis proposes a novel appearance-based target representation approach.

Appearance-based approaches can be classified into: view-based, global-statistic-based, motion-based and template-based. View-based approaches have been discarded for this Thesis because they require a training phase with a set of image samples of the object to track. Motion-based approaches have a high computational cost. Therefore, the most suitable approaches for the goals of this Thesis are the global-statistic-based and template-based approaches. Colour features have been widely used in global-statistic-based methods because they are robust to partial occlusion, scaling and object deformation. The main problem of some of these methods is that

if only spectral information is used, spatial information is lost. In this Thesis, a template-based approach has been selected because a template combines in its representation colour features with spatial information. Therefore, the proposed template-based target representation has the invariance advantages of colour-based approaches and includes spatial information, solving one of their main problems.

As it was commented in Chapter 2, one of the main problems of template based approaches is the high computational cost derived from the matching process. A main contribution of the algorithm proposed in this Thesis is the use of a new pyramid structure which permits to perform the matching process in a hierarchical way, reducing its computational cost. Chapter 2 of this Thesis presented a review of the main pyramidal structures used in image processing tasks. The main conclusion extracted after studying these pyramids, was the necessity to develop a new hierarchical structure which fulfilled to the requirements of the proposed tracking system: accurate results and low computational cost. This pyramid is the Bounded Irregular Pyramid (BIP) proposed in this Thesis. It solves the problems of regular pyramids and it has lower computational time than the irregular ones.

The goal of the Bounded Irregular Pyramid is to achieve a computationally efficient framework for template-based target representation as well as a hierarchical support for the tracking process. In this chapter the features of the BIP are discussed, presenting a comparison with the main regular and irregular pyramids previously presented in Chapter 2. In order to do the comparisons the different pyramids have been applied in a segmentation task. It has been chosen to compare the performance of the different pyramids due to two main reasons: i) in the proposed approach, target representation and segmentation are equivalent tasks with only one difference: target representation is the segmentation of only the desired object and not the segmentation of the whole image and, ii) there are well-known quantitative evaluation methods to measure the quality of segmentation results. Specifically, three types of segmentation quality measurements have been employed: the shift variance proposed in [115], the F function proposed in [89] and the Q function proposed in [15].

3.1 Introduction

In Chapter 2, a taxonomy of pyramids was presented that classified them into regular and irregular ones. Regular pyramids have a rigid structure where the decimation process is fixed. In these pyramids, the inter-level edges are the only relationships that can be changed to adapt the structure to the image layout. Thanks to their rigid structure, regular pyramids can be

represented as a hierarchy of bidimensional arrays. Each of these arrays is an image where two vertices are neighbours if they are placed in adjacent positions of the array. The possibility to express the regular pyramids as a fixed hierarchy of bidimensional arrays with well-defined neighbourhood relationships is the main advantage of this kind of pyramids, because it allows to build and traverse them with a low computational cost. But of course the simplicity of the rigid structure of regular pyramids comes with a cost [12]: non-connectivity of the obtained receptive fields, shift variance, or incapability to represent elongated objects. Irregular pyramids arose as an alternative to the inflexibility of regular structures to solve these problems. In contrast to regular pyramids, irregular ones have variable data structures and decimation processes which dynamically adapt to the image layout. Thus, the reduction factor between adjacent levels is not fixed; the size of each level and the height of the structure are unknown. Consequently, the well-defined and easy to compute neighbourhood relationships among vertices of regular structures are lost in the irregular ones. In consequence, classical irregular structures are not computationally efficient. This efficiency problem has been recently addressed using strategies such as: the height reduction achieved by the hierarchy of partitions [58], the computational efficiency of the combinatorial pyramid [19], the efficient region-growing control implemented in [86] or the combination of different procedures for uniform or non-uniform regions [63]. These new approaches are more computationally efficient than the classical ones, but they still have a execution time to date which prevents their use in real-time applications, as will be shown in Section 3.3.2 of this chapter.

The Bounded Irregular Pyramid is an irregular structure that achieves the accuracy of the main irregular structures but with lower computational cost. The key idea is to use a regular approach in the homogeneous regions of the input image and an irregular approach in the rest of regions. Specifically, the BIP's data structure is a combination of a 2x2/4 regular structure with a simple graph. Thus, while in the regular part of the BIP a regular decimation process is used, in the irregular part a union-find decimation approach is employed. The BIP solves the three main problems of regular structures and, at the same time, is computationally efficient because its regular part prevent it of a big increase of height. Specifically, as will be shown in the result section of this chapter, the height of the BIP is less than the height of the other irregular approaches.

In this chapter, the data structure and the decimation process used in the BIP are explained in Section 2. Section 3 presents the obtained experimental results and the comparisons with the main pyramidal segmentation algorithms. Finally, Section 4 makes a brief summary of the main concepts explained in this chapter.

3.2 Data structure and decimation process

In regular pyramids the son-parent relationships are fixed. For each vertex in level l+1, there is a NxN reduction window of sons at level l. The data structure of a regular pyramid is defined by the ratio NxN/q, where q is the fixed reduction factor between the sizes of two consecutive levels of the pyramid [18].

The data structure of the Bounded Irregular Pyramid is a combination of the simplest regular and irregular data structures: the 2x2/4 regular one and the simple graph irregular representation. It consists of a graph hierarchy in which each level l is represented by a graph $G_l(N, E)$, with vertices N_l, linked by a set of edges E_l. There are two types of vertices: vertices belonging to the 2x2/4 structure, named *regular* vertices, and *virtual* vertices or vertices belonging to the irregular structure.

3.2.1 Regular data structure building

A regular pyramid can be represented as a hierarchy of bidimensional arrays (images) where the vertices are represented by their positions in such arrays. Therefore, in the regular part of the BIP, each regular vertex n is represented by (i, j, l), where l represents the level and (i, j) are the x- and y-coordinate within the level.

The first step to build the 2x2/4 structure is a 4 to 1 decimation procedure. In order to perform this decimation, each regular vertex has associated two parameters:

- Homogeneity, $Hom(i, j, l)$. Regular vertices have $Hom(i, j, l) = 0$ or $Hom(i, j, l) = 1$. $Hom(i, j, l)$ of a regular vertex is set to 1 if the four vertices immediately underneath are similar according to some criteria and their homogeneity values are equal to 1. Otherwise, it is set to 0.

- Parent link, $(X, Y)_{(i,j,l)}$. If the vertex (i, j, l) is a vertex of the regular structure with $Hom(i, j, l) = 1$, then the parent link of the four cells immediately underneath (sons) is set to (i, j). It indicates the position of the parent of a regular vertex in its upper level. A regular vertex without parent has its parent link set to a NULL value. Parent links represent the inter-level edges of the regular part of the BIP.

All the regular vertices presenting an homogeneity value equal to 1 form the regular structure. Regular vertices with an homogeneity value equal to 0 are removed from the structure.

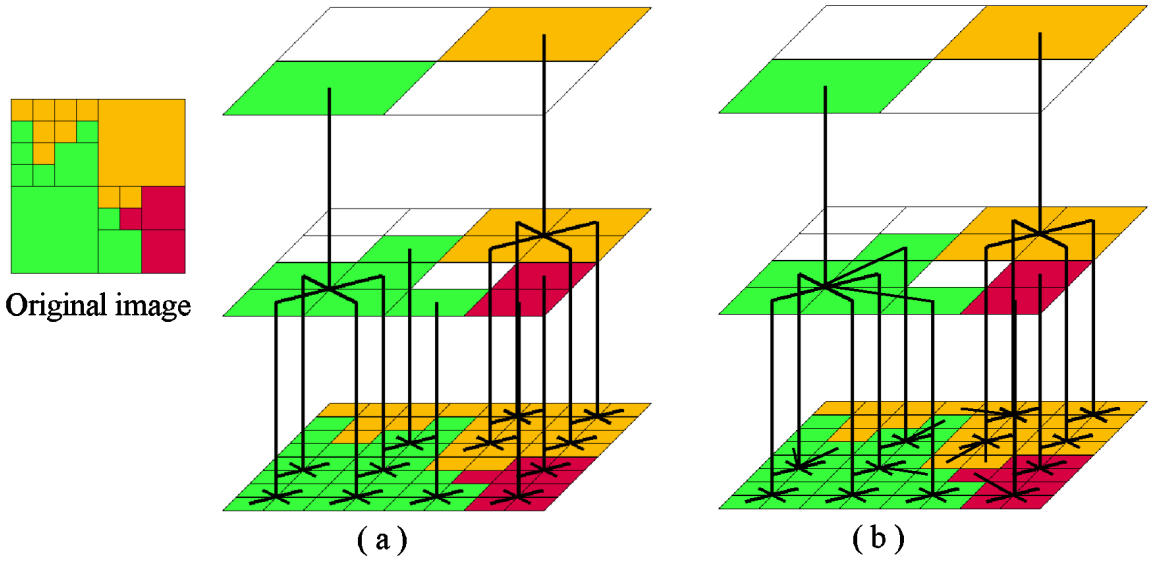

Figure 3.1: Regular vertices of the BIP and their inter-level edges a) after the generation step, b) after the parent search step.

The regular part of the BIP can be seen as an incomplete regular pyramid, i.e. where some vertices are missing, which is represented as a hierarchy of incomplete bidimensional arrays. In each of these arrays two vertices are neighbours if they are placed in adjacent positions of the array. If two vertices are neighbours at level l, their receptive fields are neighbours at the base level. Fig. 3.1.a) shows the regular part of the BIP data structure after being built. White vertices are the non-homogeneous ones. In this example the used similarity criteria is the colour distance. Two vertices are similar if they have similar colour. The base level of the structure is formed by the pixels of the 8x8 original image. The 4 to 1 decimation procedure generates a 4x4 level and a subsequent 2x2 level.

Once the regular structure is generated using the 4 to 1 decimation procedure, there are some regular orphan vertices (regular vertices without parent). From each of these vertices, a search is made for a non-orphan neighbour vertex similar to it (*parent search step*). If there are several candidate parents, the orphan vertex is linked to the most similar parent. Thus, a vertex (i, j, l) is linked to the parent $(i_p, j_p, l+1)$ of a neighbour vertex (i_1, j_1, l) that belongs to its $\xi_{(i,j,l)}$ vicinity, if the following conditions are true:

- $Hom(i, j, l) = 1 \ \& \ Hom(i_1, j_1, l) = 1$
- $d((i, j, l), (i_1, j_1, l)) < T$ (1)
- $d((i, j, l), (i_1, j_1, l)) \leq d((i, j, l), (i_k, j_k, l) \ \ \forall (i_k, j_k, l) \in \xi_{(i,j,l)}$

being $d(n_i, n_j)$ a similarity measurement between the vertices n_i and n_j and T a similarity threshold. For example, in Fig. 3.1.b), there are four orphan vertices at level 1, but only for two of them a suitable parent vertex is found that satisfies (1).

3.2.2 Irregular data structure and decimation process

The process to compute the irregular part of G_{l+1} from G_l has four stages:

- Intralevel twining: This stage links two orphan neighbour vertices of the regular structure if they are similar. To do that, from each regular orphan vertex, (i, j, l), a search is made for all neighbour orphan vertices at the same level,(i_1, j_1, l), which satisfy the following conditions:

  $$\begin{aligned} &- (X, Y)_{(i_1,j_1,l)} = NULL \\ &- Hom(i_1, j_1, l) = 1 \\ &- d((i, j, l), (i_1, j_1, l)) < T \end{aligned} \tag{2}$$

  Among the set of candidates, the studied vertex is linked with the most similar to it, generating a virtual vertex at level $l + 1$. In Fig. 3.2 the two regular vertices n_1 and n_2 are linked, generating the virtual vertex m_1.

- Virtual vertices linking: this process links two virtual orphan vertices of the level l if they are similar. A virtual orphan vertex $n_i \in N_l$ is linked with a virtual orphan vertex $n_j \in N_l$, generating a virtual vertex in the graph $G_{l+1}(N, L)$, if they satisfy the following conditions:

  $$\begin{aligned} &- n_j \in \xi_{n_i} \\ &- d(n_i, n_j) < T \\ &- d(n_i, n_j) \leq d(n_i, n_k) \quad \forall n_k \in \xi_{n_i} \end{aligned} \tag{3}$$

  n_j is in the vicinity of n_i, ξ_{n_i}, if their corresponding reduction windows $w_i \in N_{l-1}$ and $w_j \in N_{l-1}$ are neighbours in the graph $G_{l-1}(N, L)$. Two reduction windows $w_i \in N_{l-1}$ and $w_j \in N_{l-1}$ are neighbours if there are at least two vertices $n_r \in w_i$ and $n_s \in w_j$ which are connected by an edge $e_t \in E_{l-1}$:

  $$\exists (n_r, e_t, n_s) / n_r \in w_i, n_s \in w_j \tag{4}$$

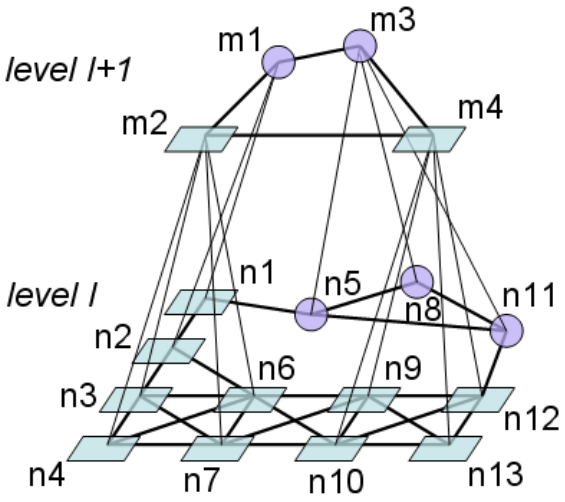

Figure 3.2: Two levels of the BIP graph hierarchy.

In Fig. 3.2 the two virtual vertices n_5 and n_8 are linked, generating the virtual vertex m_3.

- Virtual parent search: In this stage each virtual orphan vertex of G_l searches for the most similar non-orphan virtual vertex in its vicinity. Among the set of candidates the studied vertex is linked with the parent of the most similar to it. An example of this is showed in Fig. 3.2 where the virtual vertex n_{11} is linked with m_3. It must be noted that this stage does not generate any new virtual vertex.

- Intra-level edges generation: when all virtual vertices at level $l+1$ have been generated, the algorithm computes the intra-level edges. Two virtual vertices n_i and n_j of the graph $G_{l+1}(N, L)$ are connected by an intra-level edge e_k if their corresponding reduction windows w_i and w_j are neighbours in the graph $G_l(N, L)$.

3.3 Evaluation of the BIP capabilities

In order to evaluate the accuracy of the Bounded Irregular Pyramid with respect to others structures, they have been applied in a segmentation task. This task has been chosen to compare their performance due to two main reasons: i) in the proposed approach target representation and segmentation are equivalent tasks with only one difference: target representation is the segmentation of only the desired object and not the segmentation of the whole image and, ii) there are well-known quantitative evaluation methods to measure the quality of segmentation

results. The capability of the proposed pyramid as target representation tool will be qualitatively evaluated in Chapter 4 of this Thesis.

A segmentation algorithm using BIP has been implemented and compared with segmentation algorithms implemented with the main pyramids described in Chapter 2. These algorithms are briefly explained in Appendix B of this Thesis.

3.3.1 Segmentation procedure using BIP

For many years, most of the segmentation methods worked with grey level images due to the large amount of data necessary to process colour images. Recently, colour image segmentation approaches are arising thanks to the increase in computational capability of hardware. Although other image features can be used to segment an image using BIP, i.e. texturae, the evaluation of segmentation results described in this section uses colour as image feature. It must be noted that colour cue image segmentation in a bottom-up way "cannot and should not produce complete final image decomposition into meaningful objects, but it can provide a hierarchical partitioning of the image into homogeneous coloured regions" [58].

In order to segment an image using colour information, this information must be mathematically expressed employing a colour space. In this work, the HSV color space has been selected. The details of this colour space are reviewed in Appendix A of this Thesis. This choice was made because: i) HSV representation is very intuitive and ii) it closely corresponds to the human perception of color. In order to introduce colour information within the BIP, all the vertices of the structure have associated 3 parameters:

- Chromatic phasor, $S_{\angle H}(n)$. The chromatic phasor of a vertex n is equal to the average of the chromatic phasors of the vertices in its reduction window.
- V value or luminosity, $V(n)$. The V value of a vertex n is equal to the average of the V values of the vertices in its reduction window.
- Area, $A(n)$. The area of a vertex is equal to the sum of the areas of the vertices in its reduction window.

The employed similarity measurement between two vertices is the HSV colour distance [65] reviewed in Appendix A, as well as the definitions of chromatic phasor and luminosity.

The similarity threshold used to determine if two vertices are similar is not fixed for all

levels. The mathematical expression of this threshold T is the following:

$$T(l) = T_{max} * \alpha(l) \tag{3.1}$$

being

$$\alpha(l) = \begin{cases} 1 - \frac{l}{L_{reg}} * 0.7 & \text{if } l \leq L_{reg} \\ 0.3 & \text{if } l > L_{reg} \end{cases} \tag{3.2}$$

L_{reg} is the highest level of the regular part of the BIP. This threshold takes into account that usually the receptive field of a vertex in a high level is bigger than the receptive field of a vertex in a low level. Therefore, the linking of two vertices of a high level implies the merging of two larger regions at the base. This threshold makes more difficult the linking process at upper levels and then, the merging of large regions at the base.

The graph $G_0(N, L)$ is a 8-connected graph where the vertices are the pixels of the original image. All the vertices of $G_0(N, L)$ are initialized as follows:

- $Hom(i, j, 0) = 1$. Thus, all the vertices of the base level are vertices of the regular part of the BIP.
- $A(i, j, 0) = 1$.
- The chromatic phasor $S_{\angle H}(i, j, 0)$ of a vertex is equal to the chromatic phasor of its corresponding image pixel.
- The V value $V(i, j, 0)$ of a vertex is equal to the V value of its corresponding image pixel.

The process to build the graph $G_{l+1}(N, L)$ from $G_l(N, L)$ is the following:

1. Regular decimation process. In this step the regular vertices of $G_{l+1}(N, L)$ are built from the regular vertices of $G_l(N, L)$. The inter-level edges that join the regular vertices of $G_{l+1}(N, L)$ with their sons are established. The chromatic phasor, the V value and the area of each regular vertex $(i, j, l + 1)$ are updated, as previously explained, using the values in its reduction window. It must be noted that this reduction window is formed by the four vertices immediately below in G_l.
2. Parent search and intra-level twining. In this step, the parent search and the intra-level twining processes are simultaneously performed. Thus, from each regular orphan vertex

(i, j, l) a search is made for a regular neighbour vertex (i_1, j_1, l) with parent $(i_p, j_p, l + 1)|n_p \in N_{l+1}$ which satisfies the condition (1) and is linked with $(i_p, j_p, l + 1)|n_p \in N_{l+1}$ (*Parent search*). This parent can be a regular $((i_p, j_p, l + 1))$or an irregular vertex ($n_p \in N_{l+1}$) of G_{l+1}. If for the studied vertex a parent is not found, then a search is made for the most similar neighbour regular orphan vertex which satisfies condition (2) in order to generate a virtual vertex in G_{l+1} (*Intra-level twining*). The new inter-level edges are generated. The chromatic phasor, the V value and the area of each regular vertex in G_{l+1} are recomputed. The chromatic phasor, the V value and the area of each virtual vertex in G_{l+1} are computed.

3. Virtual parent search and virtual vertices linking. Each virtual orphan vertex of G_l searches for the most similar virtual vertex with parent in its vicinity with colour distance from it less than T. If a neighbour is found for the studied vertex which satisfies these conditions, then the studied vertex is linked to this parent (*Virtual parent search*). In other case, a search is made for a virtual orphan vertex in G_l which satisfies the condition (3) in order to generate a virtual vertex in G_{l+1} (*Virtual vertices linking*). The new inter-level edges are generated. The chromatic phasor, the V value and the area of each virtual vertex in G_{l+1} are computed. This decimation process to build the irregular part of G_{l+1} is a union-find strategy [141].

4. Intra-level edges generation in G_{l+1}. The vicinity of two regular vertices in G_{l+1} is indicated by their relative position in the bidimensional array corresponding to the regular part of G_{l+1}. Thus, it is not necessary to explicitly generate the intra-level edges between regular vertices. In the case of virtual vertices, the intra-level edges of G_{l+1} must be computed by taking into account the vicinity of their reduction windows in G_l.

The hierarchy stops to grow when is no longer possible to link together any vertices because they are not similar.

In order to perform the segmentation, the orphan vertices are used as roots. The receptive field of each of these vertices is a region of the segmented image. Fig. 3.3 shows some results obtained with the proposed segmentation method.

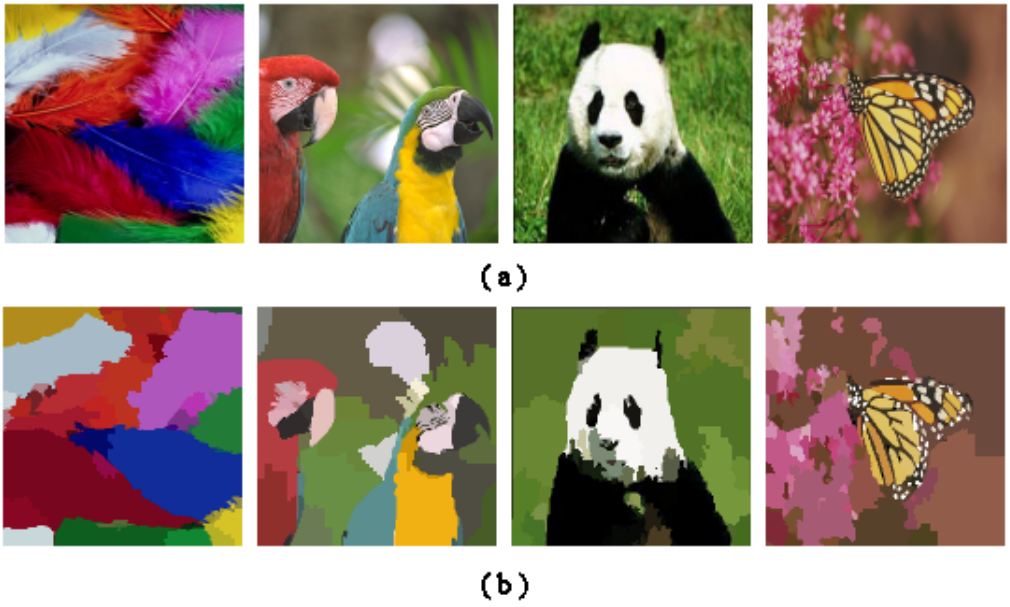

Figure 3.3: a) Original images; b) segmentation results of the proposed Bounded Irregular Pyramid.

3.3.2 Evaluation of segmentation results

3.3.2.1 Evaluation methods

There are two main types of evaluation methods to measure the quality of a given segmentation algorithm: qualitative and quantitative methods. Qualitative methods are based on the opinion of a human expert who decides on the accuracy of the studied algorithm. Although this measure depends on the human intuition and can vary across different observers, it is still very useful to evaluate some characteristics of the algorithms. On the other hand, quantitative methods are based on numerical data. According to the previous work of Zhang [166], quantitative segmentation evaluation methods can be classified into two categories: analytical and empirical methods. Analytical methods directly examine and assess the segmentation algorithms by analyzing their principles and properties. Some properties to be evaluated are the processing strategy, the processing complexity and efficiency and the segmentation resolution. These properties can aid in selecting suitable algorithms in particular applications. But usually, the segmentation results are used in more complex image processing or computer vision tasks, where the accuracy of the results is usually more important than the performance of the algorithm, which can be improved later. Hence, the empirical methods are preferred. These methods indirectly judge the segmen-

tation algorithms by applying them to test images and measuring the quality of segmentation results.

Quantitative empirical methods can be classified into two types: goodness methods and discrepancy methods. Goodness methods measure some desirable properties of segmented images by goodness parameters. These methods have the problem that these parameters depend on the human intuition. Discrepancy methods compute the ideal segmentation first and then the segmentation obtained with the algorithm is compared with the ideal one by counting differences. These methods present the problem that having a previous ideal segmentation is necessary, which depends on the human intuition too.

In this work, three empirical methods have been chosen: the Shift Variance proposed by Prewer and Kitchen [115], the F function proposed in [89] and the Q function proposed in [15]. These methods can be regarded as goodness methods, but they do not require any user-set parameter for the evaluation of the performance of the segmentation. The smaller the value of these parameters, the better the segmentation result.

The F function takes into account the following goodness indicators:

- Regions must be uniform and homogeneous according with the similarity criterium employed to perform the segmentation, i.e. colour.
- The interior of the regions must be simple, without too many small holes.
- Adjacent regions must present significantly different values for uniform characteristics.

Given a segmented image I, the F function is computed as follows:

$$F(I) = \frac{1}{1000(N \cdot M)} \sqrt{R} \sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}} \tag{3.3}$$

being NxM the image size and R the number of segmented regions. A_i and e_i are the area of the region i and its average colour error, respectively.

The Q function takes into account the same indicators, but penalizes the existence of small regions in a more rigid way.

$$Q(I) = \frac{1}{1000(N \cdot M)\sqrt{R}\sum_{i=1}^{R}[\frac{e_i^2}{1+logA_i} + (\frac{R(A_i)}{A_i})^2]} \tag{3.4}$$

being $R(A_i)$ the number of segmented regions with area equal to A_i.

Shift variance refers to the fact that the segmentation produced by pyramidal segmentation algorithms varies when the base of the pyramid is shifted slightly. This is an undesirable effect for a segmentation method. Thus, the Shift Variance (SV) can be taken as a measurement of an algorithm quality. The F and Q functions compare an original image with its segmented image. In contrast, this method compares the segmentation of an image by a given algorithm with the segmentation produced by the same algorithm on slightly shifted versions of the same image. To do that, a 128x128 pixel window from the center of the original image has been taken. This subimage has been compared with each segmented image obtained by shifting the window a maximum shift of 11 pixels to the right and 11 pixels down. Thus, there is a total of 120 images to compare with the original one. In order to do each comparison between a segmented shifted image j and the segmented original one, the root mean square difference is calculated:

$$RMSD_j = \sqrt{\frac{\sum d_i^2}{128 \cdot 128}} \qquad SV = \frac{1}{120} \sum_{j=1}^{120} RMSD_j \tag{3.5}$$

being d_i the pixel-to-pixel colour difference between the segmented images.

3.3.2.2 Comparative study

In order to compare the BIP with the main regular and irregular pyramids present in the literature, two segmentation algorithms based on regular pyramids have been implemented: the linked pyramid proposed by Burt *et al.* [24] (LRP), and the weighted linked pyramid with possibilistic linking (WRP). The weighted linked pyramid has been slightly modified to include a root extraction process that avoids the need to choose a working level. Vertices that link only weakly to all their parents have been selected as root vertices. Unforced linking [3] has been used in the linked pyramid to select region roots at different pyramid levels. Comparisons with five segmentation algorithms based on irregular pyramids have been also included: the classical RAG hierarchy employed by Bertolino and Montanvert [10] (ClIP), the localized pyramid [63] (LIP); the segmentation algorithm proposed by Lallich *et al.* [86] (MIP), the hierarchy of image partitions by dual graph contraction [82, 58] (HIP) and the hierarchical segmentation algorithm based on combinatorial pyramids proposed by Brun and Kropatsch [19] (CoIP). The two regular pyramid-based algorithms and the bounded irregular pyramid employ the HSV colour distance to perform the segmentation. In this Thesis, the algorithm proposed by Lallich et *al.* [86] has been modified to deal with HSV colour images. All these segmentation approaches based on pyramids are briefly described in Appendix B of this Thesis.

Two of the main drawbacks of the regular pyramids were qualitatively evaluated by Bister

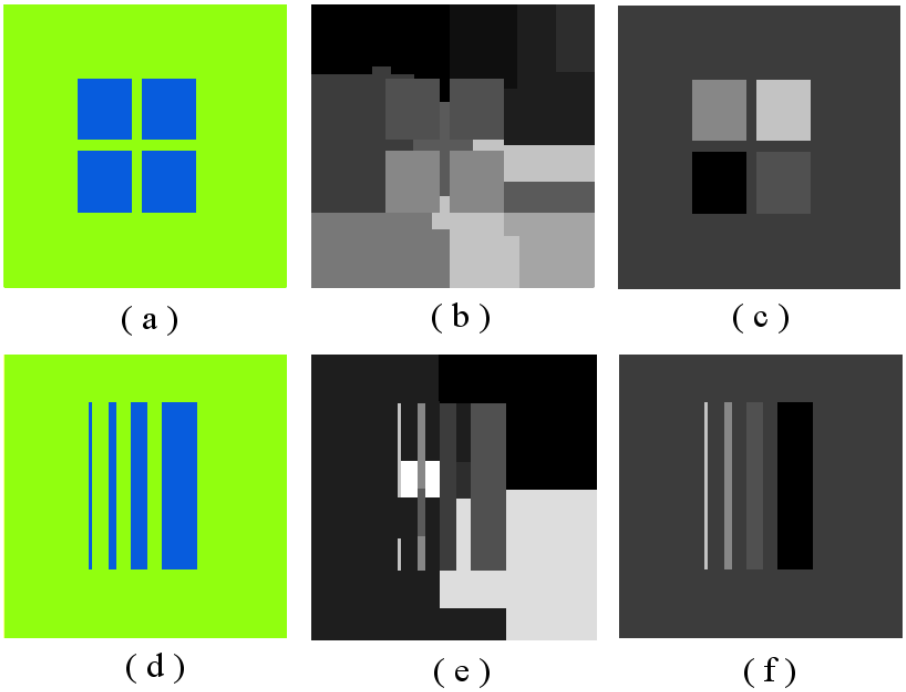

Figure 3.4: Qualitative evaluation of regular pyramid drawbacks: a) input image #1; b) linked pyramid segmentation result of a); c) BIP segmentation result of a); d) input image #2; e) linked pyramid segmentation result of d); and f) BIP segmentation result of d).

et *al*. [12]. Region connectivity is not preserved in regular pyramids because the structure does not take into account adjacency information when the pyramid is built. Figs. 3.4.b) and 3.4.c) represent the different classes resulting from the segmentation of the image in Fig. 3.4.a) using the LRP and the BIP algorithms, respectively. False colour has been used to distinguish each class from the rest. Fig. 3.4.b) shows that the linked pyramid divides up the background region into different classes. Besides, it fuses different regions into the same class, creating non-connected segmented regions. In contrast, the bounded irregular pyramid correctly segments the original image into five classes: four rectangles and the background (Fig. 3.4.c)). The second drawback of regular pyramids is related with the presence of elongated objects. The inflexibility of the structure of regular pyramids makes the adaptation of such a structure to this type of objects difficult. Fig. 3.4.d) includes a set of elongated objects presenting different aspect ratios. It is easy to note that the linked pyramid (Fig. 3.4.e)) cannot handle elongated shapes. Fig. 3.4.f) shows that the bounded irregular pyramid is capable of adapting its structure to correctly segment this type of objects.

In order to quantitatively evaluate the efficiency of the different segmentation algorithms, 30 colour images from Waterloo and Coil 100 databases have been chosen. All these images have

been resized to 256x256 pixels. A 3GHz Pentium IV PC, i.e. a sequential processor, has been employed. Algorithms proposed by Lallich et *al.* [86] and Haxhimusa and Kropatsch [58] are based on decimation procedures that have been mainly designed for parallel computing. Therefore, they do not efficiently run in this sequential computer. However, the proposed algorithm and the CoIP [19] are based on decimation techniques more suited to sequential computing. Specifically, the union-find process has proven to be very efficient when run on sequential machines. Although it employs a decimation kernel designed for parallel computing, another fast algorithm is the LIP which only processes a part of the image. In this case, the computational time associated to the local homogeneity analysis has been taken into account.

The processing times are shown in Table 3.1. The fastest algorithms are the BIP and the algorithms based on regular pyramids. BIP is faster than irregular approaches because a large part of the image is processed following a classical regular pyramid approach. Besides, it is faster than regular algorithms because it does not have relinking process. The interlevel edges are computed in only one pass. In these experiments, iterative relinking of regular structures has been bounded to a maximum value of 10 iterations per level.

Table 3.1 also presents the maximum height associated to the hierarchical representation employed to perform the segmentation. The vertices of the pyramid level associated to this height define the segmentation in the ClIP, LIP, HIP, MIP and CoIP algorithms. In the rest of algorithms, roots can be defined in different levels of the hierarchy. In any case, it must be noted that this height does not correspond to the appex of the hierarchical representation, i.e. the pyramid level that only contains one vertex. According to the obtained data (Table 3.1), it can be appreciated that the two regular representations and the BIP and HIP irregular pyramids present the minimum heights. On the contrary, the CoIP and the MIP irregular pyramids present the maximum height values. The BIP is the irregular pyramid with minimum height because its regular part avoid a high increase of the hierarchy.

Finally, Table 3.1 also shows the number of regions obtained by the different segmentation algorithms. It can be noted that the different values are very similar.

Figs. 3.5 and 3.6 show five image tests used in the experiments and the results obtained from all compared segmentation algorithms. Before quantitatively comparing the different methods, some explanations about them are required:

- The selection of the parameters of all algorithms has been conducted to obtain the best results according to the Q function.

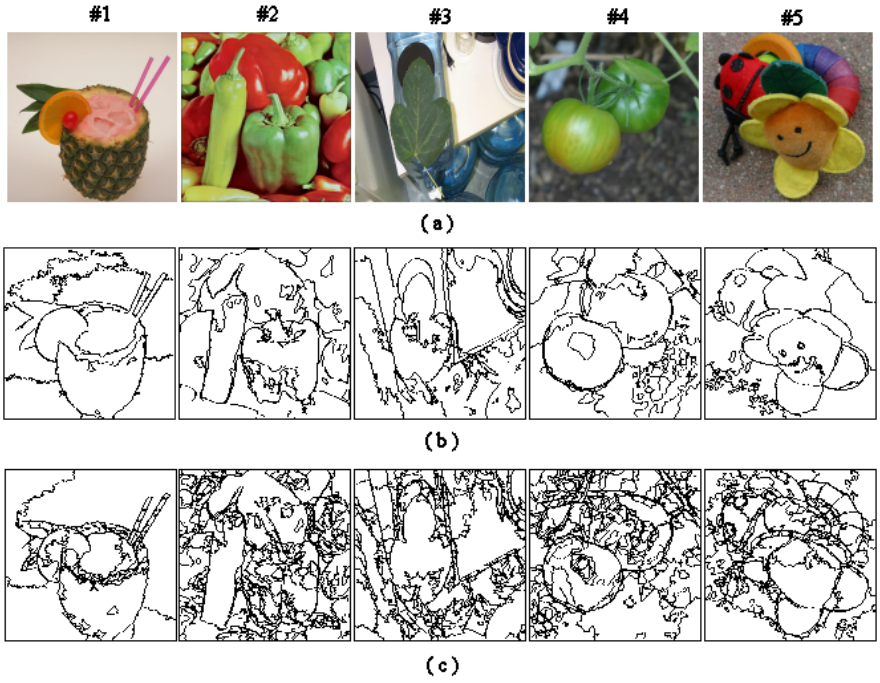

Figure 3.5: a) Input images; b) segmentation images using the linked pyramid; c) segmentation images using the weighted linked pyramid.

- Only connected regions have been considered. For the regular pyramids, unconnected regions have been split into several smaller regions.
- In CoIP method the background region growing has been limited because this produces worse results.
- In MIP, the test based on Moran's spatial autocorrelation coefficient is used to control the decimation process. Outliers are extracted in the distribution of regions merging candidates. This outlier detection results in a more detailed segmentation with more small regions. The F and Q functions penalize the existence of small regions. Therefore, the threshold which controls the outlier detection has been set to a high value (the 20 % tail of the distribution of error), to reduce the number of outliers.
- In order to reduce the number of small regions, several methods employ a threshold. In the experiments made in this Thesis, this threshold has been set to 20 pixels in all algorithms (in the CoIP framework, this cleaning procedure implies that the hierarchy presents additional levels).
- The result associated to the HIP is a hierarchy of partitions with multiple resolutions that is performed until the pyramid appex is reached. Although this hierarchy is suitable for

	Processing times (sec)			Hierarchy height			Number of regions		
	t_{min}	t_{ave}	t_{max}	h_{min}	h_{ave}	h_{max}	NR_{min}	NR_{ave}	NR_{max}
LRP	0.94	1.37	1.81	9	9	9	17	81.6	203
WRP	0.31	0.40	0.58	9	9	9	19	79.7	148
ClIP	2.51	3.96	7.68	17	36.7	72	9	84.1	210
LIP	1.71	2.78	6.13	8	25.4	51	12	73.8	210
MIP	2.43	3.47	4.47	13	33.3	62	45	107.7	201
BIP	0.14	0.17	0.39	8	8.8	15	8	83.5	229
HIP	4.07	4.29	4.91	10	11.6	18	23	76.2	149
CoIP	1.32	2.88	12.8	9	74.4	202	25	91.6	238

Table 3.1: Processing times, height of the hierarchy employed by the segmentation algorithm and number of obtained regions. Average values have been obtained from 30 different images.

further analysis, a hierarchy level must be selected in order to obtain an unique image segmentation. In this case, the level that provides the better Q has been chosen.

Table 3.2 presents the comparison measurements among methods. This table shows that all irregular pyramids obtain better segmentation results than regular ones. It can be also noted that the MIP and the CoIP present the best global results. The behaviour of the MIP is excellent, although it is the method that provides the highest number of obtained segmentation regions. In contrast, the BIP and the LIP obtain the lowest number of regions. When compared to the results provided by the ClIP, the LIP obtains less regions but with better performance in terms of the F and Q functions. The results obtained by the BIP are very similar to the ones obtained by the ClIP or the LIP. Fig. 3.6.e) shows that the HIP algorithm preserves details in low-variability regions (in this case, the background of the image). Image smoothing in low variability regions would solve this problem [58]. In any case, this method provides perceptually important partitions in a bottom-up way based only on local differences. The height of the hierarchy is one of the lowest among the irregular approaches (in fact, only the BIP has a lower height), so it is specially suitable to describe the image structure. Finally, it can be noted that the SV measure is high in the regular pyramids and in the BIP approach, due to the regular-based reduction of great part of the image. In the rest of irregular pyramids, SV measures are very similar.

Finally, in this section, the irregular pyramids are briefly qualitatively evaluated. To do that the results shown in Fig. 3.6 have been used. The first aspect which is important to point out is that the experiments have been conducted in order to obtain the best Q values. Therefore, the obtained segmentation results not always present the best partition of the original image in homogeneous coloured regions. It is the case of the HIP, where the level selected to

	F			Q			SV		
	F_{min}	F_{ave}	F_{max}	Q_{min}	Q_{ave}	Q_{max}	SV_{min}	SV_{ave}	SV_{max}
LRP	765.8	1070.4	1515.5	1052.1	1524.9	2105.4	37.8	66.9	83.5
WRP	791.2	1072.8	1428.2	1133.7	1480.6	2034.2	49.6	69.9	98.5
ClIP	329.3	840.2	1290.0	479.1	1062.7	1590.3	18.0	28.8	42.8
LIP	213.6	746.1	1345.6	489.4	1002.5	1327.4	20.8	31.7	46.7
MIP	290.4	646.6	1043.7	360.5	817.6	1292.5	19.3	30.1	42.4
BIP	198.6	711.7	1556.1	339.4	1086.7	1919.8	26.4	44.1	84.5
HIP	201.7	689.2	1201.6	458.3	957.8	1521.5	18.5	27.1	35.9
CoIP	234.3	618.8	934.9	415.5	878.5	1294.5	21.3	30.7	42.8

Table 3.2: F, Q and Shift Variance values. Average values have been obtained from 30 different images.

perform the segmentation originates good results in the Q value but the partition of the image is not clear. Although it seems to have more segmented regions than the other approaches, the number of regions is similar. It is because the obtained receptive fields have similar size without small segmented regions. Therefore, the HIP has an oversegmentation problem in homogeneous regions. This problem does not appear if an upper level is used to generate the segmentation.

The BIP also presents receptive fields with similar size. However, it can be noted that the oversegmentation problem is less important in the BIP than in the HIP. In contrast, the BIP tends to produce square-shaped regions in homogeneous areas of the image due to the 4-to-1 regular segmentation procedure.

Among the simple graph based methods, the best results are obtained by the MIP. Besides, the ClIP and the MIP present the best segmentation results in the background of the images. The results obtained with the LIP in no-homogeneous regions are similar to the obtained ones with the ClIP. The problem of the LIP is that if two similar coloured regions are considered by the algorithm as different homogeneous regions, they are not merge together. In the case of the CoIP, if a good Q value is obtained, the segmented image presents receptive fields of very different sizes: very big receptive fields and small ones in the same image.

After studying the previously commented results, it should be noted that, although the BIP does not have the best results, it has similar performance than other irregular approaches with a ten times smaller computational time. This time reduction is very important to achieve the goal of this Thesis: the development of a real time tracking system, because the BIP is used as target representation structure. To do that, as will be explained in Chapter 4, the target to track is segmented using BIP. This segmentation originates a hierarchical representation of the target and the template, which are exploited to perform the template matching process

in a hierarchical way, reducing the computational cost of this process. Therefore, the lower the computational cost of the segmentation process, the lower the computational time of the tracking procedure.

3.4 Summary

In this chapter the BIP structure has been detailed explained and compared with the main regular and irregular pyramids present in the literature. The BIP arose due to the necessity of get an irregular pyramid with similar accurate segmentation results than other irregular pyramids but faster to build and traverse. This reduction time is necessary because the BIP is the tool to build the representations of the target and the template in the proposed real time tracking approach. The first step to generate these representations, as will be detailed explained in Chapter 4, is to segment the region of the input image where the target is likely placed. Therefore, is very important that this segmentation be accurate and as quick as possible. At the same time, the BIP is used to perform the template matching in a hierarchical way. Therefore, the lower the time to traverse the BIP, the lower the time to perform the template matching.

The key idea behind the BIP is to use a $2x2/4$ regular structure in the homogeneous regions of the input image and a simple graph irregular structure in the rest of regions. The irregular part of the BIP permits to avoid the problems of regular structures and its regular part reduces its computational complexity.

In the results section of this chapter, the BIP has proven to achieve similar segmentation results than the other irregular structures but reducing at least ten times the computational time. The competence of the BIP to the proposed tracking approach is corroborated in Chapter 4.

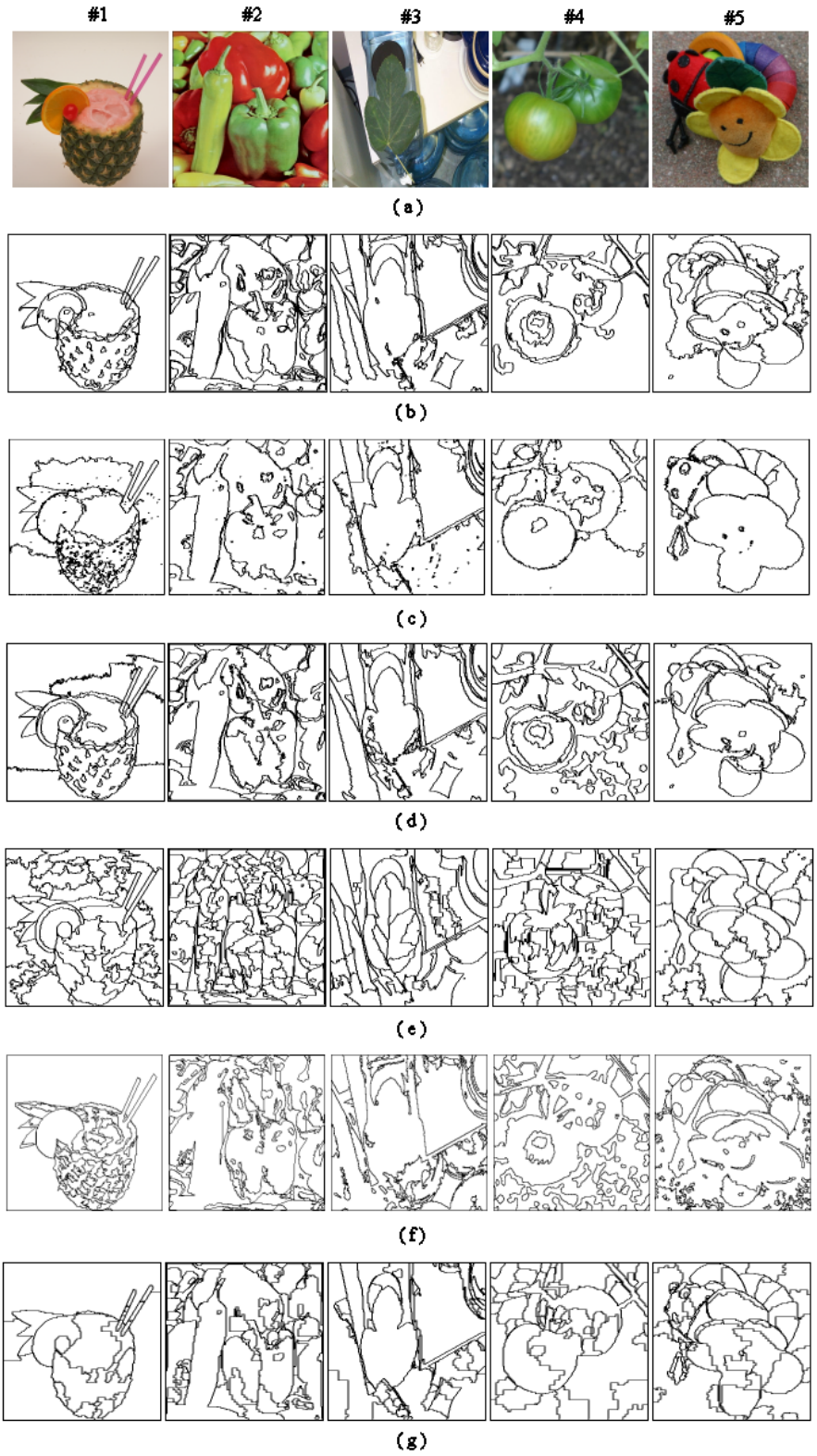

Figure 3.6: Segmentation results; a) input images; b) classical RAG hierarchy; c) Lallich et *al.* [86] proposal; d) localized pyramid; e) hierarchy of image partitions; f) combinatorial pyramid; g) BIP.

Chapter 4

Tracking algorithm

The goal of a tracking algorithm is to recognize or estimate the motion and the position of a desired object. The 2D projection of this object in the image is called target. In general, tracking algorithms can be divided in two main components:

- Target representation and localization.
- Filtering and data association.

Target representation is the way that the information about the desired tracked object is manipulated and stored in order to localize the target in each frame of the sequence. The chosen target representation approach determines the process to localize the target. For this reason, the target representation and the localization process are both included in the same component of the tracker.

Filtering is the process to predict the position of the tracked object in the current frame taking into account the past behaviours of the object and the system. It is related with the dynamics of the tracked object. This process is particularly useful when the localization process is slow, because to have an estimation of the location of the tracked object can help in the reduction of the search image area. This reduction allows to make faster the tracking system. Data association techniques try to solve the problem of measurement association when there are several objects to track. That is, how to select the real positions of the targets among the set of likely positions (measurements).

The way in which both parts are combined and balanced depends on the final application. If, for example, the goal is to track a complex object (i.e. a face) with complex dynamics in a crowded scene, the emphasis is put in the target representation and localization part of the

tracking because the appearance of the object is more important than its movement. On the other hand, if the goal is, for example, to track objects in movement in a surveillance application, the motion of the objects are the most important features. In those cases, the emphasis is put in the filtering and data association component of the tracking. Chapter 2 reviewed examples for both approaches.

The goal of this Thesis is to present a general purpose tracking system which can track rigid and non-rigid objects in cluttered sequences. The proposed tracking system only uses the target representation and localization component, showing that if the target representation is accurate and the localization process is fast enough, it is not necessary to use any filtering stage to obtain accurate results.

In this Thesis, a new approach for target representation and localization is presented. This approach addresses two of the most important causes of failure in object tracking: changes of object appearance and occlusions. The proposed target representation method is a hierarchical template-based appearance model which uses the Bounded Irregular Pyramid (see Chapter 3). The localization process is a hierarchical template matching approach. The proposed tracking system allows to track non-rigid objects in real-time by employing a weighted template which is dynamically updated and a hierarchical framework that integrates all the components of the tracker. This weighted template and the way it is updated also allow the algorithm to successfully handle partial and total occlusions of the tracked object. In addition, the proposed hierarchical tracker allows tracking of multiple objects with low increase of computational time.

In this chapter, Section 1 makes a brief summary of the main advantages and drawbacks of the target representation methods which were previously explained in detail in Chapter 2. Besides, the selection of a template-based target model is justified. Section 2 describes the target and template representation. Section 3 presents the hierarchical tracking algorithm for one object. Section 4 explains the tracking algorithm for multiple objects. Section 5 shows experimental results and, finally, Section 6 gives some conclusions.

4.1 Introduction

Chapter 2 of this Thesis reviewed the five main approaches to target representation: model-based, appearance-based, contour- and mesh-based, feature-based and hybrid methods [27]. Model-based tracking approaches [78] employ a priori knowledge about the geometry of objects in a given scene. This is a disadvantage in itself, as models for all objects that need to be

tracked are required. Apart from this lack of generality, often detailed geometry is required for the models, which results in a high computational cost. Appearance-based methods [70] track connected regions that roughly correspond to the 2D shapes of the objects based on their dynamic model. The tracking strategy relies on information provided by the entire region. Examples of such information are motion, colour and texture. These methods cannot usually cope with complex deformations of the tracked object. Contour-based methods [14] track only the contour of the object. Usually they use active contour models like snakes, B-splines or geodesic active contours. Feature-based approaches [149] use features of an object to track parts of it. Although these approaches are very stable even in case of partial occlusions, they require a means to group the features that belong to the same object. The last group of tracking approaches is designed as a hybrid between a region-based and a feature-based technique [27]. They exploit the advantages of the two by considering first the object as an entity and then by tracking its parts. The main drawback of these approaches is their high computational complexity.

This Thesis is concerned with tracking objects in image sequences using a template-based appearance model. The aim is robust real-time tracking under severe changes of viewpoint in the absence of an a priori model. Appearance models can be divided in [70]: template-based, view-based, global statistic based and motion-based methods. View-based models, usually learned with Principal Component Analysis, have the advantage of modeling variations in pose and illumination. However they also have the disadvantages of being object specific and requiring training prior to tracking in order to learn the subspace basis. Motion-based models usually have problems when motions of the target and background are similar. They are usually improved by accumulating an appearance model through time or estimating both motion and appearance simultaneously. These methods are computationally expensive. The use of local and global image statistics, such as color histograms, have been popular for tracking. Colour distribution can provide an efficient feature for tracking as it is robust to partial occlusion, scaling and object deformation. It is also relatively stable under rotation in depth in certain cases [106]. Therefore, colour distributions have been used to track non-rigid objects like heads [96] or hands [94]. A variety of statistical techniques have been used to model the colour distribution [42]. Thus, Raja *et al*. [96] modelled the colour distribution of an object using a mixture of Gaussians fitted using the EM (Expectation Maximization) algorithm. A difficulty of this parametric technique is how to choose the right number of Gaussians for the assumed model. To avoid this problem, nonparametric techniques using histograms can be used. Although colour histograms is not the best nonparametric density estimate [127], it has been successfully used to track hands [94] or other non-rigid objects against cluttered backgrounds [34]. Besides, colour histograms can be

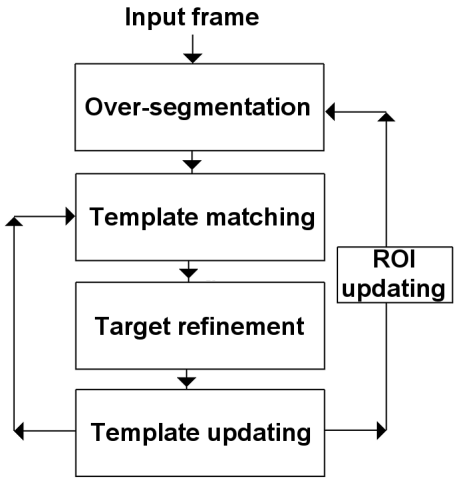

Figure 4.1: Illustration of the tracking algorithm.

easily quantized into a small number of bins to satisfy the low-computational cost requirements of real-time processing. One of the main drawbacks with colour histograms is that, if only spectral information is used to characterize the target, the similarity function can have large variations for adjacent locations on the image lattice and the spatial information is lost. To find the maxima of such functions, an expensive exhaustive search must be applied [34]. In order to avoid it, the similarity function can be regularized by masking the objects with an isotropic kernel in the spatial domain [42]. Template-based models can be seen as a way to combine colour information with spatial information. As previously discussed in Chapter 1 of this Thesis, a robust template-based approach should: i) update the template to accommodate the change of object appearance and, ii) detect an occlusion and recapture the object when the occlusion ends. In order to achieve these two goals, the algorithm proposed in this Thesis uses a hierarchical template-based model which is built using a Bounded Irregular Pyramid (BIP). This model allows tracking of non-rigid objects and handles occlusions by employing a weighted template which is dynamically updated. The template matching process is hierarchically performed by integrating it in the same hierarchical structure where the template is represented.

The remaining of this chapter explains in detail the proposed tracking algorithm to track a single object and several objects at the same time.

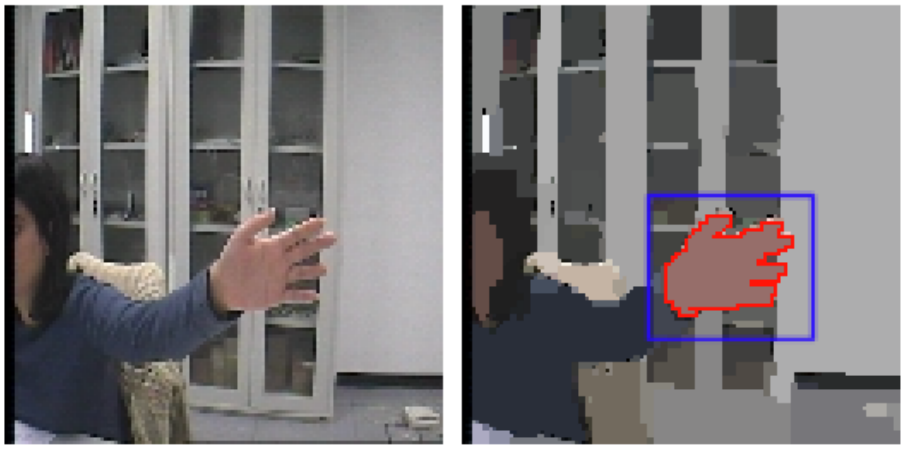

Figure 4.2: a) Original image; b) segmented image with the chosen target marked in red and the ROI marked in blue.

4.2 Single object tracking

In this section the description of the algorithm to track a single object is presented. The algorithm works in four consecutive stages (Fig. 4.1):

1. *Hierarchical representation of the Region of Interest (ROI)*: the ROI is the input image region where it is more likely that the target will be. In this process a BIP is built over the ROI as previously explained in Chapter 3.

2. *Template matching procedure*: the target is searched by means of a hierarchical template matching procedure.

3. *Refinement of the target appearance*: the target representation is completed using information from the BIP built over the ROI.

4. *Template updating*: the template is a weighted template which is dynamically updated in order to follow up the viewpoint and appearance changes of the object to track.

4.2.1 Starting the tracking

The target to track is chosen manually from the first frame of the video sequence. For this, the colour segmentation algorithm using BIP (Chapter 3) is applied. The target can be chosen to be any of the segmented regions. In Fig. 4.2.b), the segmentation of a real scene is showed. In this

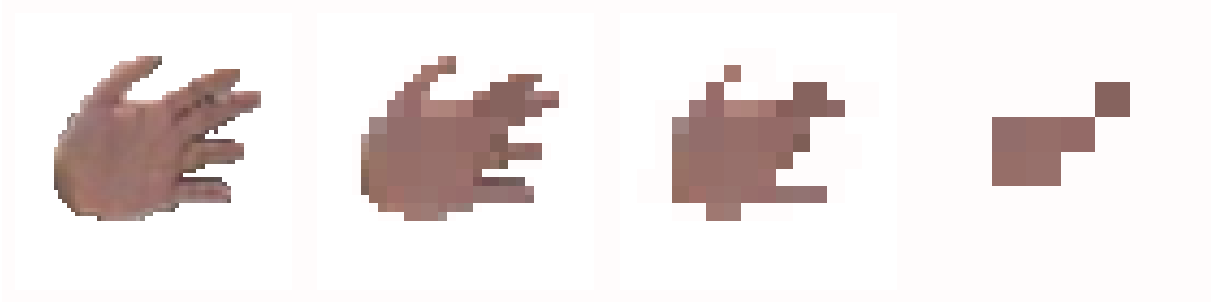

Figure 4.3: Template hierarchical representation of the hand extracted from Fig. 4.2.

Figure 4.4: Template hierarchical representation of a face.

case the hand has been selected as the target to track. Once the target is chosen, the algorithm extracts its hierarchical representation from the BIP computed in the segmentation of the first frame. The regular part of this hierarchical structure is the first template and a rectangular region centered on its centroid is the first region of interest (ROI). The size of this rectangular region depends on a parameter of the algorithm (ϵ) which will be explained in Section 4.2.6. Fig. 4.2.b) shows the ROI (marked in blue) corresponding to the selected target. The hierarchical representation of this hand, which is the first template in this example, is shown in Fig. 4.3. Another example of first template, corresponding to a face, is shown in Fig. 4.4.

To recapitulate, the hierarchical template is computed by segmenting the original image and using the regular part of the manually selected segmented region. The reasons to use only this regular part and not the whole structure are explained in Section 4.2.3. It should be appreciated that, although only the regular part of the BIP is used, the whole BIP is built to initialize the tracking because this regular part is very influenced by the irregular one. Fig. 4.5 shows the segmented image corresponding to the original one of Fig. 4.2.a) using only the regular part (regular decimation process and regular parent search) of the segmentation algorithm. It should be noted that the hand is segmented in 6 different regions. This does not permit to select the whole hand as the target and the template. If the largest region (region marked in red) is

Figure 4.5: Segmented image using only the regular part of the BIP segmentation algorithm.

Figure 4.6: Template hierarchical representation of the hand extracted from Fig. 4.5.

selected as the target to track, the resulting template representation would be the one shown in Fig. 4.6.

It must be noted that any segmentation process could be used to determine the region to track from the original image. The difference would be in the process to build the hierarchical representation of that region. If the BIP-based segmentation algorithm is used, this representation is directly computed during the segmentation process. If a different segmentation approach is used, the hierarchical representation of the target must be computed after the segmentation is completed. This is done by initializing level 0 of the BIP as follows: the only homogeneous vertices of the level 0 of the BIP are the vertices of the selected target, the rest of vertices are non-homogeneous ones. Then the BIP is built.

The five main modules of the proposed tracking system are explained in the following sections.

Figure 4.7: Over-segmentation of a ROI.

4.2.2 Over-segmentation

The first step of the tracking process is to obtain a hierarchical representation of the region of interest ($ROI^{(t)}$) in the current frame t. $ROI^{(t)}$ depends on the target position in the previous frame, being updated as described in Section 4.2.6. The hierarchical structure is built by segmenting the ROI using the segmentation process explained in Chapter 3. This ROI can be represented in each level as:

$$ROI^{(t)}(l) = \bigcup_k p_k^{(t)}(l) \tag{4.1}$$

being $p_k^{(t)}(l)$ a vertex (regular or virtual) of the level l of the Bounded Irregular Pyramid built over the ROI at frame t. The maximum colour similarity threshold T_{max} used in this segmentation process should be small enough to allow an over-segmentation of the ROI. This is a segmentation in which the number of obtained regions is very high compared with the number of real regions in the ROI. This over-segmentation avoids a high dependency of the tracking method with the segmentation results. Thus, in this process the ROI is divided up in a set of segmented regions. Each regular and irregular node of $ROI^{(t)}$ belongs to one of these regions, independently of its level. Therefore each segmented region is a hierarchical estructure formed by a set of nodes of $ROI^{(t)}$.

Fig. 4.7 shows the over-segmentation of the frame #10 of the hand sequence. In this case the ROI was oversegmented in 394 different regions with $T_{max} = 10$.

4.2.3 Template matching

In order to reduce the computational load associated with a template matching process, the Bounded Irregular Pyramid has been selected in this Thesis to represent the target and the

template. Thus, in the proposed system, each target T and template M are represented using BIP structures:

$$M^{(t)}(l) = \bigcup_k m_k^{(t)}(l) \tag{4.2}$$

$$T^{(t)}(l) = \bigcup_k q_k^{(t)}(l) \tag{4.3}$$

being $M^{(t)}(l)$ and $T^{(t)}(l)$ the level l of the pyramidal structures corresponding to the template and the target in the frame t respectively. Each level of the template is made of a set of vertices $m_k^{(t)}$. Equivalently, each level of the target is made of a set of vertices $q_k^{(t)}$. While the target representation is composed by the regular and irregular vertices of the BIP, the template representation has only regular vertices. The use of only a regular representation of the template allows to reduce the computational complexity of the process because of the well-known an easily computable neighbourhood relationships between vertices. The regular part of the BIP can be expressed as a hierarchy of bidimensional image arrays where two vertices are neighbours if they are placed in adjacent positions of one of such arrays. Thus, the template matching process is a comparative process between images. If the whole structure is used in the matching process, the matching process will be a comparative procedure between graphs, which is more complex and computationally expensive.

The target and template representations are segmentations of the template and the target, respectively. The employed segmentation process is slightly different to the previously explained one in Chapter 3. It is integrated inside of the tracking process and it is related to the results of the template matching procedure.

After the hierarchical representation of $ROI^{(t)}$ has been obtained, the algorithm looks for the target $T^{(t)}$ using a hierarchical template matching approach. In this template matching process only the regular part of the BIP structures corresponding to the ROI and the template are used. In this section, and in order to simplify the nomenclature, the regular part of the target, the template and the ROI hierarchical representations are referred as the target, the template and the ROI, respectively.

The localization of $T^{(t)}$ consists of the following steps:

1. *Working level selection.* Although the template matching process could be accomplished in any level of the pyramid, the algorithm uses as *working level* $l_w^{(t)}$, at the current frame

t, the higher level where this matching can be correctly achieved. This allows to reduce as much as possible the computational cost of the whole process. $l_w^{(t)}$ is defined as the highest level of the template representation that satisfies the following condition:

$$100 * \sum_{ij \in M^{(t)}(l_w)} A(i,j,l_w) \; / \sum_{ij \in M^{(t)}(0)} A(i,j,0) > T_A \tag{4.4}$$

That is, l_w is the highest level whose template area is at least a T_A% of the total area of the template.

It must be noted that the working level value depends on the size and the shape of the template. However, this is not a critical parameter of the algorithm. Only if the tracked object is a thin elongated object, the working level is level 0.

2. *Target localization.* The process to localize the target in the current frame t is a top-down process which starts at the working level $l_w^{(t)}$ and stops at the level where the target is found. In each level l, the template $M^{(t)}(l)$ 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. In each displacement of the template over the ROI, the corresponding vertices are compared computing an overlap value. If there is a match between a vertex of the template and a vertex of the ROI, the overlap is incremented in a value equal to the weight (Section 4.2.5) of the vertex of the template. In the experiments, it has been considered that the target is found in a position if the overlap in that position is higher than 70%. All the ROI vertices that match with vertices of the template are marked as vertices of the target in the whole structure $ROI^{(t)}$. Thus, the regular part of the hierarchical representation of the target $T^{(t)}$ is obtained. The overlap for each template displacement can be expressed as:

$$overlap = \sum_{ij \in \xi} w^{(t)}(m(i,j,l)) \tag{4.5}$$

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 4.2.5. ξ is the subset of vertices of the template that match with vertices of the ROI at level l. A vertex of the template $m(i,j,l)$ matches with a vertex of the ROI $p(i,j,l)$ if their colour similarity is less than a threshold T_C:

$$g(f_a(m), p) < T_C \tag{4.6}$$

being $g()$ a colour distance and $f_a(m)$ the displacement function which shifts the template over the ROI. Although other transformation such as rotations or scale changes could be

modelled using f_a, translation has demonstrated to be sufficient to correctly perform the tracking process, as will be shown in the results section of this chapter. Other transformations such as scale changes, rotations or deformations of the object are handled by the algorithm thanks to the target refinement process and the way the template is updated.

This displacement function f_a of the template can be represented as a set of displacements d_k of each vertex coordinate: $d_k^{(t)} = (d_k^{(t)}(i), d_k^{(t)}(j))$, 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 $d_0^{(t)} = d_f^{(t-1)}$.

4.2.4 Target refinement

In order to refine the target appearance, its hierarchical representation is rearranged level by level following a top-down scheme. At this point it might be helpful to recall some previously explained concepts. In the over-segmentation step, the ROI was segmented and $ROI^{(t)}(l)$ was obtained. In this segmentation process the ROI was divided up in a set of segmented regions R_i. In this subsection, the segmented region in which a vertex n_k is included will be denoted as $R(n_k)$.

In the template updating, the base of the target representation was obtained. This base was formed by a set of regular vertices of the hierarchical representation of the ROI. These vertices are members of segmented regions of $ROI^{(t)}$. In a first stage of the target refinement step, all the vertices (regular and virtual) of the segmented regions which have some vertices in the target representation are automatically marked as vertices of the target. In a second stage of the target refinement process, the target is more detailedly refined. The process is explained below:

For each regular vertex $p^{(t)}(i,j,l)$ of the ROI marked as vertex of the target ($p^{(t)}(i,j,l) = q^{(t)}(i,j,l) \in T^{(t)}(l)$) a search is performed among its irregular and regular neighbours $n_k \in \xi_{q^{(t)}(i,j,l)}$. Being $\xi_{q^{(t)}(i,j,l)}$ the vicinity of $q^{(t)}(i,j,l)$. The colour of each of these neighbours n_k which does not belong to the target is compared with the colour of $q^{(t)}(i,j,l)$. If their colour similarity is less than a threshold T_r then all the vertices $n_s \in R(n_k)$ are marked as target vertices. Thus, the hierarchical representation of $T^{(t)}$ is completed.

Fig. 4.8.b) shows the vertices at the base level of the target representation before refinement. Fig. 4.8.c) shows how the representation of the target is completed by the refinement process. The improvement is possible thanks to the irregular part of the BIP representation of

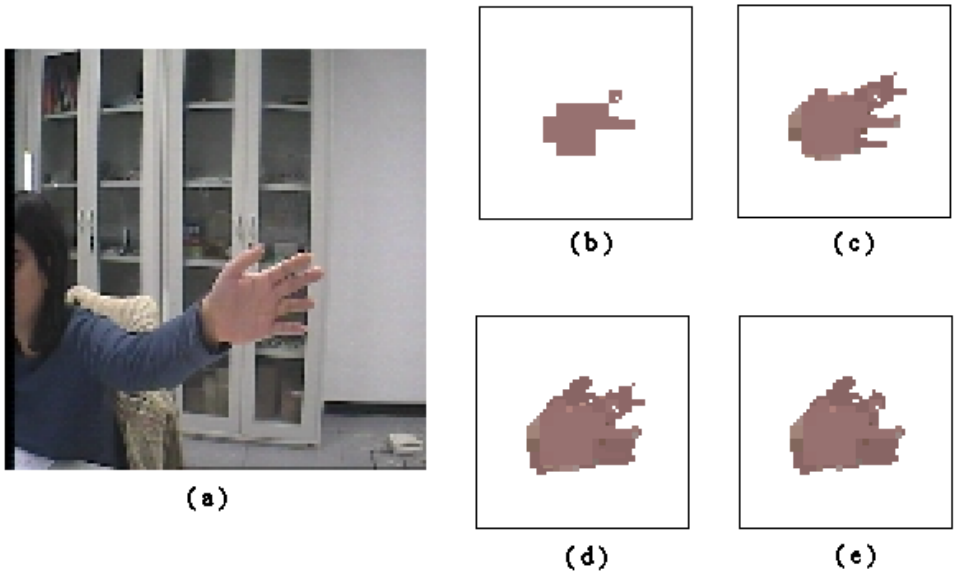

Figure 4.8: a) Frame 2 of the hand sequence; b) level 0 of the target representation before the target refinement step; c) level 0 of the target representation after the target refinement step; d) level 0 of the template representation; d) level 0 of the template representation obtained without using irregular information in the target refinement step.

the ROI. It should be noted that the representation of the target without the refinement step is poorer than the other, presenting only the biggest square regions of the target. When the irregular part is introduced in the refinement the target is completed.

4.2.5 Template updating

As objects can present severe viewpoint changes along the image sequence, the object template must be updated constantly to follow up varying appearances. In this type of situations, the current template tends to reflect the state of the process better than older templates. However, an excessively fast updating scheme would be sensitive to sudden tracking errors. Therefore, the updated template should be a compromise between the current template and the data. This can be implemented by associating a weight with each vertex of the template model, in order to give more importance to more recent data. Older data are "forgotten" in a linear and smooth manner. Thus, a new parameter is included in the template model:

- $w^{(t)}(m(i,j,l))$. It is the weight associated to each vertex $m^{(t)}(i,j,l)$ of the template $M^{(t)}$ in the current frame t.

The whole template $M^{(t+1)}$ is built by updating $M^{(t)}$. This process is performed at the same time than the template matching and target refinement processes. Thus in the template matching process:

$$m^{(t+1)}(i,j,l) = \begin{cases} m^{(t)}(i,j,l) & \text{if no match} \\ q^{(t)}(i,j,l) & \text{if match} \end{cases} \tag{4.7}$$

$$w^{(t+1)}(m(i,j,l)) = \begin{cases} w^{(t)}(m(i,j,l)) - \alpha & \text{if no match} \\ 1 & \text{if match} \end{cases} \tag{4.8}$$

where the superscript (t) denotes the current frame and the forgetting constant, α, is a predefined coefficient that belongs to the interval $[0, 1]$. This constant dictates the degree of forgetting, i.e., how fast the forgetting action will be. It is related with the degree of deformation that is expected in the tracked object. For example, in the made experiments, an α value equal to 0.1 has been used, which obtains accurate results with objects with a high degree of deformation (i.e. a hand). With $\alpha = 0.1$, a vertex of the template which is never updated will be forgotten in ten frames. If, instead, an $\alpha = 0.2$ is used, the pixel will be forgotten in 5 frames and so on. Eq. (4.7) means that every template point $m^{(t+1)}(i,j,l)$ is obtained from the previous template point $m^{(t)}(i,j,l)$ if there is no match, or from the corresponding point $q^{(t)}(i,j,l)$ in the target if there is match between template and target. Eq. (4.8) means that each weight point $w^{(t+1)}(m(i,j,l))$ is equal to 1 if there is match, or it is the previous one less the constant α if there is not a match. In any case, the lowest value for $w^{(t+1)}(m(i,j,l))$ is zero and $w^{(t+1)}(m(i,j,l)) \in [0..1]$.

The process to update the template continues in the target refinement step. In this stage of the tracking process, when a vertex of the ROI is included in the target, it is also included in the template $M^{(t+1)}$. Its corresponding weight is set to 1.

As was previously commented in this chapter, the template is made only of regular vertices. But these vertices are influenced by the irregular information of the ROI representation due to the target refinement step. Fig. 4.8.d) shows the template representation of the hand in the second frame of the hand sequence. Fig. 4.8.e) shows the same template without using the irregular information in the target refinement step. It should be noted that template cannot represent the elongated parts of the hand if only regular information is used.

Fig. 4.9 presents an example of weighted template updating. In order to illustrate the forgetting action, the intensity value of the template has been multiplied by its associated weight. Thus, darker pixels correspond to older vertices of the template, which are about to be "forgotten".

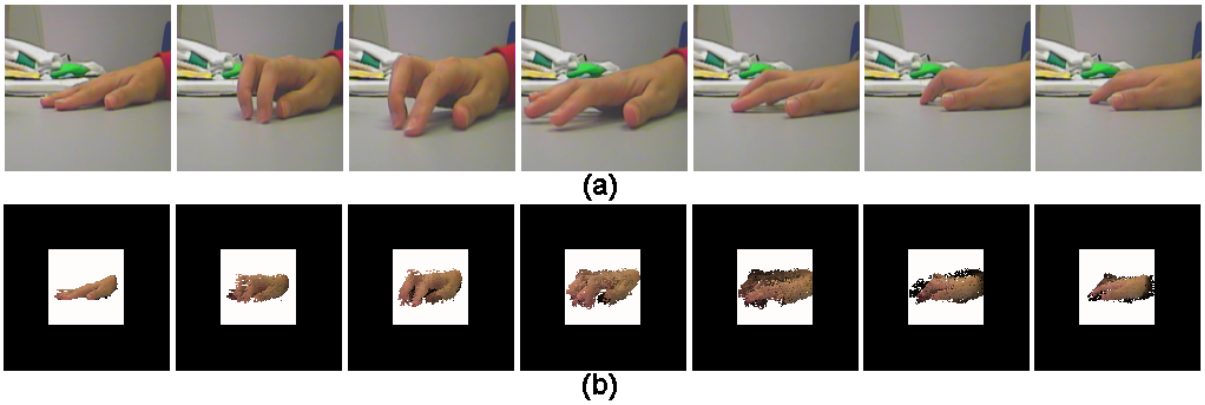

Figure 4.9: Updating the object template: a) sequence frames of a moving hand; and b) updated template.

4.2.6 Region Of Interest updating

Once the target has been found in the current frame t, the new $ROI^{(t+1)}$ can be obtained. This process has two main steps:

1. $ROI^{(t+1)}(0)$ *selection:* Level 0 of the new region of interest is obtained by taking into account the position where the target $T^{(t)}(0)$ is placed in the original image of frame t. Firstly, the algorithm calculates the bounding-box of $T^{(t)}(0)$. Then, $ROI^{(t+1)}(0)$ is made up of the pixels of the next frame which are included in the bounding box $BB(T^{(t)}(0))$ plus the pixels included in an extra border ϵ of the bounding box.

$$ROI^{(t+1)}(0) = \bigcup_{ij} p^{(t+1)}(i,j,0) \tag{4.9}$$

with

$$ij \in \{BB(T^{(t)}(0)) + \epsilon\}$$

This step is performed at the end of the tracking process t. The ϵ value depends on the velocity of the target motion.

2. *Over-segmentation of* $ROI^{(t+1)}(0)$*:* The hierarchical structure $ROI^{(t+1)}$ is built. This step is performed at the beginning of the tracking process $t+1$ and has been previously explained in Section 4.2.2.

4.2.7 Handling occlusions

The previously presented algorithm to track a single object can handle partial occlusions of the object to track due to the use of a weighted template that can automatically adapt itself to appearance changes of the target. Therefore, partial occlusions are handled in the same way as the appearance changes of the object.

With regard to total occlusion, there are two main aspects in the algorithm:

- Selection of $ROI^{(t+1)}$: If there is a total occlusion in frame t, the target will not be found. In this case, the ROI in $t+1$ is selected taking into account the position were the target was found the last time. The extra border ϵ is incremented in one pixel until the target is found or ϵ reaches a maximum value.

- The forgetting constant α: this value has influence in the duration of the total occlusions that the algorithm can handle. In the presence of a total occlusion, the vertices of the template are not updated, and their weights are "forgotten" using α. The template is totally forgotten when the weights are 0. At this moment, the tracking process stops. The α value dictates the degree of forgetting. The smaller the value of the constant, the longer occlusions will be handled. For example, an α vale of 0.1 allows to handle total occlusions that last ten frames.

The proposed tracking algorithm returns the trajectory of the tracked object and the bounding box coordinates of the found target in each frame of the sequence. The trajectory is computed as the centroid coordinates of the found target in the original image of each frame. Figs. 4.10.a)-c) show the initial frame of three video sequences provided by the Advanced Computer Vision GmbH - ACV. The figures illustrate the ground truth trajectories of a moving dot (blue points), together with the trajectories generated by the proposed tracking algorithm (red points). It can be appreciated that the obtained trajectories are very similar to the real ones, in spite of partial and total occlusions (Figs. 4.10.b)-c)). This is due to the fact that the algorithm computes the points of the dot trajectory as the centroid of the found target. When a partial occlusion occurs, the estimated centroid position, calculated from the visible part of the target, differs from the real one. In the case of total occlusions, the target is not found and the centroid keeps the last estimated value. The algorithm can satisfactorily recover the real trajectory of the dot when the occlusion ends. For example, this situation is illustrated in the middle region of Fig. 4.10.b), where the tracked dot is always occluded by the other one.

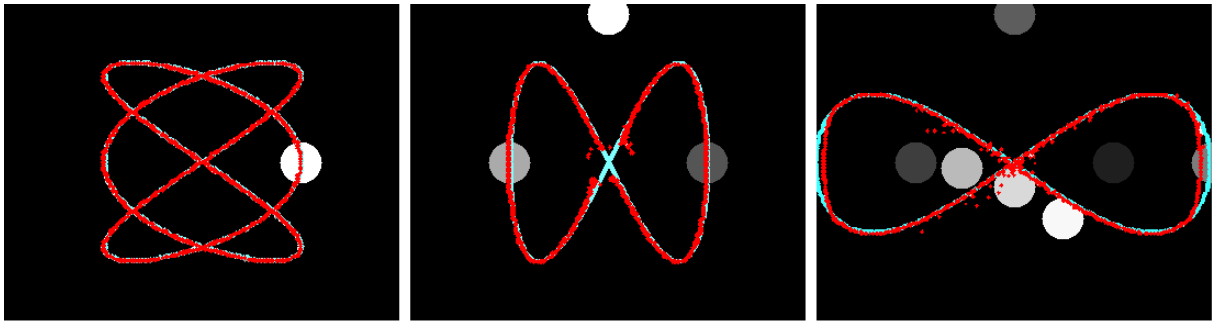

Figure 4.10: a-c) Dot tracking results: real trajectories have been marked as blue points and generated trajectories have been marked as red points.

4.3 Multiple object tracking

Tracking multiple objects using a single tracker for each target is an option. But the increase of the computational cost would be proportional to the number of objects. An adaptation of the previous algorithm to track multiple objects simultaneously with a low increase of the computational cost is presented in this section. This new approach allows to follow up the appearance and position changes of multiple objects into the same hierarchical structure. The targets to track are chosen manually from the first frame using a hierarchical segmentation algorithm in the same way as in the single object tracking process. The objects to track must be distinguishable in the first frame, i.e. if two objects are fused by the segmentation algorithm, it is not possible to split them later. Also if an object is not visible –at least partially– in this first frame, it can not be selected. An independent template is assigned to each target. The first templates and $ROIs$ are extracted from the hierarchical segmentation too. The data flow of the algorithm is the same (Fig. 4.1) with the following modifications:

Over-segmentation. In order to achieve the tracking of several objects into the same BIP, all the $ROIs^{(t)}$ must be hierarchically represented into the same structure. To do that, a BIP is built over the whole input. Level 0 of this BIP has as homogeneous vertices only the vertices of $ROI_i^{(t)}(0)$ with $i \in [1..N]$, being N the number of objects. Thus, only the ROIs are over-segmented. Fig. 4.11 shows the regions obtained in the over-segmentation of three ROIs corresponding to three different objects: a face, a green cone and a green box. The black background pixels represent non homogeneous vertices of level 0 of BIP. If two or more ROIs are overlapped in some frames because of proximity or occlusion among targets the algorithm does not fuse them, maintaining a ROI for each target.

Figure 4.11: a) Original image; b) obtained regions in the over-segmentation of 3 ROIs.

Template matching and Target refinement. Each template $M_i^{(t)}$ has associated a working level $l_{w_i}^{(t)}$. The target localization process explained in Section 4.2.3 is applied simultaneously for all the targets $T_i^{(t)}$. This process starts in the highest working level. In each level l the algorithm searches for all the targets $T_i^{(t)}$ with $l_{w_i}^{(t)} = l$ and for the targets which were not found in the upper level. Each target is only searched in its ROI. It must be noted that when all the targets are located, their hierarchical representations are all included into the same hierarchical structure. Once the targets are found, all $T_i^{(t)}(l)$ are refined in each level l as is explained in Section 4.2.4.

Handling occlusions

In the case of tracking several objects at the same time, some problems can appear when two targets share the same ROI area because of an occlusion. However they can still be correctly separated as long as their colour is not similar, following the strategy explained in Section 4.2.7.

The most important limitation of the proposed algorithm is that it is not able to track several objects with very similar colour in the case of occlusions. This disadvantage is shared with many colour-based methods [55].

Fig. 4.12 shows results in multiple object tracking. The sequences were obtained from the Advanced Computer Vision GmbH - ACV site. The ground truth trajectories are depicted in Figs. 4.12.c)-d). The trajectories obtained by the proposed method are shown in Figs. 4.12.e)-f). Similar conclusions to those of Section 4.2.7 and Fig. 4.10 can be extracted. The synthetic sequence shows several moving objects whose trajectories intersect at multiple points, resulting

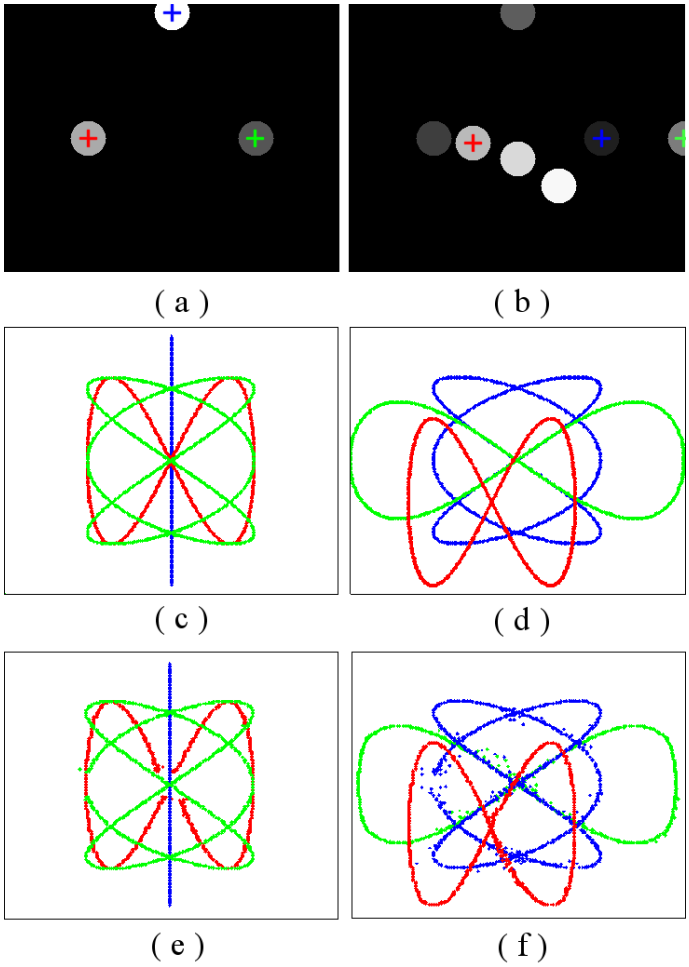

Figure 4.12: a-b) First frame of the sequences. Each tracked dot has been marked with a different colour; c-d) real trajectories of the tracked dots; e-f) generated trajectories with the proposed method.

in occlusions from which the algorithm is able to recover.

4.4 Results

4.4.1 Qualitative and Quantitative evaluation

In order to experimentally validate the accuracy of the proposed tracking system, it has been tested in different situations: partial and total occlusions, appearance changes, moving camera, the presence of other moving objects in the scene, multiple object tracking and illumination changes. In order to perform these tests, different video sequences [1] have been used. The

[1]Some of the video sequences are publicly available at www.grupoisis.uma.es

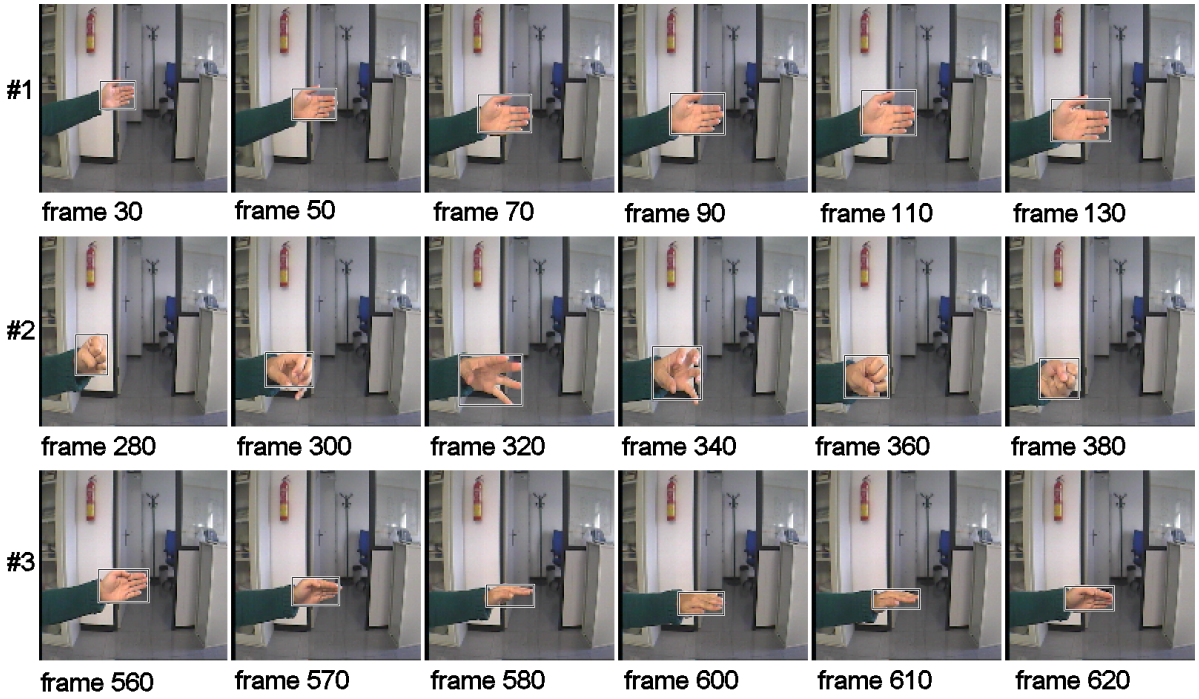

Figure 4.13: Tracking of an object with different appearance changes: #1 Zoom; #2 deformations; #3 rotations.

bounding box of the found target has been marked in all frames. What follows is a summary of the main conclusions extracted from the tests:

- Partial and total occlusions of the tracked object: the capability of the proposed system to handle partial and total occlusions has been explained and demonstrated in Sections 4.2.7 and 4.3.

- Appearance changes of the tracked object: the tracked object can suffer appearance changes due to three types of phenomenons: zooms, deformations and rotations. Fig. 4.13 shows the behaviour of the tracking system in these situations. In the first part of this sequence (#1), the hand has been moved in front of the camera in order to simulate a zoom effect. In Fig. 4.13 #2 a set of frames of the sequence where some deformations occur are shown. Finally, the #3 part of the sequence shows a rotation of the hand. All these appearance changes are correctly handled thanks to the capability of the template to store target information over time and to the target refinement step, which allows to adapt the template and the target to the new appearance of the object more accurately.

- Moving camera: Fig. 4.14 shows a sequence where the camera has been moved around a set of fixed objects. Among these objects, the red box has been tracked. The movement

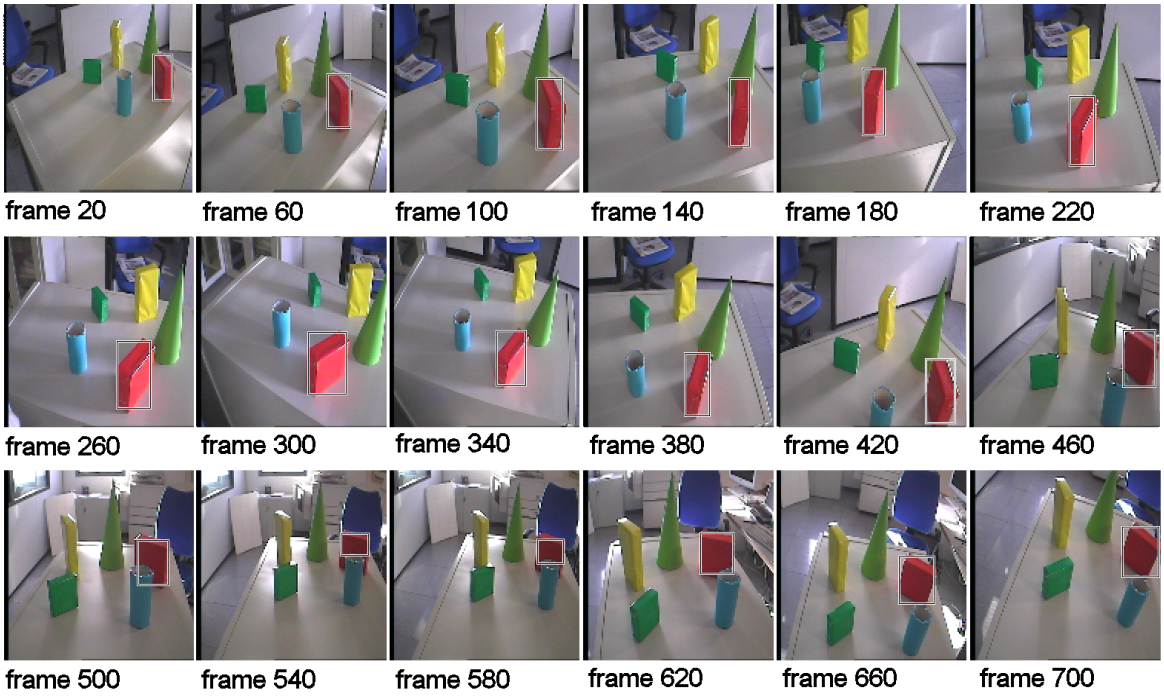

Figure 4.14: Tracking of an object in a sequence captured with a moving camera.

of the camera produces appearance changes of the tracked object due to differences in the viewpoint. These viewpoint changes are correctly handled by the algorithm also thanks to the template updating and target refinement.

- The presence of several moving objects in the scene. Different examples of the behaviour of the proposed system in this case are shown in Figs. 4.10.b)-c), Fig. 4.12.b) and Fig. 4.15. Figs. 4.10 and 4.12 have been previously commented in Sections 4.2.7 and 4.3, respectively. In Fig. 4.15 a face is tracked in the presence of other moving faces. The tracked face suffers from appearance changes (scale changes or zooms and rotations) and partial occlusions.

- Tracking of several objects at the same time. The capability of the proposed tracking system to track several objects at the same time has been previously explained in Section 4.3 of this chapter. Figs. 4.12 and 4.16 show some examples of sequences where several objects have been tracked simultaneously. In these sequences the tracked objects also suffer from partial and total occlusions. The simultaneous tracking of several objects does not imply an increase of the computational time proportional to the number of tracked objects. As it will be studied in Section 4.4.2 the proposed tracking system allows to track several objects without a high increase of the computational time.

Figure 4.15: Tracking of a face in an scene with other moving faces.

- Illumination changes. Fig. 4.17 shows a video sequence where the illumination conditions has been changed. Specifically, three main parts can be distinguished: in the first part of the video sequence - from frame #1 to frame #50 - the illumination has a value of 256 luxes. From frame #60 to frame #100 the illumination value is 64 luxes. Finally, in the third part of the video sequence - from frame #110 to frame #150 - the illumination has a value of 32 luxes. It should be mentioned that the used camera has automatic gain control. In this case, the tracked object is a green box which has been correctly tracked during the whole sequence. It must be noted that the change in the illumination has been made progressively and not abruptly. The proposed tracking approach is capable of handling an illumination change if it satisfies the following condition: the colour variation of the target between two consecutive frames must be smaller than the colour threshold T_c employed in the template matching process. In this case, the template adapts to the illumination changes along the sequence and the target is found in all the frames.

In order to quantitatively assess the accuracy of the proposed tracking method, ground truth data has been generated by manually selecting the tracked object from the input image (see Fig. 4.18.a) and Fig. 4.18.b)). Fig. 4.18.c) shows the results obtained by the proposed algorithm. The error pixels have been computed as the difference between the ground truth and the results of the tracking (Fig. 4.18.d)). The errors are mainly placed in the boundary of the target due to colour transitions. In order to calculate the number of pixels for which an error occurs, two types of pixels should be taken into account: i) pixels of the interest object that the algorithm identifies as background pixels (object errors), and ii) pixels of the background that

Figure 4.16: Tracking of three objects.

the algorithm identifies as target pixels (background errors). The number of both types of error pixels for Fig.4.18 are shown in Table 4.1.

In Fig. 4.19 some of the results obtained by the proposed method and by the mean-shift based approach [34] are shown. The mean-shift algorithm is a line-search iterative algorithm for target search optimization where the iterates are determined along some specific directions. In contrast, the proposed method can be considered as a trust-region one, that derives its iterates by solving the search problem in a bounded region iteratively. Therefore, a trust-region algorithm has more options to select the iterates and, consequently, has better tracking performance [90]. It can be appreciated in Fig. 4.19 that while the mean-shift algorithm loses the target in several frames, the proposed method tracks it correctly. In addition, in this sequence total occlusions of the target appear when the magnet moves out of the image between frames 85 and 97 and between frames 183 and 189. The proposed method successfully handles these short-term total occlusions.

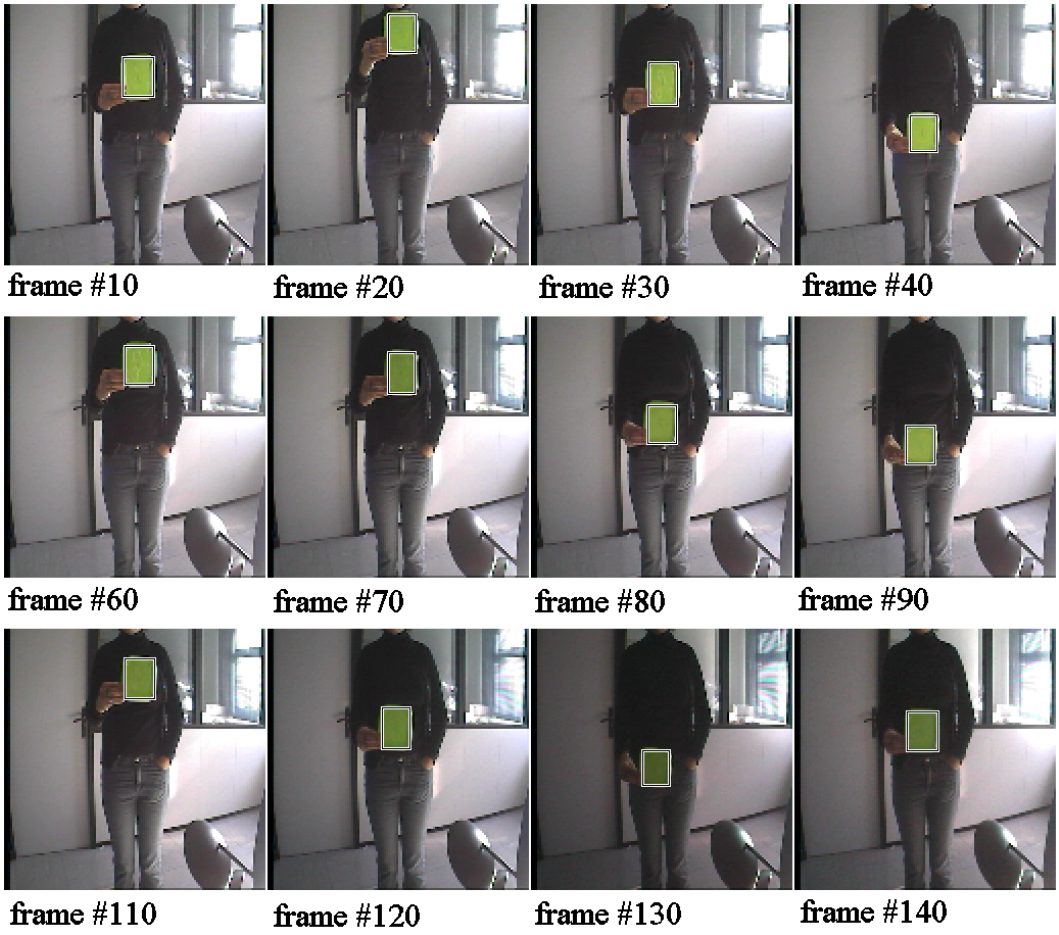

Figure 4.17: Video sequence with illumination changes.

4.4.2 Execution time analysis

In order to analyze the execution time of the proposed tracking algorithm, four different video sequences have been used. Three of such video sequences are the "hand sequence" of Fig. 4.13, the sequence with moving camera shown in Fig. 4.14 and the sequence of Fig. 4.15 where only the yellow box has been tracked. The fourth video sequence is shown in Fig. 4.20, where a green cone has been tracked. In all the experiments a 3GHz Pentium IV PC has been employed and an image size of 128x128 pixels. The tracking algorithm has been divided in three main parts: initialization, over-segmentation and matching. The matching part includes: template matching, target refinement and template updating. The execution time of each of these parts has been computed. The template matching, target refinement and template updating steps have been studied together due to the reduced execution time of the target refinement and template updating steps. The initialization part includes the time required to initialize the different structures used by the algorithm. The obtained results are shown in Table 4.2. It

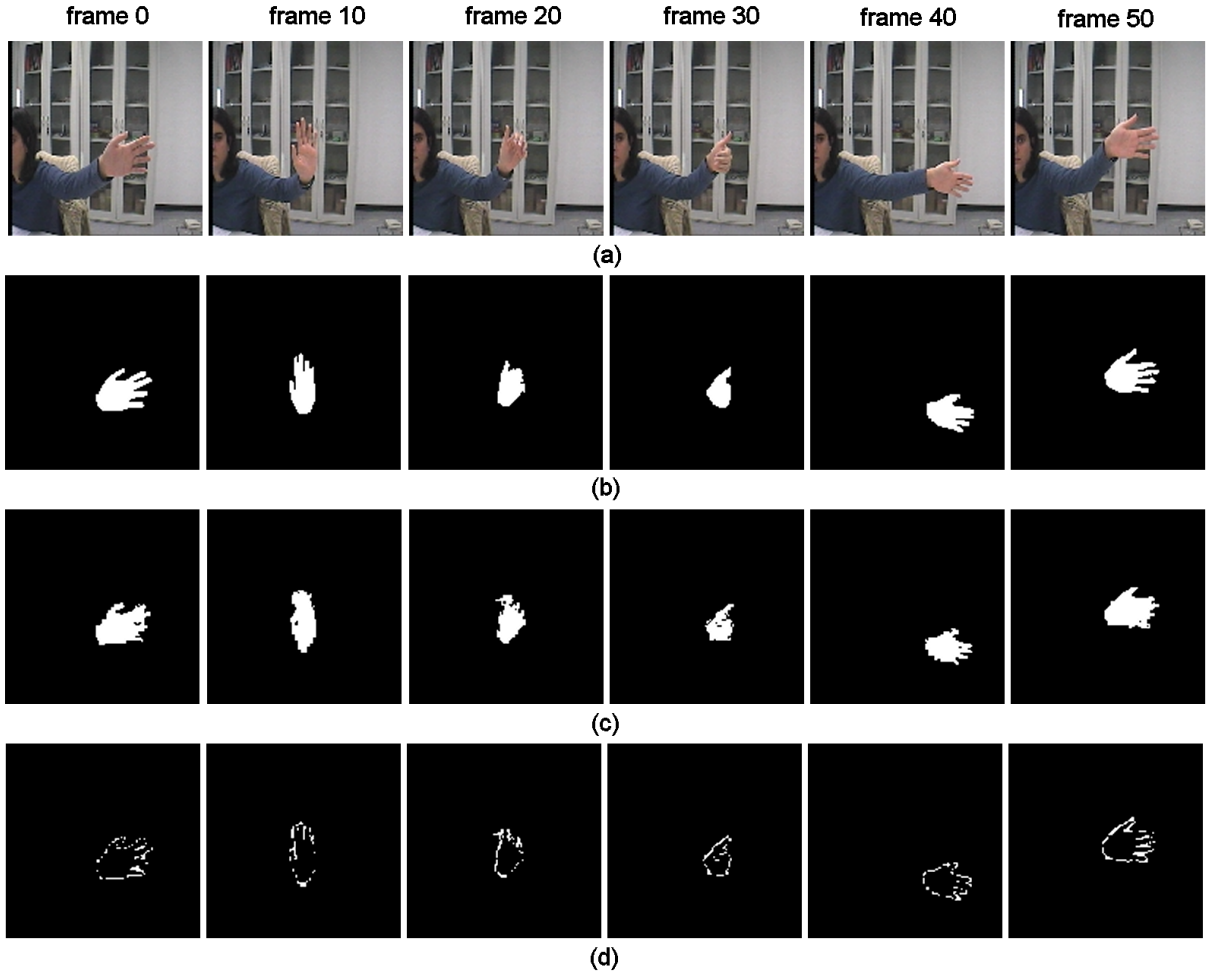

Figure 4.18: a) Sequence frames of a moving hand; b) ground truth; c) tracked targets with the proposed method; d) error pixels.

should be noted that the over-segmentation is the step which consumes more time, because the piramidal structure is built in this step. This time demonstrates the importance of use a fast segmentation algorithm. In Chapter 3 of this Thesis it was demonstrated that the BIP provides faster response than the other piramidal approaches. The over-segmentation time depends on several causes, being the most important the ROI size. In Fig. 4.20 the size of the ROI is less than in the others sequences. Therefore, the over-segmentation time is less in this sequence than in the other ones. Fig. 4.21 shows the increase of over-segmentation time versus the ROI size using the same image. The initialization step consumes an important portion of the total time due to the high complexity of the structures used in the algorithm. These structures must be initialized in each frame. The template matching process consumes a reduced time thanks to the employed hierarchical approach. This time depends on the pyramid level where the object is found. If the object is found in a low level, the matching time is higher than if the object is

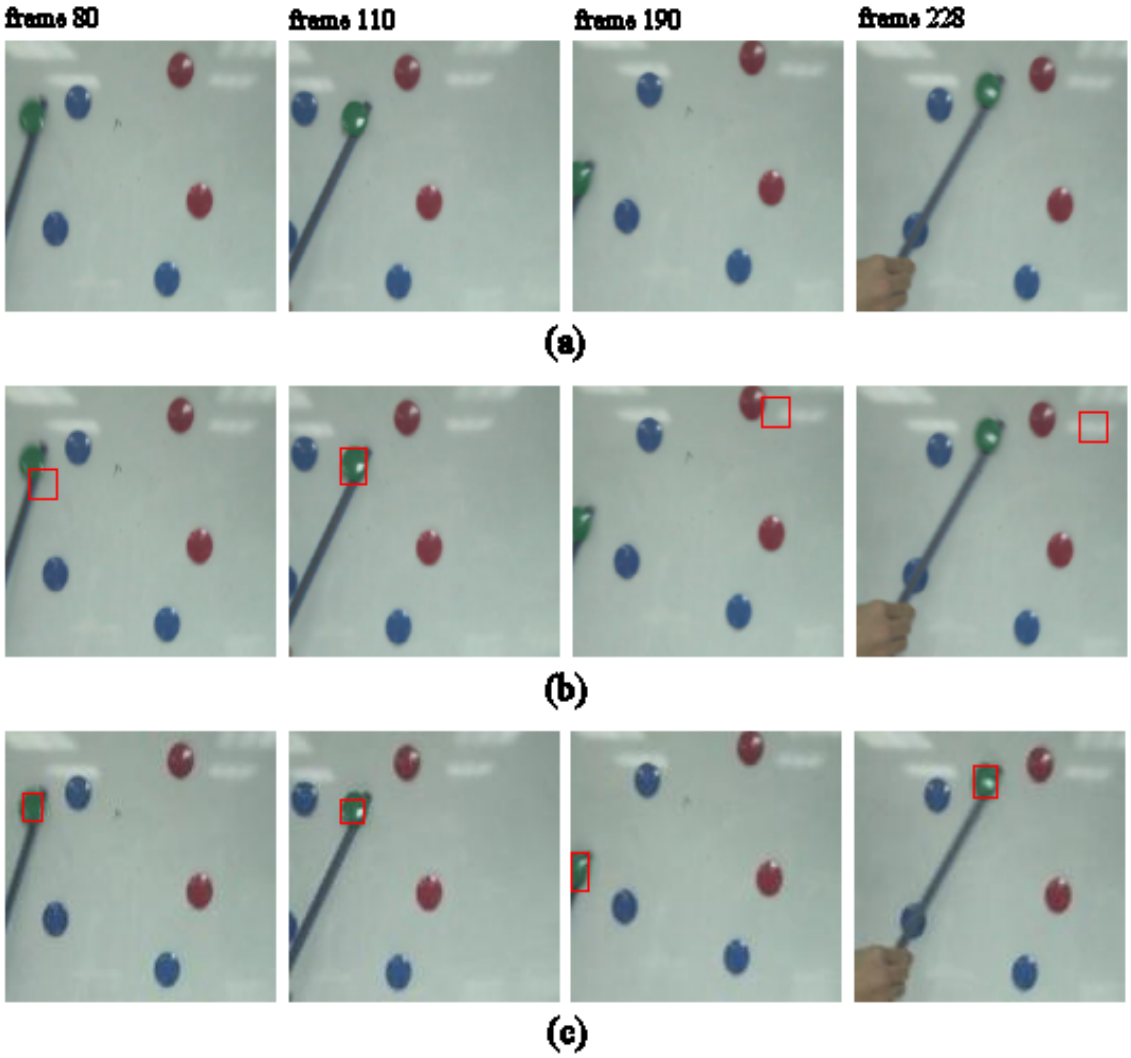

Figure 4.19: Comparison between the proposed method and by the mean-shift based approach by Comaniciu *et al.* [34].

found in a upper level. The green cone has been found in a lower level than the rest of objects of the sequences, consuming more time in the template matching process. The total average time is 36.6 milliseconds. Therefore the proposed tracking approach is capable to process 27 images in a second per average.

In order to study the increase of execution time caused by the tracking of several objects at the same time, the video sequence shown in Fig. 4.16 has been used. Table 4.3 shows the execution times obtained when: i) only the yellow box is tracked (case #1), ii) the yellow box and the red box are tracked (case #2) and, iii) the yellow box, the red box and the green cone are tracked (case #3). The time consumed by the initialization and capture steps remain

Frame	Object Pixels	Background Pixels	Object Errors	Background Errors
0	625	15759	49	62
10	469	15915	34	58
20	351	16033	40	58
30	274	16110	43	33
40	467	15917	73	22
50	615	15769	95	46

Table 4.1: Pixel errors (in numbers of pixels) in Fig. 4.18.

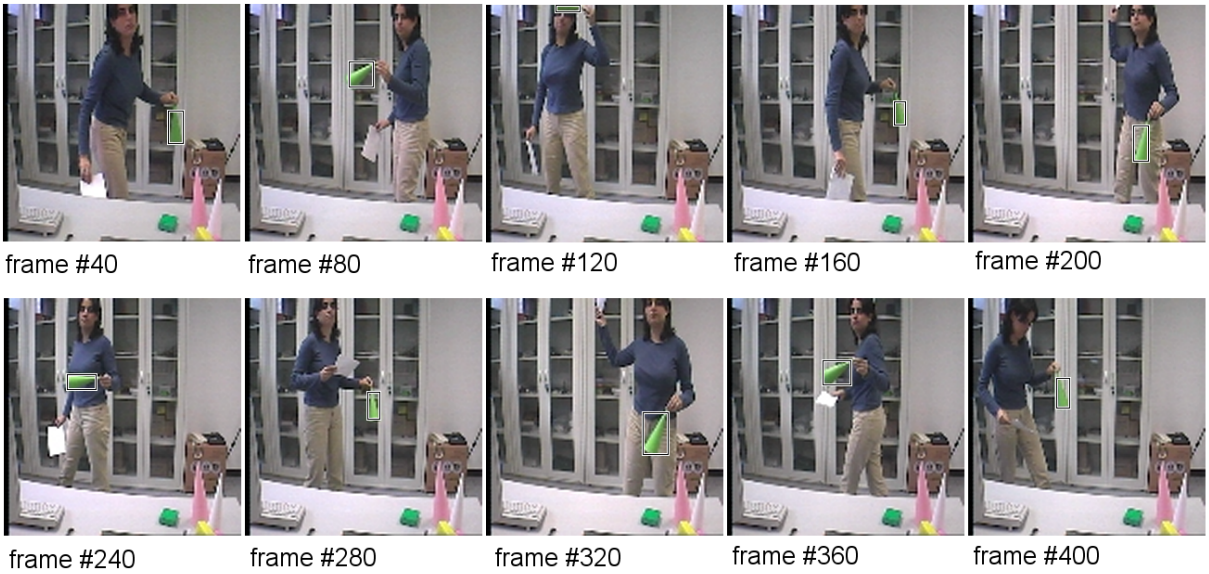

Figure 4.20: Tracking of a green cone.

constant because they are independent from the number of objects. It can be appreciated as the over-segmentation and matching time do not increase proportionally with the number of objects. The over-segmentation time depends on the ROI's size (not on the number of objects). The matching process is performed level by level for all the objects at the same time. That is, if there are three objects, the structure is not traversed three times to look for the targets, it is traversed only once. Therefore, the total time consumed by the tracking approach does not increase proportionally with the number of objects.

4.4.3 Estimation of parameters

The proposed method requires choosing values for a set of parameters. These parameters are:

- The colour threshold, T_c, which determines the maximum distance between two colours that are considered as equal. It is used in the target localization step of the tracking

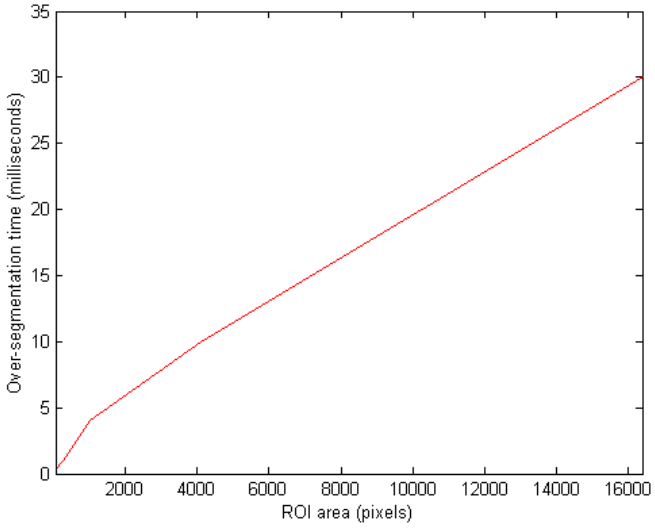

Figure 4.21: Over-segmentation time versus ROI size.

	Execution times per frame milliseconds			
Sequences	Initialization	Over-segmentation	Matching	Total [a]
Hand	6.5	12.4	5.8	38.1
Moving camera	7.5	11.1	5.9	37.1
Yellow box	6.5	10.7	5	35.6
Green cone	6.5	8.5	8.1	35.7

[a] including image capture

Table 4.2: Execution times in single object tracking.

process.

- The colour similarity threshold T_{co} employed by the over-segmentation algorithm.
- The colour similarity threshold T_{cr} used in the target refinement step.
- The forgetting constant, α, which dictates the degree of forgetting of the template.
- The extra border ϵ of the bounding box. This extra border ensures that the target in the next frame will be placed in the new ROI.
- The constant T_A which determines the working level l_w. Thus, l_w is the highest level whose template area is at least a $T_A\%$ of the total area of the template.
- The percentage of overlap between target and template necessary to consider that the target has been found in a particular position.

	Execution times per frame milliseconds			
Tracked objects	Initialization	Over-segmentation	Matching	Total [a]
case #1	6.5	10.7	5	35.6
case #2	7	13.2	7	40.2
case #3	7.6	20.1	11	51.6

[a] including image capture

Table 4.3: Execution times in multiple object tracking.

Two of these parameters, α and ϵ, are user-specified parameters that must be chosen depending on the final application. The extra border ϵ is related to the maximum speed of the movement of the tracked object. In the tests, a ϵ value of 6 pixels has demonstrated to be adequate for the speed of all tracked objects. If the target is lost in a frame the ϵ value is increased in one pixel in each subsequent frame until the target is found or the ϵ has a maximum value of 12 pixels. The constant α is related to the forgetting action associated to a situation where the tracked object is lost. In all tests a value of 0.1 has been used, i.e. it is necessary to miss the tracked object during ten frames to decide that this object is no longer in the scene.

The value T_A is not a very sensible parameter. If it is too large, the working level will be lower than the optimum value but the target will still be found. If it is too low, the working level will be higher than the optimum value. In this case, the target will be not found in the working level, but will be found in a lower level. For these two cases, the target is correctly tracked but with a higher processing time than if the working level corresponds to the optimum value. In all experiments presented in this paper a T_A value of 80 % has been used. The percentage of overlap necessary to consider that the target has been found is a more restricted parameter. If it is too high, it will be very difficult to find the target. If it is too low, the algorithm could consider that the target is at an incorrect position. A value of 70 % has been adequate for all the tests.

The colour similarity threshold T_{co} employed by the over-segmentation algorithm must not be higher than the value which produces errors in the segmentation of the ROI. That is, if T_{co} is so high, then some regions of the ROI which are not in the target can be fused with regions of the target. In order to assure that this error does not occur and that the tracking results are not very dependent of the accuracy of the segmentation, it is recommended to use small values of T_{co}, i.e. $T_{co} \in [5..20]$. The tests have shown that any value within this interval does not produce errors in the segmentation, and that the value of this parameter has not a big influence in the final result of the tracking. In all of the experiments shown in the results section

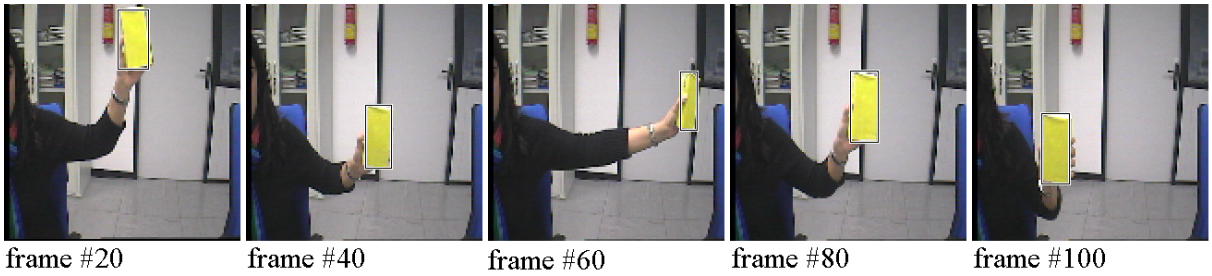

Figure 4.22: Tracking of a yellow box in front of a grey background.

Figure 4.23: Tracking of a yellow box which is mixed with a red box.

of this paper a value of $T_{co} = 10$ has been used.

The other two colour similarity thresholds are the most sensible parameters of the proposed method. They depend on the colour of the object to track, as well as on the colours present in the environment where the object is moving around. Specifically, they depend on the colour similarity between the object colour and the rest of colours in the scene. For example, suitable thresholds used in the sequence showed in Fig. 4.22 were $T_c = 30$ and $T_{cr} = 70$. These thresholds have been used to track the same yellow box in the sequences shown in Figs. 4.23 and 4.24. In the first of these sequences the yellow box is mixed with the red box. In the second one, the yellow box is mixed with the hand. The more suitable thresholds for these two sequences were $T_c = 20$, $T_{cr} = 10$ and $T_c = 30$, $T_{cr} = 20$, respectively. The Tc threshold is similar in all the cases, while the T_{cr} is very different between the first sequence and the rest ones. This is due to the fact that the yellow box in Fig. 4.22 is moving in a scene without similar colours around, front of a gray background. The threshold employed in the target refinement step can thus be larger because the target cannot be linked with any similar object. It should be mentioned that the tracking process in the sequence of Fig. 4.22 also obtains good results with a T_{cr} value equal to 10 or 20.

Figure 4.24: Tracking of a yellow box which is mixed with a hand.

4.5 Summary

In this Chapter, a novel tracking algorithm is proposed and evaluated. The algorithm has four main stages: firstly, a hierarchical representation of the region of interest (ROI) is constructed via an oversegmentation obtained with the BIP. Secondly, the template matching is performed over the hierarchical representation of the ROI. This hierarchical matching reduces the computational cost of the process. In this step, the regular base of the hierarchical representation of the target is obtained. In the third stage, the target representation is refined incorporating regular and virtual vertices from the hierarchical representation of the ROI. During the template matching and the target refinement steps, the hierarchical representation of the template is updated by including the regular vertices of the target and by updating their weights. Finally, the new ROI for the next frame is selected.

This algorithm has demonstrated to handle partial and short-term total occlusions thanks to the weighted template representation. In addition, in the results section of this chapter, the proposed algorithm has been tested in different challenging situations such as: appearance changes of the objects, illumination changes, movements of the camera and presence of multiple targets, demonstrating its capability to handle these situations. A study of the execution time of the algorithm has been presented in order to assure the real time performance of the proposed approach using images of 128x128 pixels.

Chapter 5

Applications

In the previous chapter of this Thesis, the proposed tracking algorithm has been explained and tested in different situations in order to evaluate its performance. The goal of this chapter is not to validate the proposed tracking system as an isolated process, but rather to demonstrate the usefulness of the proposed tracking in real time applications such as an attentional mechanism and a human motion capture system. Both applications are presented in this chapter, emphasizing the contributions of the tracking algorithm.

5.1 Attentional Mechanism

An attentional mechanism is a process to select the most salient information from the broad visual input in a vision system. The use of attention to reduce the amount of input data has two main advantages: i) the computational load of the whole system is reduced, and ii) distracting information is suppressed.

In this section a general purpose attentional mechanism based on the feature integration theory [150] is presented. Attentional mechanisms based on this theory are divided in two main parts. Firts, in a task-independent preattentive stage, a set of early features are computed in parallel. 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 attention is moved to each salient region to analyze it in a sequential process. A general problem in attentional mechanisms is to avoid revisiting or ignoring salient objects of the image when the system is working in a dynamic environment with moving objects. To solve this problem, it is necessary to include in the system a mechanism to avoid extracting the same objects in different frames, although they will be in different positions in

the images. The attentional mechanism should be object-oriented and not region-oriented. The way to solve the problem of revisiting or ignoring objects is called "inhibition of return". The attentional mechanism presented in this chapter implements the inhibition of return by including an intermediate semiattentive stage where the tracking algorithm proposed in this Thesis is used to track the objects extracted from the scene. This tracking allows to know the position in the current frame of the previously extracted objects. This prevent the attentional mechanism from wrongly identify them as new objects. Fig. 5.1 shows the overview of the proposed architecture. It is related to the recent proposal of Backer and Mertsching [6] in several aspects. The first is the use of a preattentive stage in which parallel features are computed and integrated into a saliency map. However, in contrast with this and other attentional systems, the skin colour as input feature is introduced in order to detect human faces or hands as possible regions of interest. Thus, skin colour is first detected using a chrominance distribution model [142] and then integrated as input feature in a saliency map. Other similarity is that this preattentive stage is followed by a semiattentive stage where a tracking process is performed. However, while Backer and Mertsching's approach performs the tracking over the saliency map by using dynamics neural fields, the proposed method tracks the most salient regions over the input image using the tracking approach presented in this Thesis. The main disadvantage of using dynamic neural fields for controlling behavior is the high computational cost of simulating the field dynamics by numerical methods. The output objects of this semiattentive stage will be the inputs of an attentive stage. This attentive stage depends on the final application of the vision system where the attentional mechanism is included. Some examples of attentive stages are the exploration of a scene or the search and tracking of a specific object. This section is focused in the task-independent part of the attentional mechanism. Therefore, an attentive stage is not explained here.

The different modules of the proposed attentional mechanism are explained in the following sections.

5.1.1 Preattentive stage

The proposed attentional mechanism uses a number of features computed from the available input image in order to determine how interesting a region is in relation to others. These features are independent of the task and they allow to extract the most interesting regions of the image. Besides, they allow to distinguish locations where a human may be placed. The chosen features are colour and intensity contrast, disparity and skin colour. Attractivity maps are computed from these features, containing high values for interesting regions and lower values

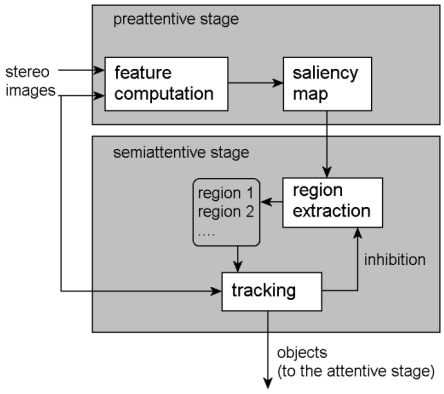

Figure 5.1: a) Overview of the proposed attentional mechanism and b) overview of the tracking algorithm.

for other regions. The integration of these feature maps into a single saliency map allows to determine what regions of the input image are the most interesting. Other features can be easily added without changes in the following steps.

5.1.1.1 Computation of early features

Colour contrast

Colour is employed to distinguish objects in most attentional models. The first step to compute colour contrast is to choose an adequate colour space. The HSV colour space has been selected due to its intuitive representation and the facility to separate the chrominance from the luminance information. Thus, the RGB colour information is firstly transformed into the HSV colour space. Secondly, the input image is segmented using the Bounded Irregular Pyramid (BIP) (Chapter 3) in order to obtain homogeneous colour regions. And finally, in contrast with other methods which only compute the colour contrast for a set of colours [6], the proposed algorithm computes a colour contrast value for each homogeneous colour region of the input image independently of its colour. The colour contrast of a region i is calculated as the mean colour gradient MCG_i along its boundary to the neighbour regions:

$$MCG_i = \frac{S_i}{PL_i} \sum_{j \in N_i} pl_{ij} * d(< C_i >, < C_j >) \tag{5.1}$$

being PL_i the length of the perimeter of the region i, N_i the set of regions which are neighbours of i, pl_{ij} the length of the perimeter of the region i in contact with the region j, $d(< C_i >, < C_j >)$ the Euclidean distance between the mean colour values $< C >$ of the regions i and j and S_i

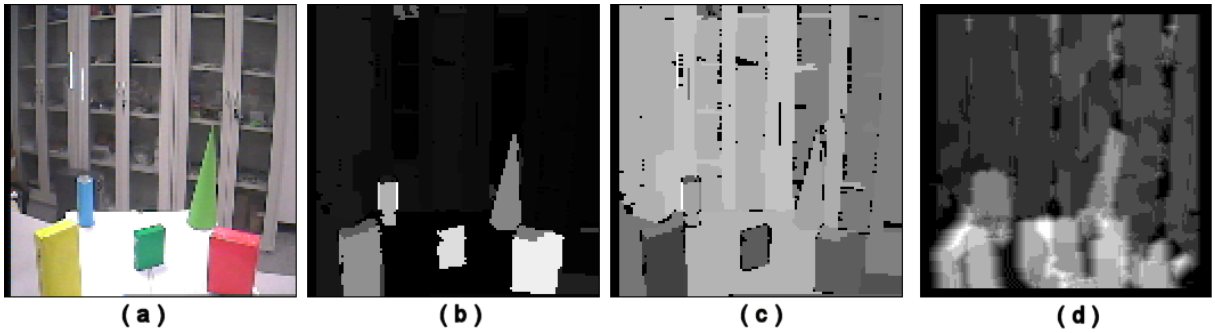

Figure 5.2: Colour and intensity contrast computation: a) left input image; b) colour contrast saliency map; c) intensity contrast saliency map; and d) disparity map.

the mean saturation value of the region i. Fig. 5.2.b) shows the colour contrast saliency map associated to Fig. 5.2.a). It must be noted that the use of S_i in the MCG avoids that colour regions with low saturation (grey regions) obtain a higher value of colour contrast than pure colour regions. The problem is that white, black and pure grey regions are totally suppressed. To take into account these regions, the intensity contrast is computed.

Intensity contrast

This feature map is computed in a similar way to the previous one. The intensity contrast of a region i is the mean intensity gradient MIG_i along its boundary to the neighbour regions:

$$MIG_i = \frac{1}{PL_i} \sum_{j \in N_i} pl_{ij} * d(< I_i >, < I_j >) \tag{5.2}$$

being $< I_i >$ the mean intensity value of the region i. Fig. 5.2.c) shows the intensity contrast saliency map associated to Fig. 5.2.a).

Skin colour

Skin colour is an important tool to distinguish locations in which a human is probably located. In order to segment skin colour regions from the input image, it is necessary to compute an accurate skin chrominance model using a colour space. The skin chrominance model has been built over the TSL colour space, using a method based on the one proposed by Terrillon and Akamatsu [142]. Thus, the skin colour has been modelled in the TSL colour space as an unimodal elliptical Gaussian joint probability density function computed on a set of 120 training images. Fig. 5.3 shows some of the used training images. This pdf function is represented by its covariance matrix C_s and its mean vector m_s. The Mahalanobis metric has been used to empirically determine a threshold value T_s that efficiently discriminates between human skin

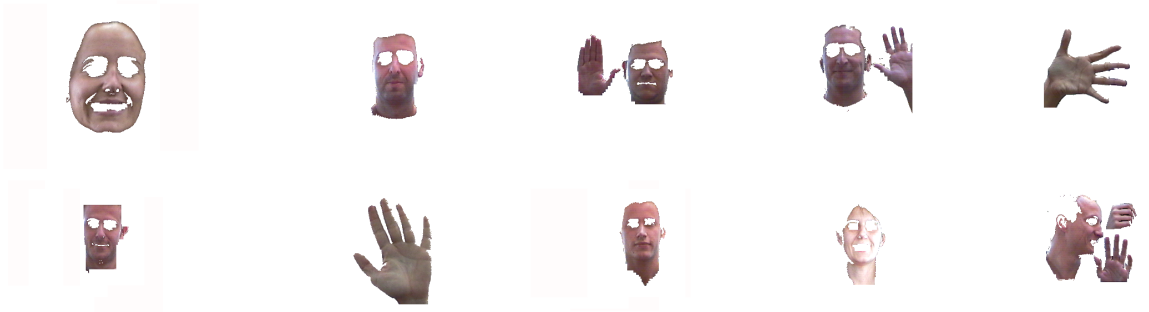

Figure 5.3: Examples of training images used in the computation of the skin colour chrominance model.

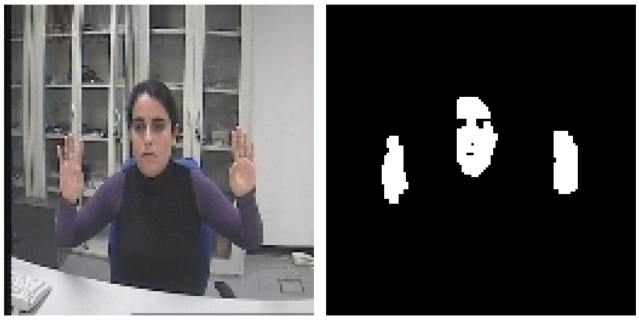

Figure 5.4: Skin colour computation: a) left input image; and b) skin colour map. White pixels correspond to pixels of the input image labelled as skin.

and other objects.

Once the chrominance model has been established, the steps to segment skin regions from an image are the following: first, the RGB input image is transformed into a TSL image. Second, the Mahalanobis distance from each pixel (i, j) to the mean vector is computed. If this distance is less than T_s then the pixel (i, j) of the skin feature map is labelled as "skin". Fig. 5.4.b) shows the skin colour saliency map associated to Fig. 5.4.a).

Disparity

Relative depth information is obtained from a dense disparity map. Closed regions are considered more important. As disparity estimator, the zero-mean normalized cross-correlation measure has been employed. It is implemented using the box filtering technique. This allows to achieve fast computation speed [136].

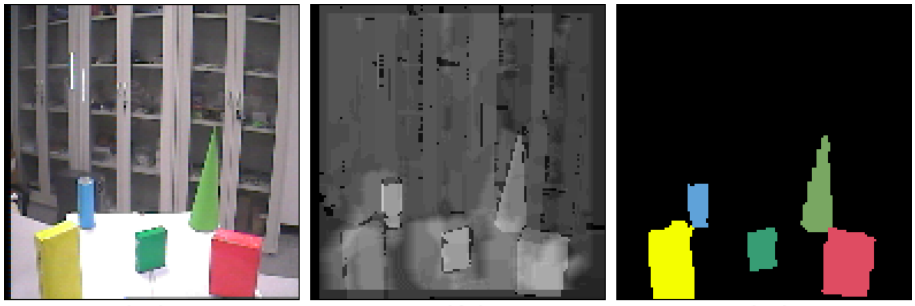

Figure 5.5: Saliency map computation and targets selection: a) left input image; b) saliency map; and c) selected targets.

Each computed zero-mean cross-correlation value is stored in a 3D disparity space with size MxNxD, where MxN is the image size and D the maximum disparity range. The disparity map is found in this space by obtaining the global 3D maximum surface which is computed using the two-stage dynamic programming technique proposed by Sun [136]. Fig. 5.2.d) shows the disparity map associated to Fig. 5.2.a).

5.1.1.2 Saliency map computation

Similarly to other models [6], the saliency map is computed by combining the feature maps into a single representation. To do that, all the feature maps are normalized to the same dynamic range, in order to eliminate cross-modality amplitude differences due to dissimilar feature extraction mechanisms. A simple normalized summation has been used as feature combination strategy. Fig. 5.5.b) shows the saliency map associated to 5.5.a).

5.1.2 Semiattentive stage

Once the saliency map is calculated, it is segmented in order to obtain regions with homogeneous saliency. Among the set of obtained regions, larger regions with a high saliency value are taken into account.

Once the most salient regions of the scene are selected, they are tracked in successive frames in order to implement correctly the inhibition of return. The employed tracking algorithm has been the proposed one in this Thesis.

The most salient regions obtained by segmentation of the saliency map are directly related to homogeneous colour regions of the segmented left input image. These homogeneous

colour regions are the targets to track. Fig. 5.5.c) shows the selected targets associated to the saliency map in Fig. 5.5.b). It must be noted that targets are not necessary associated with homogeneous saliency regions, but with homogeneous colour ones. This mechanism provides better object candidates to the tracking stage. Once the targets are choosen, the algorithm extracts its hierarchical representations from the BIP built during the colour segmentation of the input image. The regular part of each hierarchical structure is the first template $M_r^{(0)}$ and its spatial position is the first region of interest $ROI_r^{(0)}$, where $r \in [1...N]$ and N is the number of salient regions to track. In the case of skin colour regions the employed similarity criterium to build the BIP and to perform the tracking is to be a skin or a non-skin vertex using the chrominance model proposed by Terrillon and Akamatsu [142].

5.1.3 Results

The above described attentional scheme has been examined through experiments which include humans and objects in the scene. Fig. 5.6.a) shows a sample image sequence seen by a stationary binocular camera head. Every 10th frame is shown. All salient regions are marked by black and white bounding boxes in the input frames. It must be noted that the activity follows the objects closely, mainly because the tracker works with the segmented input image instead of working with the saliency image. This approach has two main advantages: i) the regions of the segmented left image are more stable across time than the regions of the saliency map, and ii) the regions of the segmented image represent real objects closer than saliency map regions. Furthermore, the tracking algorithm prevents the related object templates from being corrupted by occlusions. Backer and Mertsching [6] propose to solve the occlusion problem with the inclusion of depth information. However, depth estimation is normally corrupted by noise and is often coarsely calculated in order to bound the computational complexity. The proposed tracker is capable of handling scale changes, object deformations, partial occlusions and changes of illumination. Fig. 5.6.b) presents the saliency maps after inhibiting the regions which have been tracked in each frame. This inhibition prevents the region extraction process from extracting regions that have been already extracted in previous frames. In frame 1, the yellow box and the red extinguisher have been detected. The yellow box is tracked over the whole sequence because its saliency remains high. However, the saliency of the extinguisher goes down between frames 21 and 30 and therefore it is not tracked from frame 30 to the end of the sequence. In frame 11, a hand with a green cone is detected in the image. In frame 51, a red box is introduced in the scene. This box is not detected until frame 91, when it becomes located nearer to the cameras than the other objects. In frame 81, an occlusion of the green cone is correctly handled by the tracking

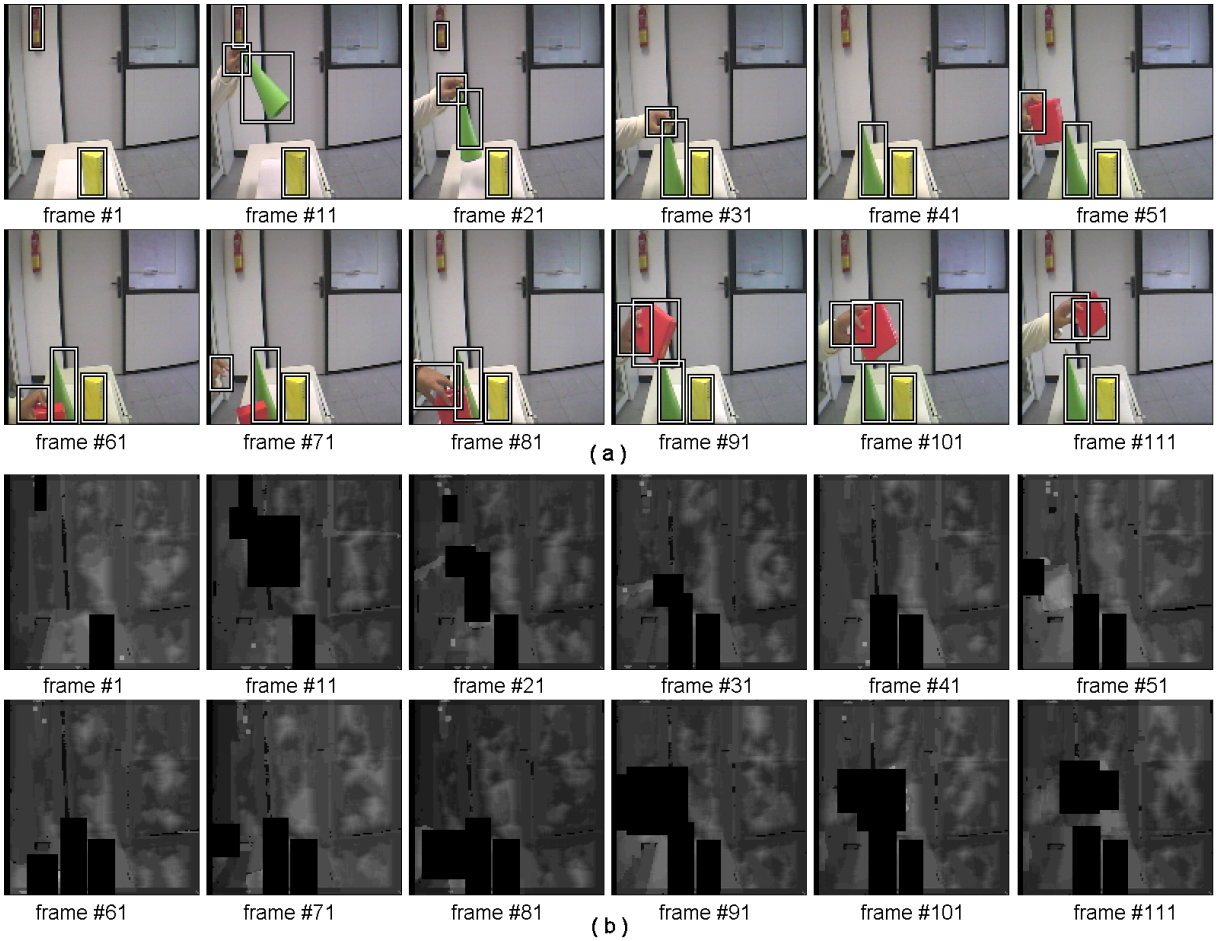

Figure 5.6: Example of selected targets: a) left input images; and b) saliency map associated to a).

algorithm, which is capable of recovering the object before frame 91. It can also be observed how the mechanism follows appearance and view point changes of the salient objects.

The proposed method runs at 5 frames per second with 128x128 24-bit colour images, being faster than Backer's proposal [6] which is reported to take 30 seconds to process one frame.

5.2 Human motion capture system

This section explains a novel real-time human motion analysis system based on the proposed hierarchical tracking and on inverse kinematics. This system is a computer-vision based, upper-body motion analysis system that works without special devices or markers. Since such system is unstable and can only acquire partial information because of self-occlusions and depth ambiguity,

a model-based pose estimation method based on inverse kinematics have been employed. The resulting system can estimate upperbody human postures with limited perceptual cues, such as centroid coordinates and disparity of head and hands.

The key idea behind this system is the assumption that in order to track the global human body motion, it is not necessary to capture with precision the motion of all its joints. Particularly, it is centered in the motion of the upper-body. It is assumed that the robot only needs to track the movement of the head and hands of the human, because they are the most significant items involved in the human-to-human interaction processes. These are modelled by weighted templates that are updated and tracked at each frame using the proposed tracking approach. The pose of the joints is then extracted through the use of a kinematic model of the human to track. It is also assumed that the human motion speed is bounded and that the pose of the different items to track is related to its last detected pose. By assuming this important constraints, the human motion capture system can estimate upper-body human motion at 25 frames per second.

The vision system employed in this human motion capture approach consists of a stereo system with limited baseline (28 mm), mounted over a HOAP-I robotic platform. The goal is to achieve that the robot imitates the movements of a human teacher without any external devices or markers.

5.2.1 Model representation

In the human motion capture system explained in this chapter, a model of human appearance is used with two purposes: i) tracking fast, non-rigid movement of head and hands, and ii) providing the joint angle information required for the robot to imitate the movement. The weighted templates associated with the hands and head of the teacher, which are used by the proposed tracking approach, are included in the model. Besides, to estimate articulated motion, the model includes a 3D geometric structure composed of rigid body parts.

5.2.1.1 Model geometry

The geometric model contains parts that represent hips, head, torso, arms and forearms of the human to be tracked. Each of these parts is represented by a fixed mesh of few triangles, as depicted in Fig. 5.7. Each mesh is rigidly attached to a coordinate frame, and the set of coordinate frames is organized hierarchically in a tree. The root of the tree is the coordinate

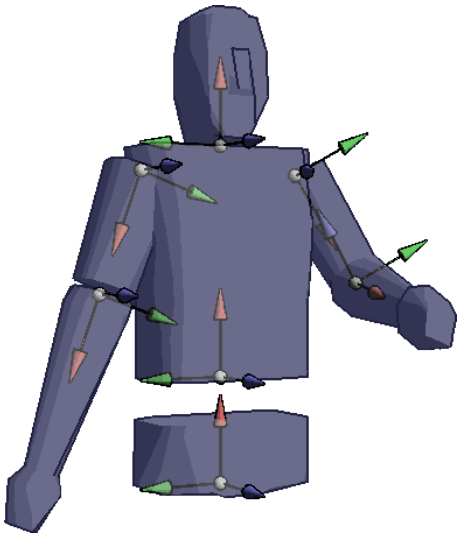

Figure 5.7: Illustration of the human upper-body kinematic model.

frame attached to the hips, and represents the global translation and orientation of the model. Each subsequent node in the tree represents the three-dimensional rigid transformation between the node and its parent. This representation is normally called a skeleton or kinematic chain (Fig. 5.7). Each node, together with its corresponding body part attached is called a bone. Each bone is allowed to rotate -but not translate- with respect to its parent around one or more axes. Thus, at a particular time instant t, the pose of the skeleton can be described by $\Phi^{(t)} = (R^{(t)}, \vec{s}^{(t)}, \phi^{(t)})$, where $R^{(t)}$ and $\vec{s}^{(t)}$ are the global orientation and translation of the root node, and $\phi^{(t)}$ is the set of relative rotations between successive children. For upperbody motion tracking, it is assumed that only ϕ needs to be updated -this can be seen intuitively as assuming that the tracked human is seated on a chair.

5.2.2 Human motion tracking algorithm

The human capturing algorithm is applied to track simultaneously the movements of the hands and the head of a human in a stereo sequence, i.e. not only the motion, but also the depth of the tracked objects is calculated in each frame by taking into account the position differences between the left and right images. The tracking algorithm can be divided in two main processes: i) movement tracking, which tracks the objects between the left frame t and the left frame $t+1$, and ii) depth estimation, which can be explained as a tracking process between the left frame t and the right frame t.

The movement tracking is performed by the tracking approach presented in this Thesis

with a slight modification: the similarity criterium employed to build the BIP and to perform the tracking is to be a skin or a non-skin vertex. To distinguish between skin and non-skin vertices, the chrominance model proposed by Terrillon and Akamatsu [142] has been used. The targets to track are automatically chosen from the initial left frame as the three largest skin color image regions.

The process to estimate the depth of the hands and the head is explained below.

5.2.2.1 Depth estimation

In order to obtain the relative depth among the tracked objects, the disparity of the objects in each stereo pair is estimated. The first step to obtain the disparity value for each target is similar to the tracking process between consecutive frames but it is applied between two stereo images. The main differences are:

- The templates do not use information from the previous frames, they take into account only the located target in the current left image. Thus the updating template step is slightly different:

$$m^{(t+1)}(i,j,l) = \begin{cases} 0 & \text{if no match} \\ q^{(t)}(i,j,l) & \text{if match} \end{cases} \tag{5.3}$$

$$w^{(t+1)}(m(i,j,l)) = \begin{cases} 0 & \text{if no match} \\ 1 & \text{if match} \end{cases} \tag{5.4}$$

- In the template matching step, each template is only shifted along the horizontal direction due to the parallel arrangement of the cameras of the stereo vision system.

When a target is found in the right image, its disparity can be roughly estimated as the shift with maximum overlap. However, due to the limited baseline of the stereo system, sub-pixel accuracy is required. Sub-pixel accuracy can be obtained by fitting a second-degree curve to the overlapping values in the neighbourhood of the previously estimated disparity. The maximun of this function constitutes a subpixel improvement of the disparity estimation for the studied target. In order to reduce the noise of the final disparity values, the previous estimations are filtered using a fourth-order low pass filter.

Finally, the (X,Y,Z) coordinates of the centroids of the hands and the head are computed using the disparity values and the calibration parameters of the cameras.

5.2.2.2 Joint angle extraction

Once the (X,Y,Z) coordinates of the hands and the head have been computed, the kinematic model is used to apply a simple and fast analytic inverse kinematics method to extract the required joint angles. This process is not explained here because it is not directly related with the proposed tracking approach. A detailed explication of this process can be found in:

- J.P. Bandera, L. Molina-Tanco, R. Marfil and F. Sandoval, A Model-based Humanoid Perception System for Real-time Human Motion Imitation, Proc. of the IEEE Conference on Robotics, Automation and Mechatronics, pp. 324-329, Singapore (Singapore), December 2004.
- J.P. Bandera, R. Marfil, L. Molina-Tanco, A. Bandera y F. Sandoval, Model-based Pose Estimator for Real-time Human-Robot Interaction, aceptado en: Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005), Singapore (Singapore), December 2005.
- L. Molina-Tanco, J.P. Bandera, R. Marfil and F. Sandoval, Real-time Human Motion Analisys for Human-Robot Interaction, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1808-1813, Alberta (Canada), August 2005.

5.2.3 Results

The experimental setting consists of two standard PCs. The first computer runs the whole system, and is connected to the second PC via a standard LAN. The second PC receives the joint angle estimations and sends them to the HOAP-1 robot via radio. The whole system, including skin color detection and segmentation, simultaneous tracking of head and hands, depth estimation and inverse kinematics runs in real-time (25 fps). Fig. 5.8 shows some example results. The top row shows frames of a sequence captured with the left-eye camera. The middle row shows the estimated model pose. Frames (a) to (d) show a correctly estimated bending action of the right elbow. Frames (e) and (f) show an arm-crossing action. The sequence shows that the proposed system can estimate upper-body poses from the estimated 3D locations of the face and hands. Thanks to the used depth estimation approach, the limited baseline does not prevent the system to correctly differentiate between stretched and bent arms (Fig. 5.8.a) and Fig. 5.8.e)). The bottom row shows the corresponding pose adopted by the robot. In frames (a) to (d) the elbow bending action is correctly imitated by the robot. However in frames (e), (f) the rotational limits of the robot joints prevent it from imitating a correctly estimated subject

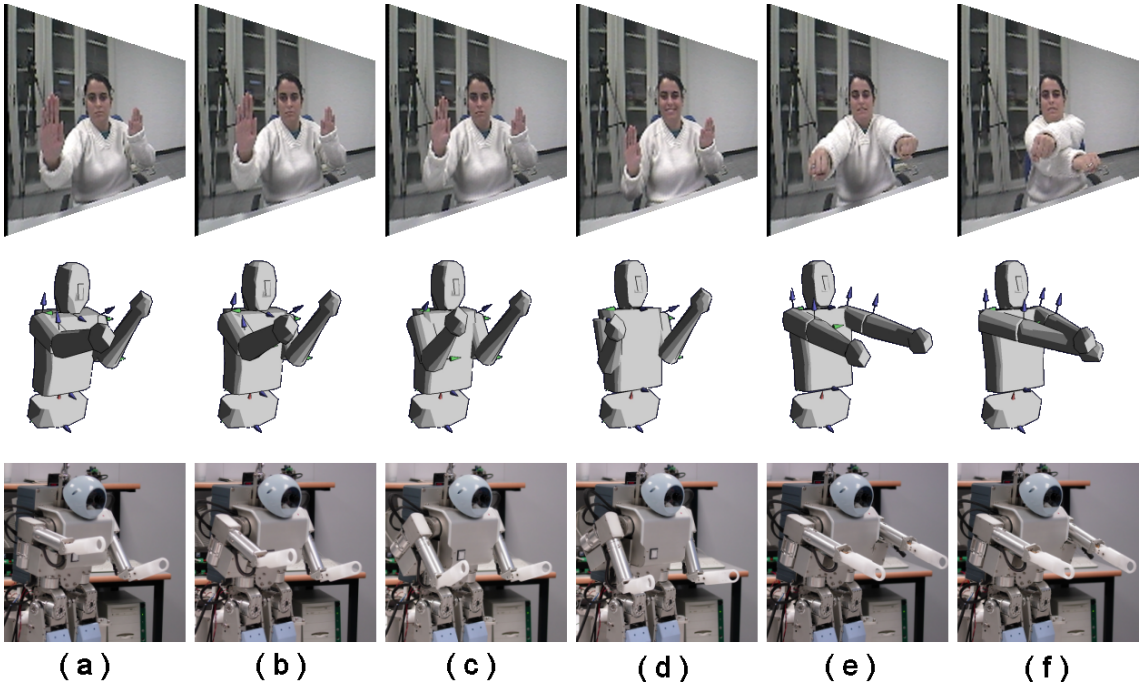

Figure 5.8: Results of upper-body motion estimation. Top row: images captured with the left camera. Middle row: Estimated model pose. Bottom row: Corresponding pose adopted by robot.

motion. The arm-crossing action, correctly estimated by the model-based tracker, cannot be performed by the HOAP-1 due to the shoulder joint limits.

5.3 Summary

Section 1 of this chapter has presented a visual attention mechanism that integrates bottom-up and top-down processing. The proposed mechanism employs two selection stages, providing an additional semiattentive computation stage. Thanks to the use of the tracking algorithm proposed in this Thesis it is possible to handle dynamic environments with deformable moving objects at 5 frames per second. Specifically, the tracking algorithm allows to correctly implement the inhibition of return in order to avoid extracting the same object in different frames.

Section 2 of this chapter has explained a novel human motion capture system which, thanks to the proposed tracking approach, can estimate upper-body human movements in real-time. Results show correct depth estimation of the tracked regions in spite of the short baseline of the robot stereo system.

Chapter 6

Conclusions and Future work

6.1 Conclusions

In this Thesis a tracking system based on a novel approach to target representation and localization has been presented. It uses a hierarchical template-based appearance model. This type of appearance model has been selected due to its capability to track non-rigid objects without a previous learning of different object views. To achieve it, the proposed method employs a weighted template which is dynamically updated in order to follow up the viewpoint and appearance changes of the object to track. Weights are used to establish a compromise between the current template and older templates. Therefore, the weight places more importance to more recent data. Older data are "forgotten" in a linear and smooth manner. This weighted template and the way it is updated allow the algorithm to successfully handle challenging situations, such as:

- Partial and total occlusions of the tracked object: the duration of the total occlusions that the algorithm can handle depends of the value of a user set parameter (α), which determines the degree of "forgetting" of old data.
- Illumination changes: the template can adapt to gradual illumination changes which produce a modification colour in the tracked object which is smaller than the threshold employed in the template matching process.
- Appearance changes of the object due to deformations, zooms, rotations or changes in the view point.
- Presence of other moving objects in the scene.

- Tracking of several objects at the same time: the proposed tracking approach allows to follow up the appearance and position changes of multiple objects. Some problems can appear when two targets with similar colour share the same ROI area because of an occlusion. The proposed approach is not able to track several objects with very similar colour if they occlude each other. This problem can be solved using adequate filtering and data association techniques [55].

Another goal of the proposed method was to run in real-time. To achieve this real-time performance a pyramidal structure has been used. The template and the target have been represented using this pyramid. These representations are generated by segmenting the region of the input frame where the target is likely placed. This segmentation is the most time consuming part of the tracking algorithm. Therefore, in order to achieve real-time performance, the used segmentation approach (pyramidal algorithm) must be as fast as possible. Besides, the pyramid is used to perform the template matching in a hierarchical way. Therefore, the lower the time to traverse the pyramid, the lower the time to perform the template matching. In order to find a pyramidal structure which satisfies these conditions, the main pyramidal structures (regular and irregular) present in the literature were detailedly studied during the development of this Thesis. Both types of pyramids -regular and irregular- have advantages and disadvantages. Regular pyramids can be built and traversed with low computational cost. However, they present important problems due to the inflexibility of their fixed structure. Irregular pyramids solve the problems of the regular ones but with a computational cost which prevents their use in real-time applications. In this Thesis, a new piramidal structure is presented: the Bounded Irregular Pyramid (BIP). The BIP arose due to the necessity of getting an irregular pyramid with similar accurate segmentation results than other irregular pyramids but faster to build and traverse. The key idea behind the BIP is to use a $2x2/4$ regular structure in the homogeneous regions of the input image and a simple graph irregular structure in the rest of regions. The irregular part of the BIP permits to avoid the problems of regular structures and its regular part reduce its computational complexity. The BIP allows the whole tracking system to run in real time with 128x128 pixels images (27 frames per second) in a 3GHz Pentium IV PC. The BIP has proven to achieve similar segmentation results than the other irregular structures but reducing at least ten times the computational time.

The proposed tracking approach has been successfully employed in two real time applications such as an attentional mechanism and a human motion capture system. These applications have been briefly presented in this Thesis, pointing out the contributions of the tracking algorithm. In the attentional mechanism, the tracking is used in the pre-attentive stage to implement

the inhibition of return, avoiding that the same object was extracted in different frames. The tracking algorithm allows the attentional mechanism to handle dynamic environments with deformable moving objects at 5 frames per second. In the human motion capture system, the tracking algorithm is employed to follow up in real time the movements of the head and the hands of the human whose movements are being captured.

6.2 Future work

The tracking method proposed in this Thesis can be improved in some aspects:

- A general problem of colour based tracking approaches is that errors can appear if there is another object similar in colour to the tracked one in the scene which occludes it. This problem could be solved by applying an adequate filtering and data association technique which prevents the algorithm from confusing both objects. Filtering techniques also allows to automatically obtain some of the parameters of the algorithm, such as the initial displacement of the template over the ROI and the ϵ value. If a Kalman filter is used, for example, the initial displacement will be the predicted value by the filter, and ϵ will be the uncertainty in this prediction.

- The proposed tracking approach is a colour based approach in which homogeous colour region of the image are tracked. It can be modified in order to track an object with several colours. The problem is that the algorithm is based on a colour segmentation (over-segmentation step). In this segmentation all the segmented regions do not have vertices in all the levels. Therefore, if an object is found in a level where not all its different coloured regions have representation, it would lose part of its vertices. The simplest way to solve this problem would be to change the conditions to chose the working level. This level could be chose as the higher level where all the the coloured regions of the object have vertices.

- In order to exploit the easiness to compare images, the template matching process included in the proposed tracking system only uses the regular vertices of the hierarchical representation of the ROI. These vertices are compared with the vertices of the template, which are all regular. An interesting future work could be the study of the behaviour of the system when the template is formed by regular and virtual vertices and the template matching process is a comparison between graphs instead of images.

Bibliography

[1] N. Ahuja. On approaches to polygonal decomposition for hierarchical image representation. *Computer Vision Graphics and Image Processing*, 24(2):200–214, November 1983.

[2] R.J. Althof, M.G.J. Wind, and J.T. Dobbins, III. A rapid and automatic image registration algorithm with subpixel accuracy. 16(3):308–316, June 1997.

[3] H. J. Antonisse. Image segmentation in pyramids. *Computer Graphics Image Processing*, 19:367–383, 1982.

[4] M. Arj and E. Vahdati-khajeh. *Target Tracking 2004: Algorithms and Applications*, chapter Problems of multiple-target tracking in vision-based applications, pages 131–134. IEE, 2004.

[5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, 2002.

[6] G. Backer and B. Mertsching. Two Selection Stages Provide Efficient Object-based Attentional Control for Dynamic Vision. In *Proc. Int. Workshop Attention and Performance in Computer Vision (WAPCV 2003)*, pages 9–16, Graz, Austria, April 2003.

[7] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[8] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451–460, 1975.

[9] S. Baronti, A. Casini, F. Lotti, L. Favaro, and V. Roberto. Variable pyramid structures for image segmentation. *Computer Vision Graphics and Image Processing*, 49(3):346–356, March 1990.

[10] P. Bertolino and A. Montanvert. Multiresolution segmentation using the irregular pyramid. In *International Conference on Image Processing*, pages 257–260, Lausanne, Switzerland, 1996.

[11] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages 232–237, Santa Barbara - CA, USA, 1998.

[12] M. Bister, J. Cornelis, and A. Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11:605–617, 1990.

[13] M.J. Black and A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, January 1998.

[14] A. Blake, M. Isard, and D. Reynard. Learning to track the visual-motion of contours. *Artificial Intelligence*, 78(1-2):179–212, October 1995.

[15] M. Borsotti, P. Campadelli, and R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19(8):741–747, June 1998.

[16] Y. Boykov and D.P. Huttenlocher. Adaptive bayesian recognition in tracking rigid objects. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages II: 697–704, Hilton Head, USA, 2000.

[17] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages 8–15, Santa Barbara - CA, USA, 1998.

[18] L. Brun and W.G. Kropatsch. Combinatorial pyramids. In *International Conference on Image Processing*, pages II: 33–36, Barcelona, Spain, 2003.

[19] L. Brun and W.G. Kropatsch. Receptive fields within the combinatorial pyramid framework. *Graphical Models*, 65(1-3):23–42, May 2003.

[20] M. G. S. Bruno. Bayesian methods for multiaspect target tracking in image sequences. *IEEE Transactions on Signal Processing*, 52(7):1848–1861, July 2004.

[21] J.M. Buenaposada, E. Munoz, and L. Baumela. Efficient appearance-based tracking.

[22] M. Burge and W. G. Kropatsch. A minimal line property preserving representation of line images. *Computing*, 62(4):355–368, 1999.

[23] P.J. Burt. Tree and pyramid structures for coding hexagonally sampled binary images. *Computer Graphics Image Processing*, 14(3):271–280, November 1980.

[24] P.J. Burt, T.H. Hong, and A. Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Trans. Systems, Man and Cybernetics*, 11(12):802–809, December 1981.

[25] J. Cadre C. Hue and P. Perez. Sequential monte carlo filtering for multiple target tracking and data fusion. *IEEE Trans. on Signal Processing*, 50(2):309–325, February 2002.

[26] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, February 1997.

[27] A. Cavallaro, O. Steiger, and T. Ebrahimi. Tracking video objects in cluttered background. *IEEE Transactions On Circuits and Systems for Video Technology*, 15(4):575–584, April 2005.

[28] J.M. Chassery and A. Montavert. A segmentation method in a voronoi diagram environment. In *Sixth Scandinavian Conference on Imaging Processing*, pages 408–415, Oulu, Finland, 1989.

[29] P.C. Chen and T. Pavlidis. Image segmentation as an estimation problem. *Computer Graphics Image Processing*, 12:153–172, 1980.

[30] Y. Chen, T.S. Huang, and Y. Rai. Parametric contour tracking using unscented kalman filter. In *International Conference on Image Processing*, pages III: 613–616, Rochester - NY, USA, 2002.

[31] H.D. Cheng and Y. Sun. A hierarchical approach to color image segmentation using homogeneity. *IEEE Trans. Image Processing*, 9(12):2071–2082, December 2000.

[32] K.J. Cho and P. Meer. Image segmentation from consensus information. *Computer Vision and Image Understanding*, 68(1):72–89, October 1997.

[33] J.M. Cibulskis and C.R. Dyer. An analysis of node linking in overlapped pyramids. *IEEE Trans. Systems, Man and Cybernetics*, 14:424–436, 1984.

[34] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[35] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision*, pages II: 484–498, Freiburg, Germany, 1998.

[36] I.J. Cox and S.L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(2):138–150, February 1996.

[37] P. Dani and S. Chaudhuri. Automated assembling of images: Image montage preparation. *Pattern Recognition*, 28(3):431–445, March 1995.

[38] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, July 2000.

[39] Q. Delamarre and O.D. Faugeras. 3d articulated models and multiview tracking with physical forces. *Computer Vision and Image Understanding*, 81(3):328–357, March 2001.

[40] F. Dornaika and J. Ahlberg. Fast and reliable active appearance model search for 3-d face tracking. *IEEE Trans. Systems, Man and Cybernetics*, 34(4):1838–1853, August 2004.

[41] G.J. Edwards, T.F. Cootes, and C.J. Taylor. Face recognition using active appearance models. In *European Conference on Computer Vision*, pages II: 581–595, Freiburg, Germany, 1998.

[42] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.

[43] P.F. Felzenszwalb and D.P. Huttenlocher. Image segmentation using local variation. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages 98–104, Santa Barbara - CA, USA, 1998.

[44] R. Feris, V. Krüger, and R. Cesar Jr. Efficient real-time face tracking in wavelet subspace. In *Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 113–118, Vancouver - BC, Canada, 2001.

[45] D.A. Forsyth and J. Ponce. *Computer Vision: a modern approach*. Prentice Hall, 2003.

[46] J. Gao, A. Kosaka, and A.C. Kak. A multi-kalman filtering approach for video tracking of human-delineated objects in cluttered environments. *Computer Vision and Image Understanding*, 99(1):1–57, July 2005.

[47] D.M. Gavrila and L.S. Davis. 3d model-based tracking of humans in action: A multi-view approach. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages 73–80, San Francisco - CA, USA, 1996.

[48] G. Gennari, A. Chiuso, F. Cuzzolin, and R. Frezza. Integrating shape and dynamic probabilistic models for data association and tracking. In *41st IEEE Conference on Decision and Control*, pages 2409–2414, Adelaide, South Australia, 2002.

[49] V. Girondel, A. Caplier, and L. Bonnaud. Real time tracking of multiple persons by kalman filtering and face pursuit for multimedia applications. In *IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 201–205, Lake Tahoe - NE, USA, 2004.

[50] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Trans. Image Processing*, 10(10):1467–1475, October 2001.

[51] J.J. Gonzalez, I.S. Lim, P. Fua, and D. Thalmann. Robust tracking and segmentation of human motion in an image sequence. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 29–32, Hong Kong, 2003.

[52] A. Griffin and J.V. Kittler. An active mesh based tracker for improved feature correspondences. *Pattern Recognition Letters*, 23(4):443–449, February 2002.

[53] A.D. Gross and A. Rosenfeld. Multiresolution object detection and delineation. *Computer Vision Graphics and Image Processing*, 39(1):102–115, July 1987.

[54] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.

[55] B. Hang, C. Yang, R. Duraiwani, and L. Davis. Bayesian filtering and integral image for visual tracking. In *6th Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005)*, Motreux, Switzerland, April 2005.

[56] Y. Haxhimusa, R. Glantz, and W.G. Kropatsch. Constructing stochastic pyramids by mides - maximal independent directed edge set. *Lecture Notes in Computer Science*, 2726:24–34, 2003.

[57] Y. Haxhimusa, R. Glantz, M. Saib, G. Langs, and W.G. Kropatsch. Logarithmic tapering graph pyramid. In *German Pattern Recognition Symposium*, pages 117–124, Zürich, Switzerland, 2002.

[58] Y. Haxhimusa and W.G. Kropatsch. Segmentation graph hierarchies. *Lecture Notes in Computer Science*, 3138:343–351, 2004.

[59] J.A. Hird and D.F. Wilson. *Optical Systems for Space and Defense*, chapter A comparison of target detection and segmentation techniques, pages 375–386. Number 1191. SPIE, Cambridge - MA, USA, 1989.

[60] J. Ho, K.C. Lee, M.H. Yang, and D.J. Kriegman. Visual tracking using learned linear subspaces. pages I: 782–789, Washington D.C., USA, 2004.

[61] T.H. Hong, K.A. Narayanan, S. Peleg, A. Rosenfeld, and T.M. Silberberg. Image smoothing and segmentation by multiresolution pixel linking: Further experiments and extensions. *IEEE Trans. Systems, Man and Cybernetics*, 12(5):611–622, September 1982.

[62] T.H. Hong and A. Rosenfeld. Compact region extraction using weighted pixel linking in a pyramid. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(2):222–229, March 1984.

[63] J. Huart and P. Bertolino. Similarity-based and perception-based image segmentation. In *International Conference on Image Processing*, pages III: 1148–1151, Genova, Italy, 2005.

[64] M. H. Gokcetekin I. Celasun, A. M. Tekalp and D. M. Harmanci. 2-d mesh-based video object segmentation and tracking with occlusion resolution. *Signal Processing: Image Communication*, 16(10):949–962, 2001.

[65] N. Ikonomakis, K.N. Plataniotis, and A.N. Venetsanopoulos. *Visual Communications and Image Processing*, chapter A region-based colour segmentation scheme, pages 1202–1209. Number 3653. SPIE, 1999.

[66] A. Ion, Y. Haxhimusa, W.G. Kropatsch, and L. Brun. Hierarchical image partitioning using combinatorial maps. In *10*.

[67] H.H.S. Ip and S.W.C. Lam. Alternative strategies for irregular pyramid construction. *Image and Vision Computing*, 14(4):297–303, May 1996.

[68] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, February 1994.

[69] M. Isard and A. Blake. C-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998.

[70] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, October 2003.

[71] J.M. Jolion. Stochastic pyramid revisited. *Pattern Recognition Letters*, 24(8):1035–1042, May 2003.

[72] J.M. Jolion and A. Montanvert. The adaptive pyramid: A framework for 2d image analysis. *Computer Vision Graphics and Image Processing*, 55(3):339–348, May 1992.

[73] S. Julier and J. Uhlmann. *Signal Processing, Sensor Fusion, and Target Recognition VI*, chapter A new extension of the Kalman filter to nonlinear systems, pages 1202–1209. Number 3068. SPIE, 1997.

[74] I.A. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, December 2000.

[75] H.G. Kang and D. Kim. Real-time multiple people tracking using competitive condensation. *Pattern Recognition*, 38(7):1045–1058, July 2005.

[76] M. Kass, A.P. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

[77] W. Kim, C. Lee, and J. Lee. Tracking moving object using snake's jump based on image flow. *Mechatronics*, 11:199–226, 2001.

[78] D. Koller, K. Daniilidis, and H.H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, June 1993.

[79] W.G. Kropatsch. A pyramid that grows by powers of 2. *Pattern Recognition Letters*, 3:315–322, 1985.

[80] W.G. Kropatsch. Building irregular pyramids by dual-graph contraction. *IEE Proceedings-Vision Image and Signal Processing*, 142(6):366–374, December 1995.

[81] W.G. Kropatsch. *Digital image processing and computer graphics: Applications in humanities and natural sciences*, chapter From equivalent weighting functions to equivalent contraction kernels, pages 310–320. Number 3346. SPIE, 1998.

[82] W.G. Kropatsch and Y. Haxhimusa. Grouping and segmentation in a hierarchy of graphs. In *IT&T SPIE Annual Symposium, Computational Imaging*, pages 147–158, San Jose - CA, USA, 2004.

[83] W.G. Kropatsch, Y. Haxhimusa, Z. Pizlo, and G. Langs. Vision pyramids that do not grow too high. *Pattern Recognition Letters*, 26(3):319–337, February 2005.

[84] W.G. Kropatsch and H. Macho. Finding the structure of connected components using dual irregular pyramids. In *Cinquieme Colloque DGCI*, pages 147–158, Clermont-Ferrand, France, 1995.

[85] V. Kruger, A. Happe, and G. Sommer. Affine real-time face tracking using gabor wavelet networks. In *International Conference on Pattern Recognition*, pages Vol I: 127–130, Barcelona, Spain, 2000.

[86] S. Lallich, F. Muhlenbach, and J.M. Jolion. A test to control a region growing process within a hierarchical graph. *Pattern Recognition*, 36(10):2201–2211, October 2003.

[87] F.F. Leymarie and M.D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):617–634, June 1993.

[88] X. Li and N. Zheng. Adaptive target color model updating for visual tracking using particle filter. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 3105–3109, The Hague, Netherlands, 2004.

[89] J.Q. Liu and Y.H. Yang. Multiresolution color image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(7):689–700, July 1994.

[90] T. Liu and H. Chen. Real-time tracking using trust-region methods. *PAMI*, 26(3):397–402, 2003.

[91] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision.

[92] F. Marques and J. Llach. Tracking of generic objects for video object generation. pages 628–632, Chicago - IL, USA, 1998.

[93] F. Martin and R. Horaud. Multiple-camera tracking of rigid objects. *International Journal of Robotics Research*, 21(2):97–113, 2002.

[94] J. Martin, V. Devin, and J.L. Crowley. Active hand tracking. In *International Conference on Automatic Face and Gesture Recognition*, pages 573–578, Nara, Japan, 1998.

[95] O. Masoud and N. P. Papanikolopoulos. A novel method for tracking and counting pedestrians in realtime using a single camera. *IEEE Trans. on Vehicular Technologies*, 50:1267–1278, 2001.

[96] S.J. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3/4):225–231, March 1999.

[97] P. Meer. Stochastic image pyramids. *Computer Vision Graphics and Image Processing*, 45(3):269–294, March 1989.

[98] P. Meer, C.A. Sher, and A. Rosenfeld. The chain pyramid: Hierarchical contour processing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(4):363–376, April 1990.

[99] I. Mikic, M.M. Trivedi, E. Hunter, and P.C. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, July 2003.

[100] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(4):307–316, April 1991.

[101] P.A.P. Moran. The interpretation of statistical maps. *Journal of the Royal Statistical Society*, Series B:246–251, 1948.

[102] P.F.M. Nacken. Image segmentation by connectivity preserving relinking in hierarchical graph structures. *Pattern Recognition*, 28(6):907–920, June 1995.

[103] H.T. Nguyen and A.W.M. Smeulders. Template tracking using color invariant pixel features. In *International Conference on Image Processing*, pages I: 569–572, Rochester - NY, USA, 2002.

[104] H.T. Nguyen and A.W.M. Smeulders. Fast occluded object tracking by a robust appearance filter. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(8):1099–1104, August 2004.

[105] H.T. Nguyen, M. Worring, and R. van den Boomgaard. Occlusion robust adaptive template tracking. In *International Conference on Computer Vision*, pages I: 678–683, Vancouver - BC, Canada, 2001.

[106] K. Nummiaro, E. Koller-Meier, and L.J. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, January 2003.

[107] J. O'Rourke and N.I. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(6):522–536, November 1980.

[108] S. Peleg, O. Federbusch, and R. Hummel. chapter Custom-Made Pyramids, pages 125–146. Academic Press Professional, San DIego - CA, USA, 1987.

[109] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, pages I: 661–675, Copenhagen, Denmark, 2002.

[110] N. Peterfreund. The velocity snake. In *IEEE Nonrigid and Articulated Motion Workshop*, pages 70–79, San Juan, Puerto Rico, 1997.

[111] N. Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(6):564–569, June 1999.

[112] M. Pietikainen and A. Rosenfeld. Gray level pyramid linking as an aid in texture analysis. *IEEE Trans. Systems, Man and Cybernetics*, 12:422–429, 1982.

[113] Y. Ping, W. Runsheng, and L. Diannong. A new image segmentation approach based on linked pyramid. In *International Conference on Signal Processing*, pages 1118–1121, San Francisco - CA, USA, 1996.

[114] A.R. Pope, J.C. Asmuth, R. Kumar, H.S. Sawhney, and S. Hsu. Registration of video to geo-referenced imagery. In *International Conference on Pattern Recognition*, pages Vol II: 1393–1400, Brisbane, Australia, 1998.

[115] D. Prewer and L. Kitchen. Soft image segmentation by weighted linked pyramid. *Pattern Recognition Letters*, 22(2):123–132, February 2001.

[116] C. Rasmussen and G.D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):560–576, June 2001.

[117] J.M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *International Conference on Computer Vision*, pages 612–617, Cambridge - MA, USA, 1995.

[118] D.B. Reid. An algorithm for tracking multiple targets. 24(6):843–854, December 1979.

[119] E.M. Riseman and A.R. Hanson. Design of a semanitcally directed vision processor. Technical report, University of Massachusets, Amherst, Massachusets, 1974.

[120] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision Graphics and Image Processing*, 59(1):94–115, January 1994.

[121] H. Rom and S. Peleg. Image representation using voronoi tessellation: Adaptive and secure. In *IEEE Computer Vision and Pattern Recognition*, Los Angeles -CA, USA, 1988.

[122] A. Rosenfeld. Some pyramid techniques for image segmentation. Technical report, Technical Report CAR-TR-203, Center for Automation Research, University of Maryland at College Park, 1986.

[123] A. Rosenfeld and G.J. van der Brug. Coarse-fine template matching. *IEEE Trans. Systems, Man and Cybernetics*, 7:104–107, February 1977.

[124] W.J. Rucklidge. Efficient guaranteed search for gray-level patterns. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages 717–723, San Juan, Puerto Rico, 1997.

[125] B. Sabata, F. Arman, and J.K. Aggarwal. Convergence of fuzzy-pyramid algorithms. *Journal of Mathematical Imaging and Vision*, 4:291–302, 1994.

[126] S. Sclaroff and J. Isidoro. Active blobs: region-based, deformable appearance models. *Computer Vision and Image Understanding*, 89(2-3):197–225, February 2003.

[127] D. Scott. *Multivariate density estimation*. Wiley, 1992.

[128] K. Seo, J. Lee, and J. Lee. Adaptive color snake tracker using condensation algorithm.

[129] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(1):51–65, January 2005.

[130] Y. Shinagawa and T.L. Kunii. Unconstrained automatic image matching using multiresolutional critical point filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(9):994–1010, September 1998.

[131] M.O. Shneier. Extracting linear features from images using pyramids. *IEEE Trans. Systems, Man and Cybernetics*, 12(4):569–572, July 1982.

[132] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, pages II: 702–718, Dublin, Ireland, 2000.

[133] M. Singh, M.K. Mandal, and A. Basu. Gaussian and laplacian of gaussian weighting functions for robust feature based tracking. *Pattern Recognition Letters*, 26(13):1995–2005, October 2005.

[134] S.M. Smith and J.M. Brady. Asset-2: Real-time motion segmentation and shape tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(8):814–820, August 1995.

[135] M. Spann, C. Horne, and H. du Buf. The detection of thin structures in images. *Pattern Recognition Letters*, 10(3):175–179, September 1989.

[136] C. Sun. Fast stereo matching using rectangular subregioning and 3d maximum-surface techniques. *International Journal of Computer Vision*, 47(1-3):99–117, April 2002.

[137] S. Sun, D.R. Haynor, and Y. Kim. Semiautomatic video object segmentation using vsnakes. *IEEE Transactions On Circuits and Systems for Video Technology*, 13(1):75–82, January 2003.

[138] T.N. Tan, G.D. Sullivan, and K.D. Baker. Model-based localization and recognition of road vehicles. *International Journal of Computer Vision*, 27(1):5–25, March 1998.

[139] S.L. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *Computer Graphics Image Processing*, 4(2):104–119, June 1975.

[140] H. Tao, H.S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(1):75–89, January 2002.

[141] R.E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22:215–225, 1975.

[142] J.C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the detection of human faces in color images. In *International Conference on Automatic Face and Gesture Recognition*, pages 54–61, Dublin, Ireland, 2000.

[143] P. Thevenaz, U.E. Ruttimann, and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Trans. Image Processing*, 7(1):27–41, January 1998.

[144] P. Tissainayagam and D. Suter. Visual tracking with automatic motion model switching. *Pattern Recognition*, 34(3):641–660, March 2001.

[145] P. Tissainayagam and D. Suter. Contour tracking with automatic motion model switching. *Pattern Recognition*, 36(10):2411–2427, October 2003.

[146] P. Tissainayagam and D. Suter. Object tracking in image sequences using point features. *Pattern Recognition*, 38(1):105–113, January 2005.

[147] C. Toklu, A.M. Tekalp, A.T. Erdem, and M.I. Sezan. 2d mesh based tracking of deformable objects with occlusion. In *International Conference on Image Processing*, page 17A2, Lausanne, Switzerland, 1996.

[148] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.

[149] C. Tomasi and J. Shi. Good features to track. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages 593–600, Seattle - WA, USA, 1994.

[150] A.M. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[151] Y. Tsaig and A. Averbuch. Automatic segmentation of moving objects in video sequences: A region labeling approach. *IEEE Transactions On Circuits and Systems for Video Technology*, 12(7):597–612, July 2002.

[152] D.C. Tseng and C.H. Chang. Color segmentation using perceptual attributes. In *International Conference on Pattern Recognition*, pages 228–231, The Hague, Netherlands, 1992.

[153] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *International Symposium on Mixed and Augmented Reality*, pages 48–57, Arlington - VA, USA, 2004.

[154] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, October 2004.

[155] P. van Beek, A.M. Tekalp, N. Zhuang, I. Celasun, and M. Xia. Hierarchical 2-d mesh representation, tracking, and compression for object-based video. *IEEE Transactions On Circuits and Systems for Video Technology*, 9(2):353, March 1999.

[156] G.J. van der Brug and A. Rosenfeld. Two-stage template matching. *IEEE Trans. Computer*, 26(4):384–393, April 1977.

[157] C.J. Veenman, M.J.T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(1):54–72, January 2001.

[158] C.J. Veenman, M.J.T. Reinders, and E. Backer. Establishing motion correspondence using extended temporal scope. *Artificial Intelligence*, 145(1-2):227–243, April 2003.

[159] C.J. Veenman, M.J.T. Reinders, and E. Backer. Motion tracking as a constrained optimization problem. *Pattern Recognition*, 36(9):2049–2067, September 2003.

[160] S. Wachter and H.H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, June 1999.

[161] K.N. Walker, T.F. Cootes, and C.J. Taylor. Automatically building appearance models from image sequences using salient features. *Image and Vision Computing*, 20(5-6):435–440, April 2002.

[162] R. Wildes, R. Kumar, H. Sawhney, S. Samasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, D. Hirvonen, M. Hansen, and P. Burt. Aerial video surveillance and exploitation. *Proceedings of the IEEE*, 89(10):1518–1539, 2001.

[163] D. Willersinn and W.G. Kropatsch. Dual graph contraction for irregular pyramids. In *International Conference on Pattern Recognition*, pages 251–256, Jerusalem, Israel, 1994.

[164] R.Y. Wong and E.L. Hall. Sequential hierarchical scene matching. *IEEE Trans. Computer*, 27:359–366, 1978.

[165] Y. Yoon, A. Kosaka, J. B. Park, and A. C. Kak. A new approach to the use of edge extremities for model-based object tracking. In *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[166] Y.J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, August 1996.

[167] B. Zhou and N.K. Bose. Multitarget tracking in clutter: fast algorithms for data association. *IEEE Transactions on aerospace and electronic systems*, 29(2):352–362, 1993.

[168] S.K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Processing*, 13(11):1491–1506, November 2004.

[169] F. Ziliani and B. Jensen. Unsupervised image segmentation using the modified pyramidal linking approach. Chicago - IL, USA, 1998.

Appendix A

HSV color space

HSI and HSV colour spaces were designed to approximate the ways humans perceive and interpret color. These systems separates color information, which is represented by H -hue- and S -saturation- values, from the image brightness, which is determined by the I and V values, respectively. Specifically, the hue value represents the color and the saturation value is a measure of the purity of the color. The difference between HSI and HSV is the computation of the brightness component. In both color spaces, a value of 0 represents the absence of light, or black. In HSV space, a maximum value means that the color is at its brightest. In HSI space, a maximum value for lightness means that the color is white, regardless of the current values of the hue and saturation components. The HSV values can be transformed from the standard RGB coordinates using well-known transformation formulas:

$$\begin{aligned} H &= arctan(\tfrac{\sqrt{3}(G-B)}{(R-G)+(R-B)}) \\ S &= 1-3\tfrac{min(R,G,B)}{R+G+B} \\ V &= max(R,G,B) \end{aligned} \tag{A.1}$$

The transformation to a hue, saturation and brightness coordinate system places a new axis that passes through all the achromatic or grey values (i.e. with R=G=B), and it specifies the color in terms of cylindrical coordinates based on this achromatic axis. The brightness or V value gives the coordinate of a color on this axis, hue H is measured by the angle around the axis and the saturation S corresponds to the distance from the axis. Although the natural shape of the HSV space is a cone, this space is artificially expanded into a cylinder by dividing the saturation value by its maximum possible value for the corresponding brightness. The cylinder shape permits to avoid complicated verification of the validity of a specified color. The cylindrical HSV color model is shown in Fig. A.1.

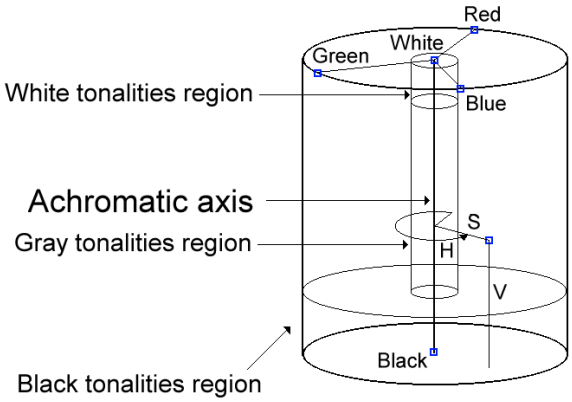

Figure A.1: Cylindrical HSV colour model.

Figure A.2: Color image.

Because of its relevance, the most of the colour segmentation algorithms only use hue as region descriptor [31]. However, these methods cannot separate in different regions two objets which present distinct tonalities but the same hue. For example, the box and the yelow torch in Fig. A.2 have the same hue value. In this Thesis, the cylindrical metric which was presented by Tseng and Chang [152] has been adopted to compute the distance measure between two colours. If i and j are two colour, the cylindrical distance $d(i,j)$ is defined as follows:

$$d(i,j) = \sqrt{(d_V(i,j))^2 + (d_C(i,j))^2} \tag{A.2}$$

where

$$d_V(i,j) = |V_i - V_j| \tag{A.3}$$

and

$$d_C(i,j) = \sqrt{(S_i)^2 + (S_j)^2 - 2S_iS_jcos\theta} \tag{A.4}$$

θ is equal to $|H_i - H_j|$ if this value is less than π. In other case, θ is equal to $(2\pi - |H_i - H_j|)$.

Appendix B

Segmentation algorithms using pyramids

B.1 Segmentation algorithms based on regular pyramids

Chen and Pavlidis [29] proposed the first pyramidal segmentation algorithm. In this approach, they define son-parent edges, which constitute vertical relationships in the pyramid, and brother-brother edges (horizontal relationships). The rigidity of this pyramid structure may give rise to artefacts [53, 62]. Specially, the difficulty of handling long shaped features in an image was closely related to the limitations of image pyramids [97]. To compensate for these artefacts, different regular pyramids were proposed. Thus, Shneier [131] focused its research on extraction of linear features from an image. However, Shneier only uses multi-resolution images to define local thresholds in a classical local thresholding method, which is not really a pure pyramid segmentation technique. Other approaches control the resolution reduction by the local image content [24, 61]. These approaches recalculate the son-parent relationships in order to achieve the adaptation of the pyramidal structure to the image layout. Particularly, the son-parent relationships are refined over several iterations, so these approaches are named global iterative approaches [59]. A typical iteration may consist of a bottom-up linking process, a bottom-up recomputation of vertex values and a top-down reassignment of vertex values. After several iterations, the inter-level edges will normally have stabilised and the segmented image is obtained from the base level vertex values. Although these global iterative approaches to pyramidal segmentation exhibit superior performance over the classical top-down approaches, this performance advantage must be considered against the greater computational requirements of the iterative algorithms [59]. Still, global iterative approaches can be considered as the main type of regular pyramidal structures and are explained below.

In other pyramidal approaches, the pyramid is built using several types of Gaussian filters. Ping et *al*. [113] use a Gaussian filter function with changeable filter scales. By modifying the filter scale, this algorithm changes the window size between pyramidal levels. When applied to segmentation, this algorithm searches for vertices in the structure that can be regarded as roots of segmented regions at the base level. Regular pyramids normally employ a square window but there are regular structures with triangular and hexagonal windows [23, 1]. Another possible modification consists in reducing the size only by half between pyramidal levels [79].

B.1.1 Pyramid Linking Approach (PLA)

Burt et *al*. [24] originally proposed the Linked Pyramid in 1981. In this pyramid, during the first iteration, each 4x4 set of vertices within a level generates a new vertex in the upper level by averaging the local image property values of the vertices in the reduction window (4x4/4 pyramid). Each level has a size four times smaller than the level below, because the 4x4 windows are 50% overlapped, as it is shown in Fig. B.1a. For each vertex at level l there is a 4x4 subarray of "candidate son" vertices at level l-1 (Fig. B.1b). The vertex itself is a member of four such subarrays for level l+1 vertex. On each iteration, the whole of the structure is covered and every vertex is linked to the most similar candidate parent from the higher level (Fig. B.1c). After linking, each vertex will have between 0 and 16 legitimate sons. The local image property value of each parent is recalculated by averaging the local image property of its sons. This process continues until the son-parent edges do not vary. Finally, in order to perform the image segmentation, a level of the pyramid (called working level) is selected as the level in charge to generate the segmentation. Each working level vertex is linked to a set of vertices at the base of the structure. These vertices represent its receptive field and define a segmented region. The local image property values from the working level vertices are propagated to their corresponding regions at the base. These regions constitute the segmented image. The selection of the working level is very important because it sets the number of resulting segmented regions, which is approximately equal to the number of vertices at the working level (there could exist vertices at the working level with null receptive field). It must be noted that the correct working level depends on the content of the image to segment, being unknown at the beginning of the process. The accuracy of the final segmentation depends on the correct selection of the working level. Because of its apparent flexibility, this adaptive hierarchical structure has been investigated by other researchers [135, 9].

The linked pyramid, as originally proposed by Burt et *al*. [24] presents four main problems. The first is the aforementioned need of choosing the working level. The other three are

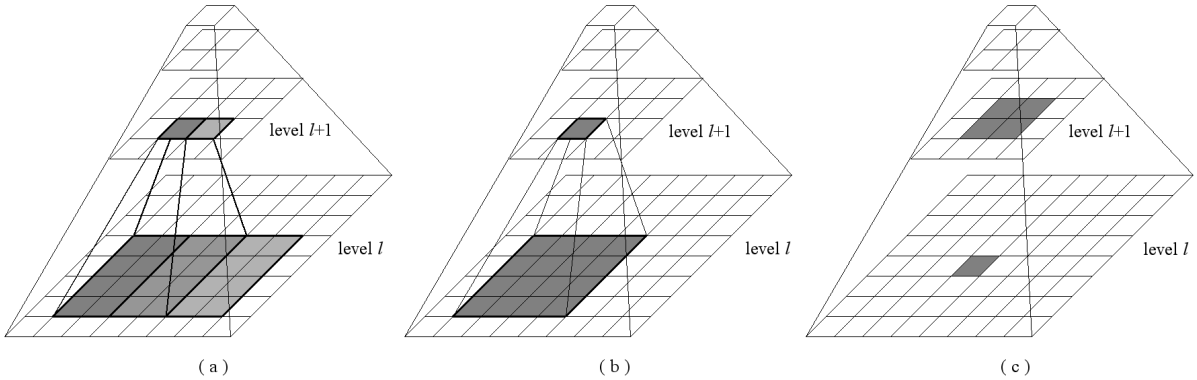

Figure B.1: Linked pyramid: a) overlapped at the linked pyramid; b) the sixteen grey vertices in level l are the candidate sons for the grey vertex in level l+1; c) the four grey vertices in level l+1 are the candidate parents for the grey vertex in level l.

related to the inflexibility of the structure [12]:

- The region connectivity is not preserved: in the son-parent relinking process, this structure does not take into account adjacency criteria in the original image; hence, adjacent nodes in a level do not necessarily originate adjacent segmented regions at the base level. When these vertices are grouped into a new vertex in the upper level, the new vertex is associated to a disjoint region at the base.
- Non-adaptability to the image layout: due to the use of a fixed size rectangular reduction window, the shape of elongated regions can not be represented in the segmented image.
- The structure is variant to small shifts, rotations or scale modifications in the original image. This problem is commonly named the shift variance problem.

On the other hand, this pyramid does not need any threshold to compute similarity between nodes. This is probably its main advantage.

It must be noted that the notion of working level is not mandatory. Burt *et al.* in [24] defined the final segmentation by pre-selecting the final level in the linking process (the working level). The working level has to be defined manually and determines the approximate number of final segmented regions obtained. Large regions with similar homogeneous local image property value usually persist through the linking process up to the highest levels, whereas smaller homogeneous regions may merge with their surroundings in a lower level. The need for pre-selecting the final level can be removed by introducing unforced linking [3] which allows

the exclusion of some vertices from the linking scheme. Thus, region roots can be selected in the pyramid at different levels if the set of receptive fields forms a partition of the initial image. This exclusion procedure, referred as seeding, allows the number of segmented regions to automatically adapt to the image content, which results in extraction of small homogeneous regions as well as large ones. Seeding rules determine whether a vertex is excluded and, therefore, they have a great influence on the result of the segmentation.

B.1.2 Modified Pyramid Linking Approach (MPLA)

Ziliani and Jensen [169] modified the classic pyramid linking approach to avoid the generation of disjoint segmented regions and the selection of a working level. The goal of the modified pyramidal linking approach (MPLA) was to achieve spatial consistency and to obtain a parameters free algorithm. Other difference with previous works [112, 33] is that the redefinition of the son-parent relationships is carried out consecutively between two levels before computing the initial values of the consecutive level. This modification increases the processing speed. In order to achieve spatial consistency, the vertices linking process is restricted to vertices that represent adjacent regions at the base level. To do that, Ziliani and Jensen [169] define a neighbourhood for each vertex. This neighbourhood specifies which vertices in the same level cover neighbouring areas in the base level. The algorithm presents a problem in the first iteration because it has not neighbourhood information yet. To avoid the selection of a working level, the algorithm uses two seeding rules. The island seeding rule assumes that a vertex that covers an entire region is surrounded by vertices with artificial features due to the overlapping of windows. Based on the neighbourhood information defined previously, the algorithm assumes a vertex to be an island if it has only one neighbour. In addition to this exclusion rule, the "parent-seeding" rule excludes all vertices for which cannot an adjacent parent to link to cannot be found, because this indicates that not similar vertex is available.

Although the modified linking approach avoids the selection of the working level and achieves spatial consistency in the segmentation, it kept the other important problems of the linked pyramid: shift variance and non-adaptability to the image layout.

B.1.3 Weighted Linked Pyramidal Segmentation Approaches

Hong et *al.* [61] developed this type of regular pyramid. The structure is similar to the Linked Pyramid but each vertex retains all the edges with its parents, so every vertex has four parent-edges, one for each parent. Every edge carries a weight value that depends on the son-parent

similarity. The value of a parent vertex is calculated as the average of its 16 sons, weighting each son value with its corresponding edge weight. Edge weights are recalculated at each iteration based on the new values of the vertices. Once the pyramid has converged, the final segmentation is achieved by using the edge with the highest weight for each vertex. The use of weights and the retention of all edges avoid forced choice in the edge modification process. This structure obtains slightly better results than the Linked one, as shown in [115]. However it also shares its rigidity problems.

Depending on the type of weights used, there are two different kinds of Weighted Linked pyramids [115]: Fuzzy Linked Pyramid and Possibilistic Linked Pyramid. In the Fuzzy Linked Pyramid [61, 125] weights are always positive and the parent edges of a vertex sum to one. In these algorithms, each vertex has only a parent edge with a value close to one after each iteration, while the rest of parent edges have a value close to zero. In these structures, it is fairly natural to use only the parent edge with the largest weight to define the preferred region for each vertex and thus perform the final segmentation of the image.

The Possibilistic Linked Pyramid [62] uses a non-normalized set of weights. Thus, some vertices link strongly with multiple parents, some link moderately with multiple parents, while some link only feebly to all their parents. Hong and Rosenfeld [62] continue to use the parent edges with the largest weight to define the final segmentation. However, this structure can be used to perform a different type of segmentation: the soft segmentation [115]. In contrast with classical segmentation, called crisp segmentation, in the soft segmentation each pixel can belong to more than one region. This segmentation avoids mistakes in region boundaries. Pixels near region boundaries usually are intermediate in value between the regions, and they can be placed in either of them during the crisp segmentation. In the soft segmentation these pixels belong to both regions. Prewer and Kitchen [115] perform the soft segmentation looking at the tree of possibilistic edge weights as a fuzzy decision tree. They determine a membership value for each of the vertices below the root by using a minimax approach, where each path to the root vertex has assigned the minimum value of the weights on that path, and each vertex takes the maximum value of its paths to the root as its degree of membership of that root. After this, each of the base level vertices has a membership value for each of the regions to which it links.

B.2 Irregular pyramid-based segmentation approaches

B.2.1 Segmentation with a hierarchy of Region Adjacency Graphs (RAG) and the adaptive pyramid

The simple graph hierarchy and the stochastic decimation procedure supposed a great novelty for hierarchical processing. These tools permitted a hierarchical data structure to adapt itself to the image layout, since the proposed hierarchy was not restricted to a rigid sampling structure. The stochastic decimation procedure was successfully applied to multi-scale smoothing of chain-coded curves [98] and segmentation of grey level images [100]. In this last case, a hierarchy of region adjacency graphs (RAG) is generated. The RAG hierarchy performs the stochastic decimation within classes. These classes or similarity subgraphs must be generated before graph contraction is made and they are derived from the RAG by local decisions. Thus, contrary to the original stochastic decimation idea, the resulting decimation procedure is dependent on the image data.

The algorithm works as follows:

1. Graph G_0 is defined by the 8-connected square sampling grid on the level 0, where each vertex is a pixel of the original image.
2. Classes at level l are defined. To do that, each vertex v_i at level l has associated a value g_i characterizing its region of the image (e.g. average grey level). For each v_i, which does not belong to any class yet, every neighbour v_j is examined and a decision is made on whether or not it belongs to the same class of v_i. This decision is based on g_i and g_j values, which are compared using a similarity function.
3. Surviving vertices of every class are chosen applying the stochastic decimation algorithm into the class.

In order to define the classes, several approaches have been experimentally proven [100]. The simplest approach is to define class membership by thresholding the grey level differences between a vertex and their neighbours. This symmetric class membership criterion does not achieve satisfactory results because it strongly influences the structure of the hierarchy and therefore the final segmentation of the image. To overcome this problem, a non-symmetric class membership criterion based on the maximum averaged contrast method was also proposed. Finally, this work also deals with the problem of root detection, defining a root measure. Although the results presented in [100] show that the RAG hierarchy correctly reflects the structure of

the image, the stochastic concept inherent to the method causes changes in the segmentation results when the algorithm is successively applied to the same input image.

The main drawback of the stochastic decimation process is that different outcomes of the random variable produce different structures and segmentations. Thus, the segmentation of an image varies between executions with the same input parameters. Besides, the decimation process should be controlled in order to assure that there exists at least a root for each interest region of the original image. Jolion and Montanvert [72] propose to modify the decimation process in order to bias toward vertices with high information value. Instead of a random variable, the adaptive pyramid uses an interest variable in the decimation process: the grey level variance gv_i from the receptive field of a pyramid vertex v_i. After survivor extraction, non-surviving vertices are linked to the most similar surviving vertex of its neighbourhood. In the adaptive pyramid the use of the interest variable avoids the definition of classes before the decimation process.

Jolion and Montanvert [72] introduce a root extraction process into the algorithm. A vertex is the root of an original image region if it satisfies the following conditions: i) a root vertex must be very different to the surviving vertices of its neighbourhood; and ii) the size of a region defined by a root must be large enough to avoid local variations due to noise; a small region must compensate its low size with a high contrast with all its neighbours.

Using the classical RAG pyramid, Bertolino and Montanvert [10] propose to generate distinct segmentations of an image at different resolutions by using the tree structure represented by the graph hierarchy. Thus, a region of any level can be recursively split into subregions at the level below. Fig. B.2 shows a graph hierarchy and its corresponding tree structure. Starting at the highest structure level, a homogeneity criterion is evaluated for each region: the standard deviation of every region is compared with a threshold σ_M to decide if the region must be split or not. Depending on σ_M the segmentation preserves more or less detail. Each region of the original image is extracted in the level where its representation is optimum. Since the standard deviation could be only suitable for certain kinds of images, other scale parameters may be used [10].

B.2.2 Segmentation with the localized pyramid

The graph pyramid is usually initialized with as many vertices as the number of pixels in the original input image. In the localized pyramid [63], only a subset of the image pixels are segmented (undefined zones), while the rest of image pixels is associated to one or several

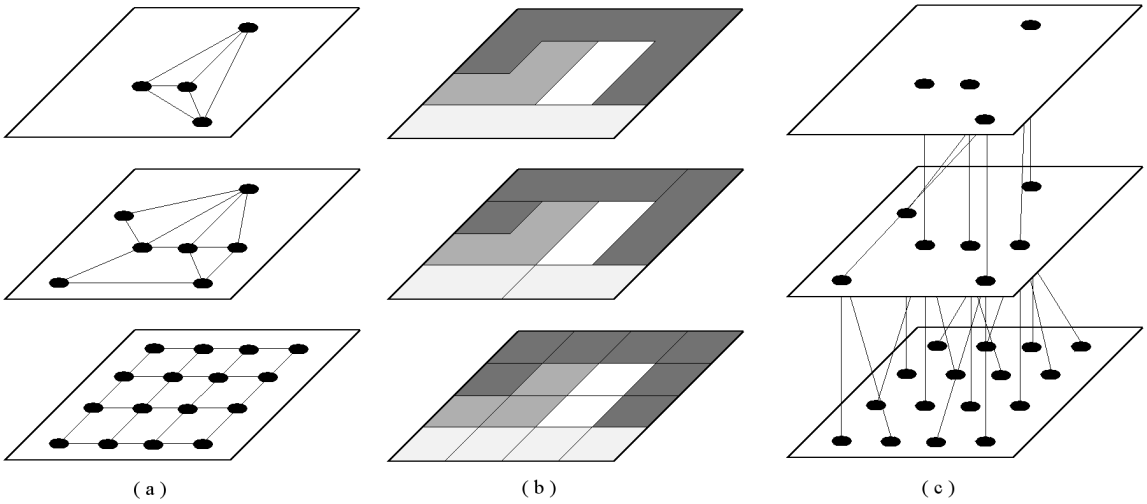

Figure B.2: Top-down segmentation based on the RAG hierarchy: a) region adjacency graphs; b) receptive fields pyramid; and c) corresponding tree structure.

vertices called roots. To initialize the local pyramid and to determine which pixels of the original image are going to be segmented and which not, a local homogeneity analysis can be performed. Thus, Huart and Bertolino [63] propose to compute a homogeneity image (H-image) from the CIE L*a*b* colour space. This H-image is a grey-scale image whose pixel values are the H-values representing the image discontinuities according to a homogeneous feature. Low values correspond to homogeneous regions (roots) and high values correspond to possible discontinuities (undefined zones). The pixels of the undefined zones are segmented using a simple graph data structure and a modified stochastic decimation process. This pyramid allows that, during the segmentation process, the pixels of the undefined zones merge together and/or with a neighboring root according to a similarity criterion.

When the segmentation has been locally performed, segmented regions are grouped using Gestalt criteria (perception-based image segmentation). In this region grouping process the local pyramid is extended with additional levels. The final result is a stack of partitions with very few objects [63].

B.2.3 Consensus image segmentation

Cho and Meer [32] propose a new approach for unsupervised segmentation based on RAG. This approach is derived from the consensus of a set of different segmentation outputs on one

input image. The probabilistic component of the RAG pyramid based segmentation implies that each time the algorithm is run the obtained result is slightly different. Differences are more important in the neighbourhoods where the piecewise constancy is less valid. In order to extract this information, local homogeneity is determined by collecting statistics for every pair of adjacency pixels, instead of statistics characterizing the spatial structure of the local neighbourhood of a pixel. The proposed segmentation algorithm works as follows:

- Given the input image, N different segmentations are obtained by exploiting the probabilistic component of the hierarchical RAG pyramid based technique [100]. An example of the variation in the structure of hierarchy is illustrated in Fig. B.3. Figs. B.3a and B.3d show that different surviving vertices were obtained because of the different random number assignation.

- The N segmented images are registered on the 8-connected mesh of the input image. Therefore, every pixel has N values associated. For every adjacent pixel pair a co-occurrence probability, i.e. the probability of belonging to the same delineated region, is derived. The set of all co-occurrence probabilities defines the co-occurrence field of the input image studied under the homogeneity criterion which defines the class distribution.

- Since the co-occurrence probabilities are derived from the initial image segmentations, they capture global information about the image at the local (pixel pair) level. The final segmentation of the input image is obtained by processing the co-occurrence probability field with a weighted RAG pyramid technique. This new graph is needed because each edge of the 8-connected mesh of the co-occurrence probability field has now a co-occurrence probability associated to it. Then, pixel pairs with high co-occurrence probability are grouped together based on the consensus about local homogeneity.

B.2.4 Image segmentation by connectivity preserving relinking

Segmentation by relinking [24] is performed by iteratively updating the class membership of pyramid vertices, i.e. by adapting parent-son edges. This technique, originally proposed for regular pyramids (see subsection B.1.1), presented serious drawbacks, the main of which is that classes represented by a vertex need not correspond to connected regions. Nacken [102] modifies the original relinking procedure and applies it to a RAG pyramid. The decimation algorithm proposed by Nacken [102] is briefly described in subsection 2.3.3.2. In this section, its application to segmentation purposes is shown.

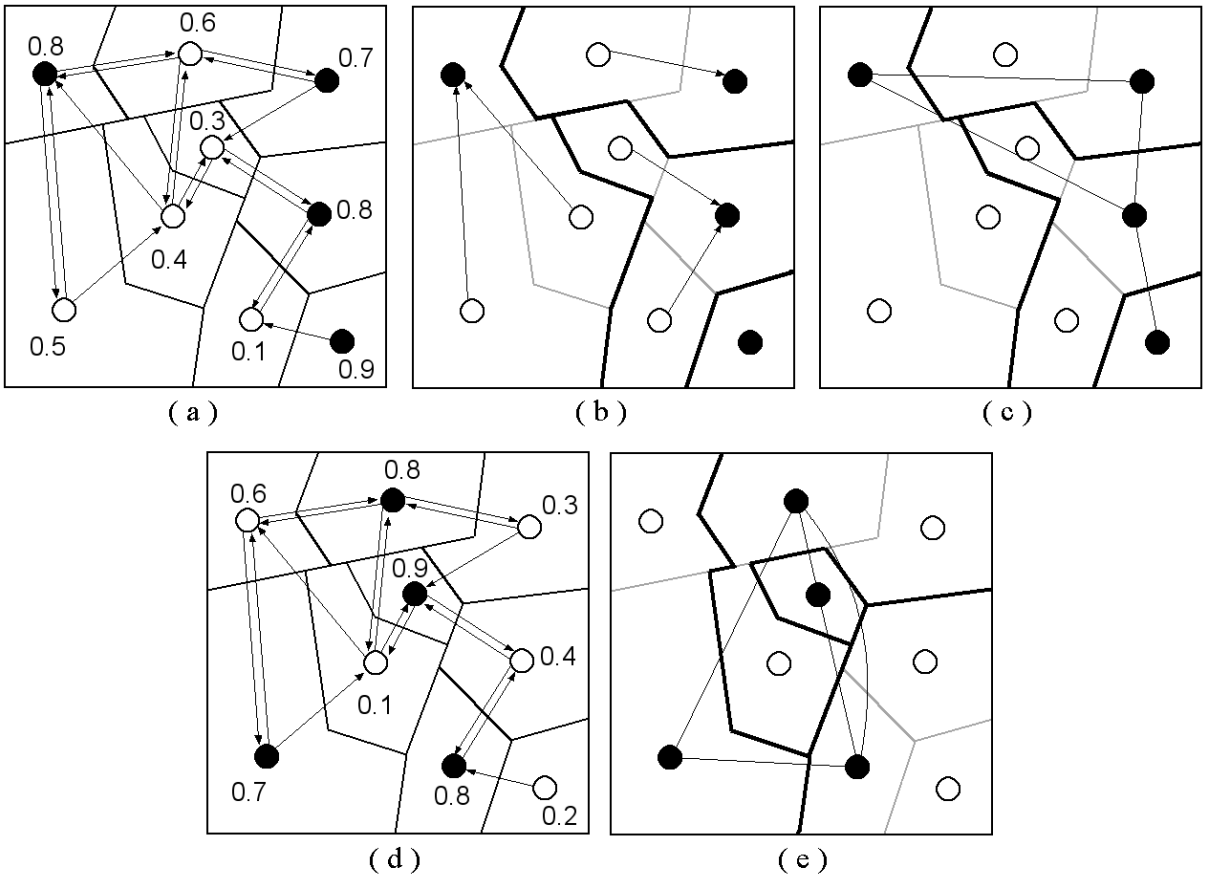

Figure B.3: Influence of the random component on the structure of the RAG pyramid: a) the RAG at level l. Arrows show the decomposition of RAG into classes using a non-symmetric class membership. Surviving vertices into each class are marked in black; b) non-surviving vertices allocation; c) the RAG at level l+1 from a-b); d) the RAG at level l with different random numbers assigned to the vertices; and e) the RAG at level l+1 from d).

In this pyramid, to create the vertices of level $l+1$, the vertices of level l are partitioned in a number of connected regions, as explained in section 2.3.3.2. In order to apply this scheme to segmentation purposes, [102] used as γ_i-value of a vertex v_i the area of the receptive field of this vertex. He defined two dissimilarity measures S_1 and S_2 between nodes:

$$S_1(v,w) = |g(v) - g(w)| - \frac{1}{2}(\sigma(v) + \sigma(w)) \tag{B.1}$$

$$S_2(v,w) = \frac{|g(v) - g(w)|}{1 + \frac{1}{2}(\sigma(v) + \sigma(w))} \tag{B.2}$$

being $g(v)$ the average grey value within the receptive field of a vertex v and $\sigma(v)$ the standard deviation of the grey value.

Once the vertices of G_{l+1} have been selected from G_l, and the son-parent edges between each non-surviving vertex and the survivor in its vicinity have been established, the connectivity-preserving relinking procedure is performed. For each vertex of level l in turn, a new parent is chosen from a set of candidate parents which preserve structure connectivity. The selected new parent could be the parent which minimizes the grey level difference. Another way to select the most suitable parent is to minimize the following energy function in each level:

$$E_{region}[l] = \sum_{v \in V_l} n(v)[g(v) - g(\pi(v))]^2 \tag{B.3}$$

V_l being the set of vertices at level l, $n(v)$ the receptive field of v, $g(v)$ its grey level value and $\pi(v)$ its parent.

Finally, Nacken [102] proposes to combine region and boundary information in the segmentation process (edges of the RAG correspond to the boundaries between receptive fields in the input image). The proposed boundary based relinking criterion is based on the minimization of the energy

$$E_{boundary}[l] = \sum_{v \in V_{l-1}} \eta(R(v)) \tag{B.4}$$

where $\eta(R(v))$ is the average response of an edge detection filter along the receptive field of a vertex v. New edge strength measures are defined based on boundary information. The combination of boundary and region information is then performed in a combined edge strength measure, which takes into account the previously defined measures.

B.2.5 Region growing stopping based on spatial autocorrelation

The application of the decimation process to a graph hierarchy to obtain segmentation requires defining a criterion to stop this reduction procedure when the best segmentation is obtained. In [86], a statistical test to control the region growing process is proposed. This test is applied to the adaptive pyramid [72]. Let $F_{l,l+1}$ be the decimation graph defined by

$$F_{l,l+1} = (V_l, E_{l,l+1}) \subset G_l \tag{B.5}$$

where $E_{l,l+1}$ are the inter-level edges between the levels l and $l+1$.

Thus, an edge in the decimation graph stands for the merging of two regions. If it is assumed that G_l is the best segmentation, then any edge in $F_{l,l+1}$ must be inappropriate. Therefore, it can be seen as an edge of a random graph that does not correctly correlate the associated vertices [86]. On the contrary, if G_l does not define the best segmentation, it must have an edge with a significant correlation between the associated vertices. Then, spatial autocorrelation can be used to control the decimation procedure, e.g. the region growing. Therefore, Lallich *et al.* [86] uses one of the most popular indicators to measure global spatial autocorrelation: the Moran's test [101].

B.2.6 Hierarchy of partitions by internal and external contrast measures

The aim of the hierarchy of partitions defined by Haxhimusa and Kropatsch [58] is to build a minimum weight spanning tree (MST) of the input image [43]. This MST will allow to find the region borders in a bottom-up way and, thus, to perform the image segmentation. Although the used data structure is the dual graph and the employed decimation process is the MIES proposed by Haxhimusa et *al.* [57], in Haxhimusa and Kropatsch [58] the construction of the dual graph is formulated as the building of a MST of the input image (level 0 of the graph hierarchy). Thus, an algorithm based on Boruvka's proposal [58] is used to build in a hierarchical way a MST preserving the image topology. The method is based on a previous work of Felzenszwalb and Huttenlocher [43].

This MST is built as follows:

In a hierarchy of graphs, where G_l defines the graph on level l of the hierarchy, every vertex u_i of G_l has a receptive field in the base level $CC(u_i)$. In each level l the union of the receptive fields of the vertices in the level defines a partition $P_l = \{CC(u_i)\}_{i=1...n}$. Then, to build the level $l+1$ from the level l, the goal is to find a partition P_{l+1} by merging members of P_l. Haxhimusa and Kropatsch [58] define the following pairwise merge criterion:

$$Comp(CC(u_i), CC(u_j)) \begin{cases} 1 \; if \; Ext(CC(u_i), CC(u_j)) \leq \\ \qquad\qquad PInt(CC(u_i), CC(u_j)), \\ 0 \; otherwise \end{cases} \tag{B.6}$$

where $PInt(\cdot,\cdot)$ and $Ext(\cdot,\cdot)$ are the minimum internal contrast difference and the external contrast between two receptive fields, respectively [43]. $Ext(CC(u_i), CC(u_j))$ is the smallest dissimilarity between $CC(u_i)$ in P_l and $CC(u_j)$ in P_l. $PInt(\cdot,\cdot)$ is defined as

$$\begin{aligned} PInt(CC(u_i), CC(u_j)) = min(&Int(CC(u_i)) + \tau(CC(u_i)), \\ &Int(CC(u_j)) + \tau(CC(u_j))) \end{aligned} \tag{B.7}$$

$Int(CC(u_i))$ being the internal contrast of the $CC(u_i)$ in P_l. This contrast measure is defined as the largest dissimilarity of component $CC(u_i)$. The threshold function τ controls the degree to which the external variation can actually be larger than the internal variations and still have the receptive fields be considered similar [43].

This hierarchical partitioning algorithm has been applied to the combinatorial pyramid framework by Ion et *al.* [66]. Results show that the algorithm can handle large variations and gradient intensity in images.

B.2.7 Segmentation based on combinatorial pyramids and union-find algorithm

Brun and Kropatsch [19] proposed a segmentation application based on the combinatorial pyramid and the union-find based decimation algorithm. The segmentation algorithm works on grey level images and can be briefly summarized as follows:

1. The original image is quantized into K gray levels. Each vertex in level 0 of the graph hierarchy encodes a connected component of the pixels whose gray level values are mapped onto a same interval. The background of the image is determined by selecting the largest region adjacent to the exterior of the image.

2. All regions included in the background whose size is lower than a given threshold T are merged with the background (level 1).

3. In order to perform the union-find process and to build a level $l+1$ from l the mean gray level of each vertex of l is used to initialize a gray-level histogram. The frequency $h(i)$ of one entry i of the histogram is set to the number of vertices whose mean gray level is equal to i. This histogram is then quantized into K values. The algorithm merges any couple of adjacent vertices whose mean gray values are mapped into the same interval. The vertices which are merged together generate a new vertex of the level $l+1$.

The last step is iterated until no merge occurs. It must be noted that the quantization process only provides a partition of the range of grey values. The encoding of the partition and the merge operations are performed using the combinatorial pyramid model.

MENÚ SALIR

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

ADDENDUM A LA TESIS DOCTORAL

TRACKING OBJECTS WITH THE BOUNDED IRREGULAR PYRAMID

AUTOR: Rebeca Marfil Robles
Ingeniera de Telecomunicación

2006

Resumen

En cualquier proceso de seguimiento de objetos usando visión artificial, tanto la forma de representar y modelar el objeto a seguir u objetivo (*target*), como el proceso de localización de dicho objeto en cada fotograma de la secuencia, son procesos centrales. En la literatura, ambos procesos suelen agruparse en uno solo, denominado Representación y Localización del objetivo.

El propósito de esta Tesis es realizar, en tiempo real, el seguimiento de objetos no rígidos que pueden sufrir cambios importantes de apariencia. Además, dicho seguimiento se llevará a cabo sin utilizar ningún modelo que requiera un entrenamiento previo del sistema. Para alcanzar esta meta, en esta Tesis se propone un nuevo modelo de representación del objetivo, que almacena su apariencia en una máscara (*template*) jerárquica. Esta representación permite que el proceso de localización del objeto se lleve a cabo mediante correspondencia jerárquica, lo que reducirá el coste computacional asociado a dicho proceso. Además de esta característica, fundamental para poder realizar en tiempo real el proceso de seguimiento, el modelo deberá ser robusto a situaciones tales como oclusiones, cambios de iluminación, deformaciones del objeto, cambios en su orientación o en el punto de vista y presencia de otros objetos en movimiento en la escena.

Como se ha comentando, para conseguir reducir el coste computacional, en esta Tesis se propone una nueva estructura para representar la máscara y llevar a cabo el proceso de correspondencia o localización del objeto de forma jerárquica. Esta nueva estructura jerárquica se ha denominado Pirámide Irregular Acotada (*Bounded Irregular Pyramid (BIP)*), y será la base de un sistema que permite seguir la evolución de objetos no rígidos en tiempo real. En resumen, las principales contribuciones de esta Tesis son las siguientes:

- La implementación y evaluación detallada de las características de la estructura piramidal propuesta para el procesado de imagen: la Pirámide Irregular Acotada.
- El desarrollo de una nueva representación del objetivo a seguir utilizando la Pirámide Irregular Acotada. Esta representación o modelo consiste en una máscara jerárquica que

almacena la apariencia del objeto seguido a lo largo del tiempo.

- La implementación de un algoritmo de seguimiento de objetos basado en correspondencia (*template matching*) que utiliza dicha representación.
- La validación experimental del sistema propuesto en las situaciones, previamente comentadas, de oclusiones, cambios de iluminación, deformaciones del objeto, cambios en su orientación o en el punto de vista y presencia de otros objetos en movimiento en la escena.
- El estudio del comportamiento del sistema de seguimiento propuesto cuando es utilizado como parte de sistemas más complejos, que operan en tiempo real, tales como un sistema de captura de movimiento humano y un mecanismo atencional.

Esta Tesis está dividida en 6 capítulos principales. El capítulo 1 es una introducción donde se explican las motivaciones que originaron el desarrollo de esta Tesis, sus objetivos y cuáles han sido sus principales contribuciones. El segundo de ellos hace una revisión de los métodos de seguimiento más relevantes presentes en la literatura, así como de las principales estructuras piramidales. En los dos capítulos siguientes se explican en detalle la Pirámide Irregular Acotada y el sistema de seguimiento propuesto, respectivamente. El capítulo 5 estudia el uso del algoritmo de seguimiento propuesto en las dos aplicaciones en tiempo real comentadas anteriormente: el sistema de captura de movimiento humano y el mecanismo atencional. En el capítulo final se presentan las conclusiones extraídas del desarrollo de esta Tesis y se dan algunas ideas de cuál podría ser el trabajo futuro a realizar. Esta Tesis incluye también dos apéndices los cuales explican, respectivamente, el espacio de color HSV empleado para construir la BIP, y los principales métodos de segmentación piramidal presentes en la literatura.

Este documento constituye un resumen en español de los capítulos centrales de esta Tesis (capítulos 3, 4 y 5).

La Pirámide Irregular Acotada (BIP)

Las estructuras piramidales presentes en la literatura pueden ser divididas en dos categorías principales: regulares e irregulares. Tras estudiar estas estructuras durante el desarrollo de esta Tesis, se llegó a la conclusión que ninguna de ellas satisfacía las necesidades del sistema de seguimiento propuesto: las regulares debido a su incapacidad para representar determinados tipos de objetos, y las irregulares debido a su excesivo coste computacional. La necesidad de obtener una estructura piramidal que satisfaga los requisitos de bajo coste computacional

y buenos resultados hace que surja la Pirámide Irregular Acotada. Esta estructura combina las ventajas de las pirámides regulares e irregulares dentro de una misma estructura. Así, esta pirámide permite procesar imágenes diez veces mas rápido que el resto de estructuras irregulares y con resultados similares. Esta reducción del tiempo de proceso permite utilizar la BIP en aplicaciones en tiempo real tales como el algoritmo de seguimiento de objetos propuesto en esta Tesis.

La idea principal de la Pirámide Irregular Acotada es utilizar una estructura regular en las zonas homogéneas de la imagen de entrada y una estructura irregular en el resto de regiones. Específicamente, la estructura de datos de la BIP es una combinación de una estructura regular $2x2/4$ con un grafo simple. De esta forma en la parte regular de la BIP se emplea una diezmado regular y en su parte irregular se lleva a cabo un proceso de diezmado denominado *union-find*. La parte irregular de la BIP permite solucionar los tres problemas principales de las estructuras regulares: no conectividad de las regiones resultantes, no adaptabilidad a la estructura de la imagen de entrada y obtención de diferentes resultados para pequeños desplazamientos de la imagen (*shift-variance*). Por otro lado, la BIP es computacionalmente eficiente porque su parte regular evita que la estructura crezca demasiado. Por ello, su uso permite llevar a cabo el proceso de seguimiento en tiempo real, ya que esta estructura se construye y recorre muy rápidamente.

Para evaluar los resultados de la Pirámide Irregular Acotada y compararla con las estructuras piramidales más importantes presentes en la literatura, se ha aplicado la misma en un proceso de segmentación de imágenes en color. Se ha elegido evaluar los resultados de segmentación por dos motivos principales: i) el proceso de representación del objetivo a seguir propuesto en esta Tesis es un proceso de segmentación, en el cual no se segmenta la imagen completa, sino únicamente la porción de la misma donde el objeto es más probable que esté localizado; y ii) existen métodos muy conocidos para evaluar los resultados de segmentación. A continuación se presenta brevemente el proceso de segmentación utilizando la BIP.

Procedimiento de segmentación utilizando la Pirámide Irregular Acotada

Para la caracterización del color se ha utilizado el espacio de color HSV, empleando como criterio de similitud la distancia en color dentro de este espacio. El umbral de similitud es un umbral adaptativo por nivel, que dificulta el proceso de enlazado de vértices en los niveles superiores de la estructura, evitando así la fusión de regiones muy grandes en la base.

El proceso de construcción de un nivel $l+1$ a partir del nivel l es el siguiente:

1. Proceso de diezmado regular: Este proceso consiste en un promediado 4 a 1 de los vértices regulares del nivel l generando los vértices regulares del nivel $l+1$.

2. Búsqueda de padre y hermanamiento regular: cada vértice regular del nivel l que no tiene padre en el nivel $l+1$ busca entre sus vecinos un vértice con padre similar a él para enlazarse (*búsqueda de padre*). Si no lo encuentra, busca un vértice vecino similar a él para enlazarse y generar un nodo irregular en el nivel superior (*hermanamiento*).

3. Búsqueda de padre y hermanamiento irregular: el proceso anterior es realizado de nuevo pero ahora entre vértices irregulares.

4. Generación de enlaces en el nivel $l+1$: se establecen los enlaces entre los vértices del nivel $l+1$ teniendo en cuenta las vecindades de sus hijos en el nivel inferior.

Para realizar la segmentación, los vértices sin padre de cualquier nivel se eligen como raíces de las regiones de segmentación.

Estudio comparativo

En esta Tesis se han elegido 3 métodos empíricos para comparar los resultados obtenidos por los diferentes métodos de segmentación piramidal: la función F propuesta en [9], la función Q propuesta en [2] y la *Shift Variance* propuesta por Prewer y Kitchen [10].

Para realizar la comparación se han implementado dos estructuras piramidales regulares: la pirámide enlazada propuesta por Burt *et al.* [4] (LRP), y la pirámide enlazada ponderada con enlazado probabilístico [6] (WRP). También se han incluido comparaciones con 5 métodos irregulares: la jerarquía clásica de grafos utilizada por Bertolino y Montanvert [1] (ClIP), la pirámide localizada [7] (LIP), el algoritmo propuesto por Lallich *et al.* [8] (MIP), la jerarquía de particiones de la imagen [5] (HIP) y la pirámide combinatoria [3] (CoIP).

Las imágenes utilizadas en el proceso de comparación proceden de la base de datos *Waterloo and Coil 100*. Se ha utilizado un tamaño de imagen de 256x256 y un PC Pentium IV a 3GHz.

Los tiempos de proceso se muestran en la Tabla 1. Loa algoritmos más rápidos son la BIP y los algoritmos regulares. La BIP es más rápida que los enfoques irregulares debido a que una parte de la imagen es procesada siguiendo un proceso regular. Además, es más rápida que las estructuras regulares debido a que la BIP no sigue un proceso iterativo en su construcción,

	Tiempos de proceso (seg)			Altura de la pirámide			Número de regiones		
	t_{min}	t_{med}	t_{max}	h_{min}	h_{med}	h_{max}	NR_{min}	NR_{med}	NR_{max}
LRP	0.94	1.37	1.81	9	9	9	17	81.6	203
WRP	0.31	0.40	0.58	9	9	9	19	79.7	148
ClIP	2.51	3.96	7.68	17	36.7	72	9	84.1	210
LIP	1.71	2.78	6.13	8	25.4	51	12	73.8	210
MIP	2.43	3.47	4.47	13	33.3	62	45	107.7	201
BIP	0.14	0.17	0.39	8	8.8	15	8	83.5	229
HIP	4.07	4.29	4.91	10	11.6	18	23	76.2	149
CoIP	1.32	2.88	12.8	9	74.4	202	25	91.6	238

Cuadro 1: Tiempos de proceso, altura de la pirámide y número de regiones obtenidas en la segmentación. Los valores medios se han calculado utilizando 30 imágenes

como las pirámides regulares, sino que es construida en sólo una pasada. La BIP es la pirámide irregular con una altura menor debido a que su parte regular la previene de crecer demasiado.

La Tabla 2 presenta los resultados obtenidos en la comparación de las diferentes pirámides utilizando los métodos empíricos anteriormente comentados. Esta tabla muestra que todas las pirámides irregulares obtienen mejores resultados que las regulares. Se observa como los resultados obtenidos por la BIP son, aunque muy similares, ligeramente peores que los del resto de estructuras irregulares, debido a su parte regular. En resumen, se puede afirmar que la BIP obtiene resultados similares a los del resto de estructuras irregulares, reduciendo al menos 10 veces el tiempo de proceso.

Las Figs. 1 y 2 muestran 5 de las imágenes utilizadas en las comparaciones y los resultados obtenidos por los diferentes métodos.

	F			Q			SV		
	F_{min}	F_{med}	F_{max}	Q_{min}	Q_{med}	Q_{max}	SV_{min}	SV_{med}	SV_{max}
LRP	765.8	1070.4	1515.5	1052.1	1524.9	2105.4	37.8	66.9	83.5
WRP	791.2	1072.8	1428.2	1133.7	1480.6	2034.2	49.6	69.9	98.5
ClIP	329.3	840.2	1290.0	479.1	1062.7	1590.3	18.0	28.8	42.8
LIP	213.6	746.1	1345.6	489.4	1002.5	1327.4	20.8	31.7	46.7
MIP	290.4	646.6	1043.7	360.5	817.6	1292.5	19.3	30.1	42.4
BIP	198.6	711.7	1556.1	339.4	1086.7	1919.8	26.4	44.1	84.5
HIP	201.7	689.2	1201.6	458.3	957.8	1521.5	18.5	27.1	35.9
CoIP	234.3	618.8	934.9	415.5	878.5	1294.5	21.3	30.7	42.8

Cuadro 2: Valores de F, Q y Shift Variance. Los valores medios se han calculado utilizando 30 imágenes.

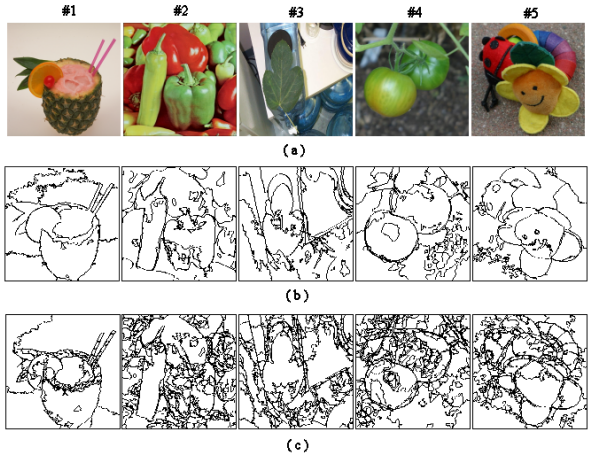

Figura 1: a) Imágenes originales; b) resultados obtenidos con la LIP; c) resultados obtenidos con la WIP.

Algoritmo de seguimiento

En esta sección se hace un resumen del algoritmo de seguimiento propuesto en esta Tesis, y de cómo la Pirámide Irregular Acotada es utilizada para representar de forma jerárquica una máscara del objeto a seguir. Esta máscara y la forma en la que es actualizada permite al algoritmo manejar cambios de apariencia del objeto a seguir y oclusiones tanto parciales como totales.

La Fig. 3 muestra los diferentes bloques del algoritmo propuesto. Cada uno de estos bloques se comentan a continuación.

Inicialización del algoritmo

El objeto a seguir se elige manualmente del primer fotograma de la secuencia. Para ello se segmenta la imagen de entrada utilizando la BIP, pudiendo seleccionarse el objeto a seguir como cualquiera de las regiones obtenidas. La parte regular de la estructura jerárquica de la región elegida es la primera máscara.

Sobre-segmentación

El primer paso del proceso de seguimiento es realizar una sobre-segmentación de la porción de la imagen de entrada donde es más probable que se encuentre el objeto (*Region de Interés (ROI)*) utilizando la BIP. Un proceso de sobre-segmentación es aquel que divide la zona de la imagen en un número de regiones mucho mayor del existente realmente. Cada región

será una estructura jerárquica formada por un conjunto de vértices de la BIP obtenida en la sobre-segmentación.

Correspondencia jerárquica

Una vez se ha realizado la sobre-segmentación de la ROI, se procede a buscar el objeto. Para ello se utilizan únicamente las partes regulares de las BIPs correspondientes a la máscara y la ROI. El proceso para localizar el objetivo en un fotograma determinado tiene dos pasos principales:

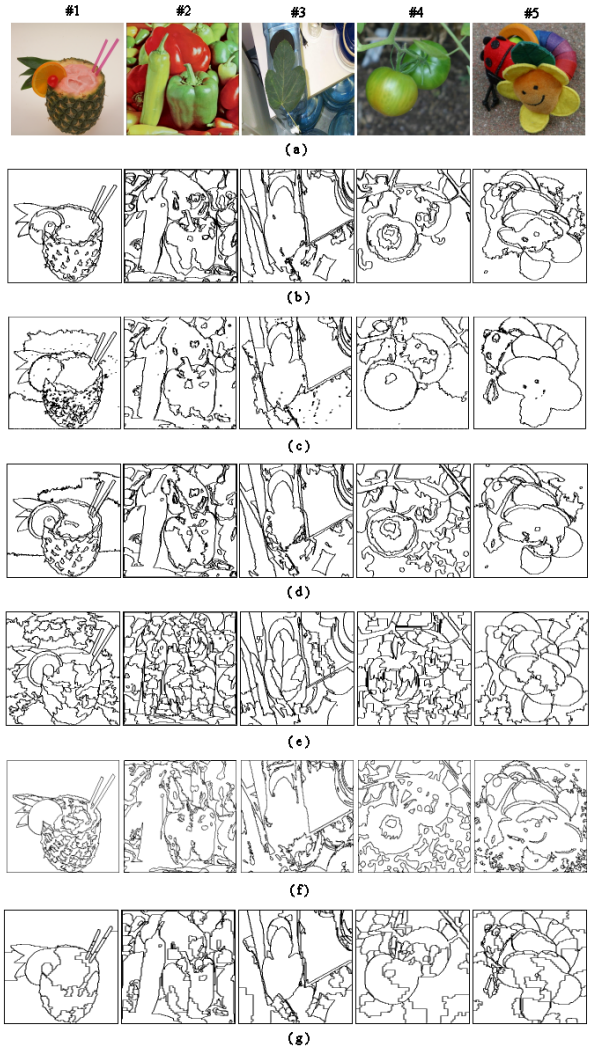

Figura 2: Resultados de segmentación; a) Imágenes originales; b) ClIP c) MIP; d) localized pyramid; e) HIP; f) CoIP; g) BIP.

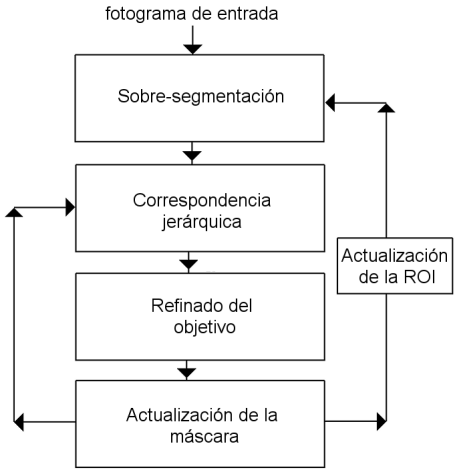

Figura 3: Diagrama de bloques del algoritmo de seguimiento.

1. Elección del nivel de trabajo: la correspondencia jerárquica empieza en un nivel denominado nivel de trabajo. Este nivel es el nivel más alto de la estructura jerárquica de la ROI dónde el área del objeto es mayor que un porcentaje T_A del área total del objeto.

2. Localización del objetivo: la búsqueda comienza en el nivel de trabajo, para ello el nivel correspondiente de la máscara se pone y desplaza sobre el mismo nivel de la ROI. En cada posición se calcula el número de vértices que se corresponden entre ambos porque son similares en color. La posición en la que se encuentra el objetivo será aquella con una mayor correspondencia y que supera un cierto umbral. Si el objetivo no se encuentra en el nivel de trabajo se busca en el nivel inferior y así sucesivamente. Una vez encontrado el objetivo los vértices de la ROI que se corresponden con él son marcados como vértices del objetivo.

Refinado del objetivo

Una vez encontrado, la apariencia del objetivo debe ser refinada. Para ello, lo primero es incluir en el objetivo aquellos vértices de la estructura jerárquica de la ROI que pertenecen a una región de la sobre-segmentación que incluye algún vértice del objetivo. El segundo paso es estudiar las vecindades de cada vértice del objetivo en la ROI. Si en estas vecindades existen vértices similares al objetivo, éstos, y todos los miembros de su región de sobre-segmentación, son incluidos en la representación del objetivo.

Actualización de la máscara

La máscara utilizada almacena información del objetivo en el fotograma actual y también en fotogramas anteriores, para ello utiliza un peso que otorga mas importancia a la información reciente. La información más antigua es olvidada de forma lineal. Esta máscara permite manejar cambios de apariencia del objetivo y oclusiones parciales.

Seguimiento de varios objetos

El algoritmo propuesto permite seguir varios objetos de forma simultánea sin aumentar de forma proporcional el tiempo de proceso. Para ello, en lugar de tener una BIP para cada ROI lo que se hace es representar todas las ROIs dentro de la misma estructura jerárquica. Además el proceso de correspondencia se ejecuta nivel a nivel para todos los objetivos, es decir no se recorre la estructura para cada objetivo, sino que se recorre una sola vez.

Resultados

Para validar experimentalmente el método de seguimiento propuesto, éste ha sido probado en diferentes situaciones. La Fig. 4 muestra alguna de estas situaciones y como el algoritmo ha sido capaz de manejarlas. Concretamente esta figura muestra fotogramas de tres secuencias de video diferentes. En la primera de ellas (secuencia #1) se aprecia como la cámara está en movimiento. En la secuencia #2 se observa un gran cambio en la apariencia del objeto seguido. Finalmente, en la secuencia #3 se muestra el seguimiento de varios objetos de forma simultánea con oclusiones parciales y totales. Además en la secuencia #1 se aprecia un cambio en la iluminación de la escena.

La Tabla 3 muestra los tiempos de ejecución del algoritmo propuesto en las tres secuencias de la Fig. 4. El tamaño de imagen utilizado ha sido de 128x128 píxeles en un PC Pentium IV a 3GHz. Los tiempos demuestran que el algoritmo funciona en tiempo real con una velocidad de proceso, en el caso de seguimiento de un solo objeto, de 27 fotogramas por segundo. En el caso de seguimiento de 3 objetos, esta velocidad baja únicamente a 20 fotogramas por segundo. Vemos como la mayor parte del tiempo de proceso se consume en la sobre-segmentación. De ahí la importancia de tener un algoritmo rápido de segmentación jerárquica, como es el caso de la BIP. Este tiempo depende del tamaño de las regiones de interés. También se aprecia como los tiempos de sobre-segmentación y correspondencia no aumentan proporcionalmente con el número de objetos.

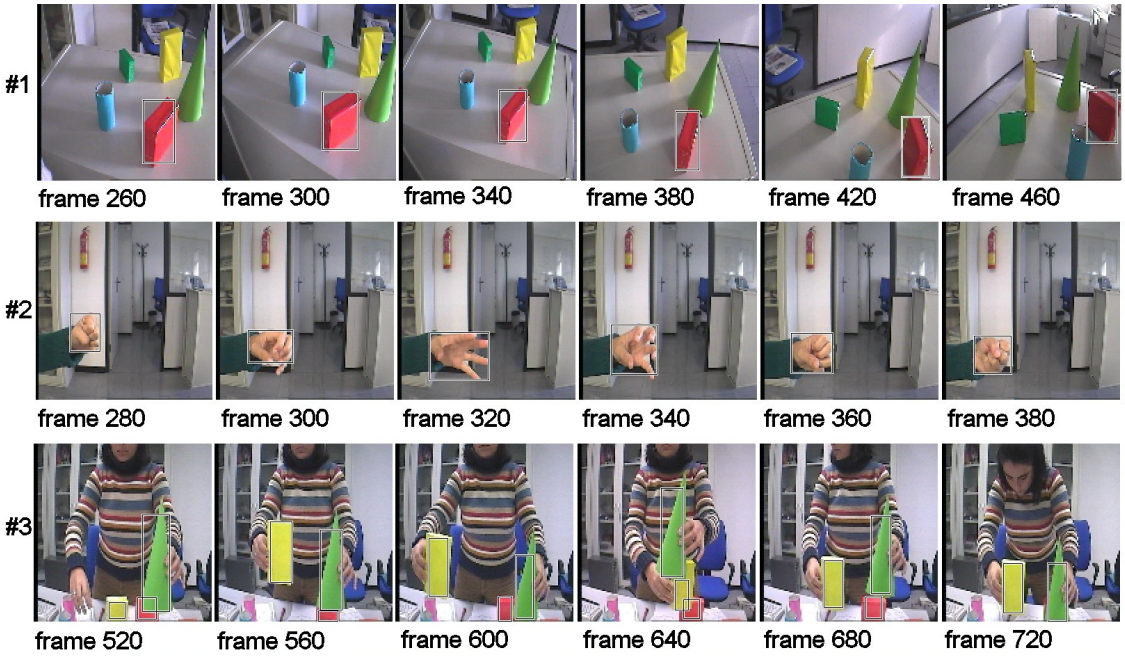

Figura 4: Resultados del algoritmo de seguimiento propuesto; #1 cámara en movimiento; #2 cambios de apariencia; #3 seguimiento de varios objetos.

	Tiempos de ejecución por fotograma milisegundos			
Secuencias	Inicialización	Sobre-segmentation	Correspondencia	Total [a]
Mano	6.5	12.4	5.8	38.1
Cámara en movimiento	7.5	11.1	5.9	37.1
Varios objetos	7.6	20.1	11	51.6

[a]incluyendo la captura de imágenes

Cuadro 3: Tiempos de ejecución

Aplicaciones

En esta sección se resumen las dos aplicaciones de tiempo real en las que ha sido probado el correcto funcionamiento del algoritmo de seguimiento propuesto en esta Tesis: el mecanismo atencional y el sistema de captura de movimiento humano.

Mecanismo atencional

El mecanismo atencional presentado en esta Tesis es un mecanismo de propósito general basado en la teoría de integración de características. Consta de dos etapas principales: una etapa preatentiva y una etapa semiatentiva.

En la etapa preatentiva, se calculan en paralelo un conjunto de características de bajo

nivel, que son posteriormente combinadas en un único mapa denominado *mapa de importancia*. Este mapa muestra la importancia de cada región de la imagen con respecto a las demás regiones.

La etapa semiatentiva tiene dos módulos principales: un módulo que extrae las regiones más importantes del mapa de importancia, y un módulo que realiza el seguimiento de estas regiones. Concretamente el algoritmo propuesto en esta Tesis realiza el seguimiento, en el fotograma actual, de las regiones extraídas del mapa de importancia en los fotogramas anteriores. Las posiciones de estas regiones ayudan al módulo de extracción de regiones a no extraer en el fotograma actual regiones que ya han sido previamente extraídas en fotogramas anteriores. Este proceso es denominado *inhibición de retorno* y es de vital importancia en los mecanismos atencionales que pretenden manejar entornos dinámicos con objetos en movimiento.

Etapa preattentiva

El mecanismo atencional propuesto calcula una serie de características de bajo nivel de la imagen de entrada para determinar cuáles son las regiones más importantes de la misma. Cada una de estas características es representada por medio de una imagen o mapa en niveles de gris, en el cual, regiones importantes con respecto tienen un valor alto de gris y viceversa. Estos mapas de características se combinan en un único mapa de importancia que muestra la importancia de cada región de la imagen.

Las características calculadas han sido: contraste de color, contraste de intensidad, disparidad y color piel. La última de éstas características permite localizar en la imagen posiciones donde una persona puede estar situada. El mapa final de importancia se calcula realizando la suma normalizada de los mapas de características.

Etapa semiattentiva

Para extraer las regiones mas importantes de la imagen de entrada, el mapa de importancia es segmentado en regiones con un valor homogéneo de importancia. Sólo se tienen en cuenta aquellas regiones con un valor de importancia elevado y con un valor de área superior a un cierto umbral. La representación jerárquica de estas regiones son los primeros templates que utilizará el algoritmo de seguimiento.

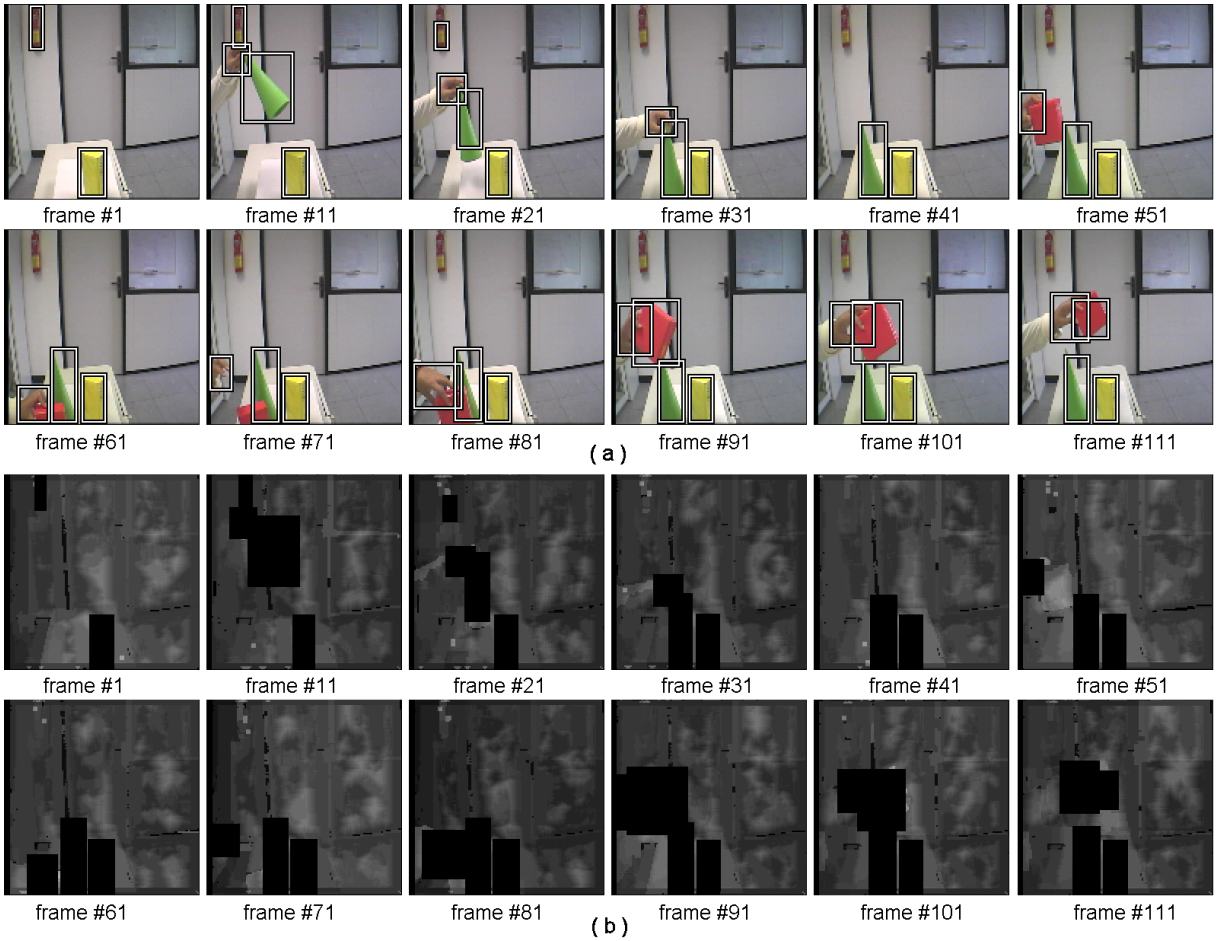

Figura 5: Ejemplo de regiones extraídas: a) imágenes de entrada; y b) mapa de importancia asociado con a).

Resultados

La Fig. 5 muestra una secuencia que ilustra el funcionamiento del mecanismo atencional propuesto. Los mapas de importancia muestran las regiones inhibidas. La utilización del algoritmo de seguimiento propuesto en esta Tesis permite implementar eficientemente la inhibición de retorno sin retardar el resto del proceso, haciendo que el sistema completo pueda ejecutarse a una velocidad de 5 fotogramas por segundo.

Sistema de captura de movimiento humano

Este sistema captura el movimiento de la parte superior del cuerpo de una persona sin utilizar dispositivos especiales o marcadores. Para ello, el sistema consta de dos módulos

principales: un módulo de visión, encargado de extraer la posición 3D de los centroides de las manos y la cabeza de la persona; y un módulo de extracción de ángulos que, mediante un modelo cinemático de la persona y un algoritmo de cinemática inversa, calcula los ángulos de las articulaciones de los brazos de la persona.

Esta Tesis se ha centrado en el módulo de visión, estando el módulo de extracción de ángulos detalladamente explicado en la bibliografía proporcionada en esta Tesis.

Módulo de visión

El módulo de visión es el encargado de realizar el seguimiento de las manos y la cabeza de la persona en cada fotograma de la secuencia de entrada, calculando las coordenadas (X, Y, Z) de los mismos.

El seguimiento 2D de los objetivos anteriormente mencionados se realiza utilizando el algoritmo de seguimiento propuesto en esta Tesis. Este algoritmo de seguimiento ha sido, además, modificado ligeramente para permitir la extracción en cada fotograma de la disparidad de los centroides de los objetivos. Para ello la disparidad se ha calculado realizando el seguimiento de los objetivos entre la imagen izquierda de un fotograma y su correspondiente imagen derecha. Las modificaciones realizadas en el algoritmo de seguimiento para este cálculo de disparidad han sido las siguientes:

- Las máscaras no almacenan información de máscaras previas, sólo almacenan información de los objetivos en el fotograma actual.

- En el el proceso de correspondencia, las máscaras se desplazan únicamente a lo largo del eje horizontal debido a la disposición paralela de las cámaras.

La disparidad se calcula como el desplazamiento que provoca una correspondencia máxima entre la máscara y la imagen derecha. Una precisión a nivel sub-pixel es conseguida realizando una aproximación polinomial de segundo orden alrededor de los valores de correspondencia de la disparidad previamente estimada.

Las coordenadas 3D de los centroides de los objetivos se calculan utilizando las posiciones 2D, la disparidad y los parámetros de calibración de las cámaras.

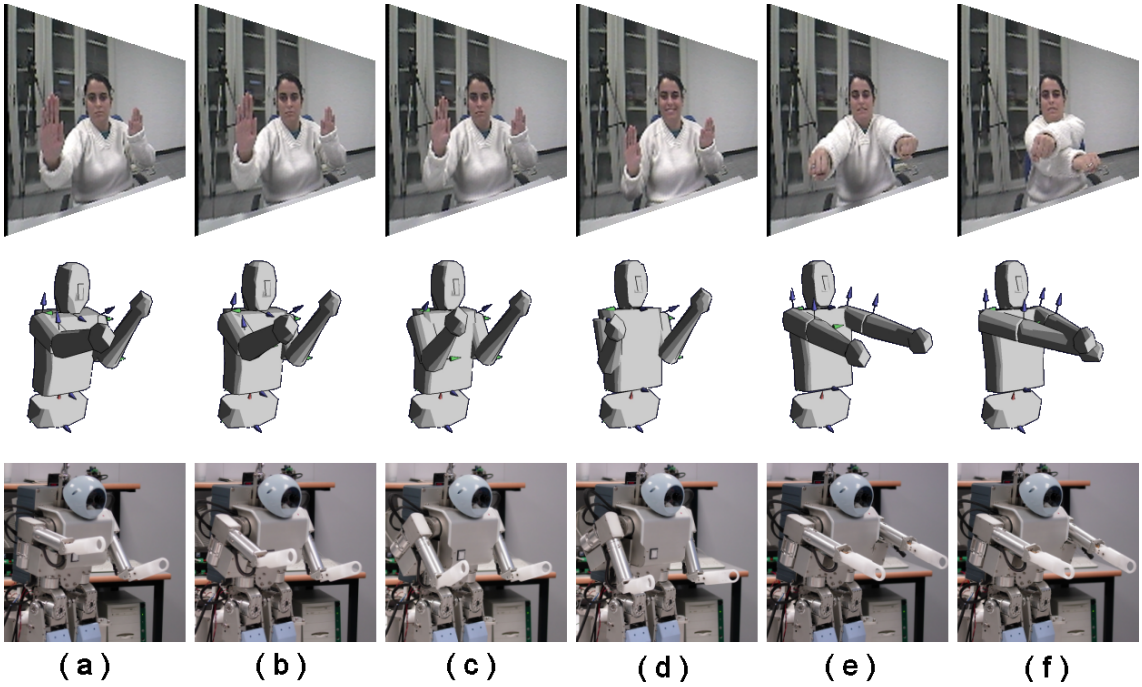

Figura 6: Resultados de estimación del movimiento observado. Fila superior: imágenes capturadas con la cámara izquierda. Fila central: posición estimada del modelo. Fila inferior: posición correspondiente adoptada por el robot.

Resultados

La Fig. 6 muestra un ejemplo de los resultados obtenidos. La fila superior muestra los fotogramas correspondientes a los movimientos de la persona capturados por la cámara izquierda. La fila central representa los movimientos realizados por el modelo cinemático tras la extracción de los ángulos de las articulaciones. La fila inferior representa la ejecución de estos movimientos en una plataforma robótica HOAP-I.

El sistema completo funciona a una velocidad de 25 fotogramas por segundo.

Bibliografía

[1] P. Bertolino and A. Montanvert. Multiresolution segmentation using the irregular pyramid. In *International Conference on Image Processing*, pages 257–260, Lausanne, Switzerland, 1996.

[2] M. Borsotti, P. Campadelli, and R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19(8):741–747, June 1998.

[3] L. Brun and W.G. Kropatsch. Receptive fields within the combinatorial pyramid framework. *Graphical Models*, 65(1-3):23–42, May 2003.

[4] P.J. Burt, T.H. Hong, and A. Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Trans. Systems, Man and Cybernetics*, 11(12):802–809, December 1981.

[5] Y. Haxhimusa and W.G. Kropatsch. Segmentation graph hierarchies. *Lecture Notes in Computer Science*, 3138:343–351, 2004.

[6] T.H. Hong and A. Rosenfeld. Compact region extraction using weighted pixel linking in a pyramid. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(2):222–229, March 1984.

[7] J. Huart and P. Bertolino. Similarity-based and perception-based image segmentation. In *International Conference on Image Processing*, pages III: 1148–1151, Genova, Italy, 2005.

[8] S. Lallich, F. Muhlenbach, and J.M. Jolion. A test to control a region growing process within a hierarchical graph. *Pattern Recognition*, 36(10):2201–2211, October 2003.

[9] J.Q. Liu and Y.H. Yang. Multiresolution color image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(7):689–700, July 1994.

[10] D. Prewer and L. Kitchen. Soft image segmentation by weighted linked pyramid. *Pattern Recognition Letters*, 22(2):123–132, February 2001.

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

CONCLUSIONES DE LA TESIS DOCTORAL

TRACKING OBJECTS WITH THE BOUNDED IRREGULAR PYRAMID

AUTOR: Rebeca Marfil Robles
Ingeniera de Telecomunicación

2006

Conclusiones

En esta Tesis se ha presentado un nuevo enfoque para el proceso de Representación y localización del objetivo en un sistema de seguimiento de objetos. Este nuevo enfoque utiliza un modelo de apariencia del objetivo basado en una máscara jerárquica. Se ha elegido este tipo de modelo de apariencia debido a su capacidad para seguir objetos no rígidos sin una etapa previa de aprendizaje de diferentes vistas del objeto. Para conseguirlo, el método propuesto emplea una máscara pesada, la cual es actualizada de forma dinámica para seguir los cambios de apariencia y punto de vista del objeto seguido. Los pesos se utilizan para establecer un compromiso entre la máscara actual y las máscaras de fotogramas anteriores. De esta forma, los pesos dan más importancia a los datos más recientes, mientras que los datos más antiguos son linealmente "olvidados". Esta máscara pesada, y la forma en la que es actualizada, permite al algoritmo manejar de forma satisfactoria situaciones tales como:

- Oclusiones parciales y totales del objeto seguido: la duración de las oclusiones totales que el algoritmo puede manejar se controla mediante un parámetro definido por el usuario (α), que determina el grado de "olvido"de los datos antiguos.
- Cambios de iluminación: la máscara puede adaptarse a cambios graduales de iluminación que produzcan una modificación en el color del objeto seguido menor que el umbral utilizado en el proceso de correspondencia.
- Cambios de apariencia del objeto debidas a deformaciones, zooms, rotaciones o cambios en el punto de vista.
- Presencia de otros objetos en movimiento en la escena.
- Seguimiento de varios objetos al mismo tiempo: el método de seguimiento propuesto permite seguir la apariencia y los cambios de posición de varios objetos simultáneamente. Pueden aparecer algunos problemas cuando dos objetivos de color parecido comparten la misma ROI por causa de una oclusión. El método propuesto no es capaz de seguir objetos de color similar si se ocluyen unos a otros. Este problema se puede resolver utilizando una técnica adecuada de filtrado y asociación de datos.

En esta Tesis se estableció como objetivo del método propuesto funcionar en tiempo real. Para conseguir este objetivo se ha utilizado una nueva estructura piramidal. La máscara y el objetivo han sido representados utilizando esta pirámide. Estas representaciones se generan segmentando la región del fotograma de entrada dónde el objetivo es más probable que

esté localizado. Esta segmentación es la parte del algoritmo de seguimiento que consume más tiempo. Así, para conseguir funcionamiento en tiempo real, el método de segmentación utilizado debe ser lo más rápido posible. Además, la pirámide es utilizada para realizar el proceso de correspondencia de forma jerárquica. De esta forma, cuánto menor sea el tiempo que se tarda en recorrer la pirámide, menor será el tiempo para llevar a cabo el proceso de correspondencia. Para encontrar una estructura piramidal que cumpliera estas características, las principales estructuras piramidales (regulares e irregulares) presentes en la literatura fueron detalladamente estudiadas durante el desarrollo de esta Tesis. Ambos tipos de pirámides -regular e irregular- tienen ventajas y desventajas. Las pirámides regulares pueden ser construidas y recorridas con un bajo coste computacional, pero tienen problemas importantes debidos a la inflexibilidad de su estructura. Las pirámides irregulares solucionan los problemas de las regulares a cambio de un coste computacional que hace que no puedan ser utilizadas en aplicaciones de tiempo real. En esta Tesis, se ha presentado una nueva estructura piramidal: la Pirámide Irregular Acotada (*Bounded Irregular Pyramid (BIP)*). Esta pirámide surge debido a la necesidad de tener una pirámide irregular con unos resultados similares al resto de pirámides irregulares pero más rápida de construir y recorrer. La idea clave de la BIP es usar una estructura regular $2x2/4$ en las zonas homogéneas de la imagen de entrada y una estructura irregular de grafo simple en el resto de regiones. La parte irregular de la BIP soluciona los problemas de las estructuras regulares y su parte regular reduce su complejidad computacional. La BIP permite que el sistema completo funcione en tiempo real con imágenes de 128x128 píxeles (27 fotogramas por segundo) en un PC Pentium IV a 3GHz. Se ha probado que los resultados de segmentación obtenidos con la BIP son similares a los resultados obtenidos con otras estructuras irregulares pero reduciendo al menos diez veces el tiempo de cómputo.

El algoritmo de seguimiento propuesto ha sido utilizado en dos aplicaciones en tiempo real: un mecanismo atencional y un sistema de captura de movimiento humano. Estas aplicaciones han sido presentadas brevemente en esta Tesis, destacando las contribuciones del algoritmo de seguimiento. En el mecanismo atencional, el algoritmo de seguimiento es utilizado en la etapa preattentiva para implementar la inhibición de retorno, evitando que el mismo objeto sea extraído en diferentes fotogramas. El algoritmo de seguimiento permite al mecanismo atencional manejar entornos dinámicos a 5 fotogramas por segundo. En el sistema de captura de movimiento, el algoritmo de seguimiento es utilizado para seguir en tiempo real los movimientos de la cabeza y las manos de la persona cuyos movimientos están siendo capturados.