Real-time Object tracking using Bounded Irregular Pyramids ⋆

R. Marfil, L. Molina-Tanco, J.A. Rodríguez and F. Sandoval

Departamento de Tecnología Electrónica
E.T.S.I. Telecomunicación, Universidad de Málaga
Campus de Teatinos, 29071-Málaga, Spain

Abstract

Target representation and localization is a central component in visual object tracking. In this paper a new approach for target representation and localization is presented. This approach tackles two of the most important causes of failure in object tracking: changes of object appearance and occlusions. We propose a modified template matching approach which does not require a priori learning of object views. This method 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. Our hierarchical tracker allows tracking of multiple objects with low increase of computational time. The capability of the tracking system to handle occlusions and target distortions is demonstrated for several video sequences.

Key words: Non-rigid object tracking, multiple object tracking, target representation and localization, hierarchical template matching, template-based tracking.

1 Introduction

Object tracking has been one of the main fields of study in Computer Vision for the last 20 years. Real-time object tracking is a critical task in many applications such as surveillance, object-video compression and driver assistance. Typically, a visual tracking system can be divided into two major components:

⋆ This work has been partially granted by the Spanish Ministerio de Educación y Ciencia and FEDER funds project No. TIN2004-05961

Email address: rebeca@uma.es (R. Marfil).

i) target characterization and localization, and ii) filtering and data association (Comaniciu et al. (2003)). The first component is mostly a bottom-up process, which must be capable of dealing with changes in appearance and partial occlusions of the target, while the second component is usually a top-down process dealing with the dynamics of the objects and the evaluation of different assumptions. The way the two components are weighted and integrated in the same framework depends on the final application, having a great influence in the robustness and efficiency of the tracking process.

In this paper, we propose a system to track one or multiple objects in cluttered scenes. Typically, when the goal is to track objects in cluttered scenes, the application relies more on target representation and localization (Comaniciu et al. (2003)). Therefore, this work puts the emphasis in the first component, presenting a new approach to target representation and localization which allows handling of occlusions and appearance changes of the objects and performs the tracking process in real time.

Five main object tracking approaches have been developed depending on the target representation (Cavallaro et al. (2005)):

- Model-based methods (Koller et al. (1993)), which employ a priori knowledge about the geometry of objects in a given scene. Therefore, they present two major drawbacks: the need for object models with detailed geometry for all objects that could be found in the scene, and the lack of generality. Besides, they are usually computationally expensive.
- Appearance-based methods (Jepson et al. (2003)), which 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 (Blake et al. (1995)), which track only the contour of the object. Usually they use active contour models like snakes, B-splines or geodesic active contours.
- Feature-based methods (Shi and Tomasi (1994)), which use features of an object to track parts of it. Although these approaches are very stable even in case of partial occlusions, the problem of grouping the features to determine which of them belong to the same object is its current major drawback.
- Hybrid methods (Cavallaro et al. (2005)), which are designed as a hybrid between a region-based and a feature-based technique. 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 paper 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 (Jepson et al. (2003)):

- Template-based models (Nguyen and Smeulders (2004)) which use an image sample or template of the target to track.
- View-based models (Ho et al. (2004)), which are usually learned with Principal Component Analysis. They have the advantage of modelling 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, which usually have problems when motions of the target and background are similar. They are usually improved by accumulating an appearance model through time (Irani et al. (1994)) or estimating both motion and appearance simultaneously (Jepson et al. (2003)). These methods are computationally expensive.
- Global statistic based methods (Comaniciu et al. (2003)), which use image statistics to represent the tracked object.

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 (Nummiaro et al. (2003)). Therefore, colour distributions have been used to track non-rigid objects like heads (Raja et al. (1999)) or hands (Martin et al. (1998)). A variety of statistical techniques have been used to model the colour distribution (Elgammal et al. (2002)). Thus, Raja et al. (1999) modelled the colour distribution of an object using a mixture of Gaussians fitted using the EM (Expectation Maximization) algorithm. The major drawback of this parametric technique is 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 (Scott (1992)), it has been successfully used to track hands (Martin et al. (1998)) or other non-rigid objects against cluttered backgrounds (Comaniciu et al. (2003)). Besides, colour histograms can be easily quantized into a small number of bins to satisfy the low-computational cost imposed by 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 (Comaniciu et al. (2003)). In order to avoid it, the similarity function can be regularized by masking the objects with an isotropic kernel in the spatial domain (Elgammal et al. (2002)).

Template-based models can be seen as a way to combine colour information

with spatial information. 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 behind this classical idea 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 occlusion of the object.

To do that, the tracker should: i) update the template to accommodate the change of object appearance and, ii) detect the occlusion and recapture the object when the occlusion ends.

In order to achieve these two goals, the algorithm proposed in this paper uses a hierarchical template-based model which is built using a Bounded Irregular Pyramid (BIP) (Marfil et al (2004, 2004b)). 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. This hierarchical matching runs in real time ($25 \sim 30$ Hz in a Pentium 850 MHz PC)).

The paper is organized as follows: Section 2 describes the target and template representation using BIP. 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.

2 Target and template representation

2.1 Overview

In template-based tracking, the simplest template representation one could use would be a fixed template of the target to track. This approach would be reliable over a short period of time, but it would cope poorly with appearance changes over longer periods that occur in most applications. In general, the template should be updated over time. A fast updating scheme that acquires the template from the preceding frame (Sidenbladh et al. (2000)) will fail at the presence of occlusions or abrupt changes in illumination 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. (2000, 2002) the template is updated using a weighted sum between the old template and current data. Their tracking approach combines compact object shape, motion, and appearance in a Bayesian framework. Nguyen et al. (2001) tracked rigid objects using a template matching approach where the intensities in the template are estimated by robust and adaptive Kalman filters. They use 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 (Nguyen and Smeulders (2002, 2004)), 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 of 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 (1997) proposed a template matching approach based on a multi-resolution search strategy. The search takes into account not only translation of a gray-scale template but also affine transformations. 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. A pyramid is a hierarchical structure where the base is the original image and each pyramid level is recursively obtained by processing its underlying level. A set of links connects the

nodes between levels, creating son-father relationships among them. Most of previous work present in the literature is related to image registration and not to template-based tracking. However in both applications the interest area is represented as a template. First attempts to use low-level resolution images in template matching were done by Vanderbrug and Rosenfeld (1977). They used a subwindow first to find likely 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 Rosenfeld and Vanderbrug (1977), it was 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. (1997) 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 Dani and Chaudhri (1995). Wong and Hall (1978) combined the sequential similarity detection algorithm (SSDA) with pyramidal speed-up. Thévenaz et al. (1998) applied a cubic spline based pyramid along with the minimization of the mean square intensity difference between the images. Kumar et al. (1998) 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 (1998). Krüger and Sommer (2000) and Feris et al. (2001) have proposed Wavelet pyramids 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 The Bounded Irregular Pyramid structure

In this work, the Bounded Irregular Pyramid (BIP) (Marfil et al. (2004b)) is used to represent targets and templates in order to reduce the computational cost of the template matching process. This section briefly describes the structure of this irregular pyramid.

The BIP is a tool to represent an image or a part of an image in a hierarchical way. To reduce the computational cost of building this representation, the BIP combines in the same pyramid a regular approach –for homogeneous regions– with an irregular one –for the rest of the image. Therefore, the data structure of the BIP is a combination of a 2x2/4 regular structure with a simple graph irregular structure. The whole structure can be described as a graph hierarchy for which each level l is represented by a graph $G_l(N, E)$ consisting of a set of nodes, N_l, linked by a set of arcs or edges, E_l. There are two types of nodes: nodes belonging to the 2x2/4 structure, named regular nodes,

and virtual nodes or nodes belonging to the irregular structure (Marfil et al. (2004)). In order to develop an algorithm which is robust to strong and abrupt illumination variations, each node of the structure is characterized by the Hue (H), Saturation (S) and Brightness (V) components of the HSV colour space. In this work, we consider that two nodes are similar or have similar colour if their Euclidean distance in HSV space is less than a similarity threshold T_c. Fig. 1 shows the schematic representation of the BIP structure associated to an object, where regular and virtual nodes are present in the different levels of the representation. Regular nodes are generated by 2x2 blocks of nodes of the level below. Virtual nodes are generated by combination of two or more neighbour nodes, regular or virtual. In the figure, links of the regular part of the BIP are not shown in order to make the structure better visible. Virtual nodes can be in contact to other nodes. These intra-level links are shown in the figure as dotted lines.

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

(1) Regular decimation process. A regular pyramid can be represented as a hierarchy of image arrays where the nodes are represented by their positions in such arrays. In the regular part of the BIP each regular node n is represented by (i, j, l), where l represents the level and (i, j) are the (x, y) coordinates 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 node has associated five parameters:

- Homogeneity, $Hom(i, j, l)$. This parameter is used to identify and build the regular part of the BIP. Regular nodes have $Hom(i, j, l) = 1$. $Hom(i, j, l)$ of a node is set to 1 if the four nodes immediately underneath have similar colour and their homogeneity values are equal to 1. Otherwise, it is set to 0. Nodes with $Hom(i, j, l) = 0$ are not taken into account. In Fig. 5 only the nodes of the represented object are homogeneous nodes.
- Chromatic phasor, $S_{\angle H}(n)$. The chromatic phasor of a regular node n is equal to the average of the chromatic phasors of the nodes in its reduction window. The reduction window of a node is the set of its sons in the level below.
- V value or luminosity value, $V(n)$. The V value of a node n is equal to the average of the V values of the nodes in its reduction window.
- Area, $A(n)$. The area of a node is equal to the sum of the areas of the nodes in its reduction window.
- Parent link, $(X, Y)_{(i,j,l)}$. The parent link of the four cells immediately underneath (sons) of a regular node (i, j, l) are set to (i, j). The parent link thus indicates the position of the parent of a regular node in its upper level (a regular node without parent has a parent link set to a NULL value). Parent links represent the inter-level edges of the regular part of the BIP.

The regular part of the BIP can be seen as an incomplete regular pyramid with only nodes associated to homogeneous regions in the base level, and it can be represented as a hierarchy of incomplete image arrays. In each of these arrays two nodes are neighbours if they are placed in adjacent positions of the array. If two nodes are neighbours in a level l, their receptive fields are neighbours in the base level. The receptive field of a node is the set of its sons in the base level. Fig. 1 shows how five regular nodes have been generated at level 1, and only one regular node is presented at level 2.

(2) Parent search and intra-level twining. Each regular orphan node (i, j, l) searches for the regular neighbour node (i_1, j_1, l) with parent $(i_p, j_p, l + 1)|n_p \in N_{l+1}$ most similar to it and it 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 a virtual node ($n_p \in N_{l+1}$) of G_{l+1}. If the studied node does not find a parent to link to it, then it looks for the most similar neighbour regular node without parent. Both are twinned, and a new virtual node in G_{l+1} is generated(*Intra-level twining*). In Fig. 1, it can be noted that two sets composed of four and six nodes of level 0 have generated the two irregular nodes of level 1. These sets have been generated in the intra-level twining and parent search step. It can be also appreciated that two orphan nodes have been linked to a regular node of level 1.

(3) Virtual parent search and virtual nodes linking. From each virtual node of G_l without parent, a search is made for the most similar virtual node with parent in its vicinity. If for the studied node a neighbour is found which satisfies these conditions then the studied node is linked to this parent (*Virtual parent search*). In other case, the studied virtual node looks for the virtual node without parent in G_l most similar to it, in order to generate a virtual node in G_{l+1} (*Virtual nodes linking*). Fig. 1 shows that the two virtual nodes of level 1 have generated an unique node at level 2.

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

The hierarchy stops to grow when is not possible to link together any nodes because they are not similar.

The BIP shares the low computational cost of regular pyramids and some desired properties of certain irregular pyramids as preservation of connectivity, shift invariance and ability to represent elongated objects. Table 1 shows the computational time to build several irregular pyramidal structures. These times have been computed using a set of 30 images with size 256 x 256 pixels.

t_{min} is the minimum time, t_{max} is the maximum and t_{ave} the average time. These results show that the BIP is consistently faster to compute than all the other irregular pyramids evaluated.

2.3 Target and template representation

In order to reduce the computational load associated with the template matching process, the Bounded Irregular Pyramid has been used in this work to obtain the target and template representations. Thus, in the proposed system, each target T and template M are represented using the regular nodes of BIP structures:

$$M^{(t)}(l) = \bigcup_{ij} m^{(t)}(i,j,l) \tag{1}$$

$$T^{(t)}(l) = \bigcup_{ij} q^{(t)}(i,j,l) \tag{2}$$

being $M^{(t)}(l)$ and $T^{(t)}(l)$ the level l of the pyramid structures corresponding to the template and the target in the frame t respectively. Each level of the template is made of a set of regular nodes $m^{(t)}(i,j,l)$. Equivalently, each level of the target is made of a set of regular nodes $q^{(t)}(i,j,l)$. These levels are successively reduced versions of the template and the target, respectively. The process employed to generate these hierarchical representations is slightly different to the previously explained one in Section 2.2, because they are integrated inside the tracking process. Therefore, they are related to the results of the template matching procedure. Although these representations only have regular nodes, these nodes are obtained from a complete BIP structure. The way in which these BIP structures are constructed is explained in the following section.

3 Single object tracking

In this section the description of the algorithm to track a single object in a cluttered scene is presented. The algorithm works in four consecutive stages. Firstly, it performs an over-segmentation of the Region of Interest (ROI) using BIP, i.e. the input image region where it is more likely that the target will be. This over-segmentation stage increases the stability and the robustness of the tracking process, because it takes into account neighbourhood information in the tracking of each pixel. Secondly, the target is searched by means of a template matching procedure. Once the target has been correctly localized in

the current frame, a refinement step improves the appearance of the target. Finally, the template is updated with the information provided by the last localized target. It must be noted that the tracking procedure is integrated in the same hierarchical structure where the over-segmentation and refinement stages are performed. The use of this hierarchical structure allows the whole process to run in real time ($25 \sim 30$ Hz in a Pentium 850 MHz PC). The data flow diagram of the proposed tracking algorithm is given in Fig. 2.

The target to track is chosen manually from the first frame of the video sequence. To do that, we use a colour segmentation algorithm which is also based on a BIP structure (Marfil et al. (2004)), but in principle any other segmentation algorithm could be used. The target can be any of the resulting segmented regions. Fig. 3b shows the segmentation of a real scene. In this case, the segmented region associated to the hand has been manually selected as the target to track. Once the target is chosen, the algorithm extracts its BIP structure (the part of the BIP generated in the segmentation process which corresponds to the selected target). The regular nodes of this hierarchical structure generate the first template $T^{(0)}$. The spatial position of the target in the original image is the first region of interest (ROI). Level 0 of the BIP structure of the target is initialized by setting the target nodes as homogeneous nodes and by setting the rest of nodes as non-homogeneous ones. Then the BIP is built as explained in Section 2. Fig. 3b shows the ROI (marked in blue) corresponding to the selected target. The representation of this first template is shown in Fig. 4. Only coloured nodes are nodes of the structure (the white areas of the figure are used only to help in the figure representation). Fig. 5 shows another example of template representation. It can be appreciated that the different levels of the template are not connected. This is not an issue. The template representation is obtained from a BIP structure, which assures that the different regular nodes of the template are connected by virtual nodes. Therefore, the final region associated to the template *is always a connected region at the base level*.

The five main modules of the proposed tracking system (Fig. 2) are explained in the following sections.

3.1 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 3.5. The hierarchical structure is built as explained in Section 2.2, and can be

represented in each level as:

$$ROI^{(t)}(l) = \bigcup_k p_k^{(t)} \tag{3}$$

being $p_k^{(t)}$ a regular or a virtual node of the BIP built over the ROI. The colour similarity threshold T_{co} used in this segmentation process should be small enough to allow an over-segmentation of the ROI. By over-segmentation it is meant a segmentation in which the number of obtained regions is very high compared with the number of real regions in the image. This over-segmentation avoids a high dependency of the tracking method with the segmentation results. Thanks to this over-segmentation, neighbourhood information is taken into account in the tracking of each pixel. In this process the ROI is divided up in a set of homogeneous colour regions. Each node of $ROI^{(t)}$ belongs to one of this regions.

3.2 Template matching

After the hierarchical representation of $ROI^{(t)}$ has been obtained, the algorithm looks for the target $T^{(t)}$ using a hierarchical template matching approach. 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:

$$100 \cdot \frac{\sum_{ij \in M^{(t)}(l_w)} A(i,j,l_w)}{\sum_{ij \in M^{(t)}(0)} A(i,j,0)} > T_A \tag{4}$$

That is, l_w is the highest level whose template area is at least a $T_A\%$ of the total area of the template. The working level value depends on the size and the shape of the template. However, this is not a critical parameter of the algorithm. In our case, a threshold value of $T_A = 50$ has demonstrated to be adequate for all experiments 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. The displacement of the template can be represented as $d_k^{(t)} = (d_k^{(t)}(i), d_k^{(t)}(j))$, being $d_0^{(t)}$ the first displacement and $d_f^{(t)}$ the final displacement. $d_f^{(t)}$ is the displacement that situates the template in the position where the target is placed in the current frame. The algorithm chooses as initial displacement in the current frame $d_0^{(t)} = d_f^{(t-1)}$. In order to localize the target and obtain $d_f^{(t)}$, the overlap $O^{(t)}_{d_k^{(t)}}$ between $M^{(t)}(l)$ and $ROI^{(t)}(l)$ for each template displacement k is calculated:

$$O^{(t)}_{d_k^{(t)}} = \sum_{ij \in \xi} w^{(t)}(m(i, j, l_w^{(t)})) \tag{5}$$

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 3.4. ξ is the subset of nodes that satisfy the following condition:

$$g(r, s) < T_c, \tag{6}$$

with

$$r = f(m^{(t)}(i, j, l_w^{(t)}), a(t))$$
$$s = p^{(t)}(i + d_k^{(t)}(i), j + d_k^{(t)}(j), l_w^{(t)})$$

being $g(r, s)$ the colour distance between r and s and T_c a colour similarity threshold. $f(m^{(t)}(i, j, l_w^{(t)}), a(t))$ is a coordinate transformation of $m^{(t)}(i, j, l_w^{(t)})$ that establishes the right correspondence between $m^{(t)}(i, j, l_w^{(t)})$ and $p^{(t)}(i + d_k^{(t)}(i), j + d_k^{(t)}(j), l_w^{(t)})$. $a(t)$ denotes the parameter vector of the transformation, which is specific to the current frame. Although other transformations such as rotations or scale changes could be modelled using $a(t)$, 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. Eq. (6) is satisfied when a match occurs.

If there is a match between a node of the template and a regular node of the ROI, the overlap is incremented in a value equal to the weight of the node of the template (Eq. (5) and (Eq. (6)). In our experiments, we consider that the target has been found in a position if the overlap in that position is higher than 70%. All the ROI regular nodes that match with nodes of the template are marked as nodes of the target in the whole structure $ROI^{(t)}$. Thus, the hierarchical representation of the target $T^{(t)}$ is obtained.

3.3 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 section, the segmented region in which a node n_k is included will be denoted as $R(n_k)$.

In the target refinement step, for each regular node $p^{(t)}(i,j,l)$ of the ROI marked as node of the target $p^{(t)}(i,j,l) = q^{(t)}(i,j,l) \in T^{(t)}(l)$ a search is performed among its virtual 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_{cr} then all the regular nodes included in $R((q^{(t)}(i,j,l))$ are marked as target nodes. Thus, the hierarchical representation of $T^{(t)}$ is completed.

3.4 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, current template values tend to reflect the state of the process better than older template values. However, an excessively fast updating scheme would be sensitive to sudden tracking error. Therefore, the updated template should be a compromise between the current template and the data. In our case, we associate a weight with each node 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 node $m^{(t)}(i,j,l)$ of the template $M^{(t)}$ in the current frame t.

The value of this weight, $w^{(t)}(m(i,j,l))$, is always in the range [0,1]. The whole template is updated at each sequence frame:

$$m^{(t+1)}(i,j,l) = \begin{cases} m^{(t)}(i,j,l) & \text{if no match} \\ f^{-1}(q^{(t)}(i,j,l), a^{(t)}) & \text{if match} \end{cases} \tag{7}$$

$$w^{(t+1)}(m(i,j,l)) = \begin{cases} max(0, w^{(t)}(m(i,j,l)) - \alpha) & \text{if no match} \\ 1 & \text{if match} \end{cases} \quad (8)$$

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. Eq. (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 in the template matching step, or from the corresponding point $q^{(t)}(i,j,l)$ in the target via the inverse coordinate transformation $f^{-1}(q^{(t)}(i,j,l), a^{(t)})$ (Section 3.2), if there is match between template and target. Eq. (8) means that each weight point $w^{(t+1)}(m(i,j,l))$ is equal to 1 if there is match, or else 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. There is a match when Eq. (6) is satisfied.

The template must be updated with the nodes that were included in the target in the refinement step. Each of these nodes is included in $M^{(t+1)}$ using the same inverse transformation of Eq. (7). Their corresponding weights are set to 1. As was previously commented in this paper, although the template is composed only by regular nodes, it is influenced by the virtual nodes included in the BIP representation of the ROI.

Fig. 6 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.

3.5 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, $ROI^{(t+1)}(0)$ will be made up of the pixels of the next frame $p^{(t+1)}(i,j,0)$ which are included in the bounding box $BB(T^{(t)}(0))$ plus the pixels included in an 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.

$$ROI^{(t+1)}(0) = \bigcup_{ij} p^{(t+1)}(i,j,0) \quad (9)$$

with

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

This step is performed at the end of the tracking process t. In this work a ϵ value of 6 pixels in the case that the target was found in the previous frame has demonstrated to be adequate for all the experiments. If the target was not found in the previous frame the region of interest is obtained using the position where the target was found the last time. The ϵ value is incremented in one pixel until the target is found or the ϵ has a maximum value of 12 pixels.

(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 3.1.

3.6 Handling occlusions

The previously presented algorithm to track a single object can implicitly 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 and are handled in the same way as 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 by 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. In our experiments a maximum value of 12 pixels was sufficient to solve all total occlusion cases.
- 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 nodes 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. 7.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 the moving dot (light grey points), together with the trajectories generated by the proposed tracking algorithm (black gray points). It can be appreciated that the obtained trajectories are very similar to the real ones, in spite of partial and total occlusions (Figs. 7.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. 7.b, where the tracked dot is always occluded by the other one.

4 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. 2) 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 nodes (Section 2.2) only the nodes of $ROI_i^{(t)}(0)$ with $i \in [1..N]$, being N the number of objects. Thus, only the regions of interest [ROIs] are over-segmented. 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.

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 3.2

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 3.3.

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 3.6

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 Hang et al. (2005)

Fig. 8 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. 8 c-d. The trajectories obtained by our method are shown in Figs. 8 e-f. Similar conclusions to those of Section 3.6 and Fig. 7 can be extracted. The synthetic sequence shows several moving objects whose trajectories intersect at multiple points, resulting in occlusions from which the algorithm is able to recover.

5 Results

The proposed algorithm has been tested with a number of image sequences. In this section we present some representative results. The targets were usually human faces, hands and bodies, although sequences are also shown where rigid objects are tracked. As commented in Sections 3 and 4, templates are always initialized using a previous segmentation of the first frame of the video sequence.

In the first example, we applied the proposed tracking system for real-time face tracking. Fig. 9 shows the capability of the tracker to handle scale changes, rotations of the face, partial occlusions and changes of illumination. In the second example (Fig. 10), we track two objects simultaneously: a face and a green cone. This sequence demonstrates the ability of the proposed algorithm

to deal with overlaps between the tracked objects (i.e. frame 115) and total occlusions (in frame 108 the green cone is out of the image but it still is recovered in the next frames). In Fig. 11, the tracking of a rigid object (a ping-pong ball) is shown. The main difficulty of this sequence is that the movement of the ball from frame to frame is larger than its size. However, the ball is reliably tracked over the whole sequence.

In order to quantitatively assess the accuracy of the proposed tracking method, we generate ground truth data by manually selecting the tracked object from the input image (see Fig. 12.(a) and Fig. 12.(b)). Fig. 12.(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. 12.(d)). All the obtained targets are very close to the desired targets. The errors are mainly placed in the boundary of the target due to colour transitions. In order to calculate the percentage 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 the algorithm identifies as target pixels (background errors). The percentages of both types of error pixels for Fig.12 are shown in Table 2.

In Fig. 13 some of the results obtained by the proposed method and by the mean-shift based approach (Comaniciu et al. (2003)) 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 (Liu and Chen (2003)). It can be appreciated in Fig. 13 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.

Regarding the computational times, tracking the two objects of the sequence of Fig. 10 was done with a frame rate of 35 fps on a 850 MHz PC using 128x128 images [1]. When we track only the face in the same sequence the frame rate is 45 fps. If the image size is increased to 256x256 the frame rate is 11 fps and 17 fps for the two object tracking and the face tracking, respectively. In order to asses the advantages of using a hierarchical approach to perform the template matching process, we have compared the proposed hierarchical

[1] By fps it is meant the average number of frames the proposed method can treat within a second

template matching with a template matching carried out only in the base of the BIP structure. Although the results are similar, the computational time increases considerably (i.e. 3 fps for the 256x256 sequence of Fig. 10 and 17 fps for the same 128x128 sequence). Video sequences shown in Figs. 10, 12 and 9 are now public available at http://www.grupoisis.uma.es.

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 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 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.

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 our 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 we are used a value of 0.1, 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 we have used a T_A value of 80 %. 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. We have found a value of 70 % to be adequate for all our 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, we recommend to use small values of T_{co}, i.e. $T_{co} \in [5..20]$. Our 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 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. Several combinations of the parameters were selected and the best combination was chosen. In our tests, the best choices for the thresholds were $T_c = 60$ and $T_{cr} = 40$.

6 Conclusions and future work

Target representation and localization is a central component in visual object tracking. In this paper, a new approach for target representation and localization using a template-based appearance model is proposed. This approach allows tracking of non-rigid objects without a previous learning of different object views. 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. This weighted template and the way it is updated allow the algorithm to successfully handle partial occlusions and short-term total occlusions. Besides, the template and the target are hierarchically represented using BIP. This representation makes possible to perform the tracking algorithm using a hierarchical approach which reduces the computational cost and it allows the whole system to run in real time with 128x128 pixels images. This real-time performance makes possible to employ the proposed tracking algorithm in interactive applications. It has been successfully employed in a Human Robot Interaction application (Molina-Tanco et al. (2005)).

Our future work will look at introducing a filtering stage (i.e Kalman filter)

which will allow for prediction of the position of the tracked objects in the next frame and to dynamically adapt the size of the extra border ϵ used in the selection of $ROI^{(t+1)}(0)$ (see Section 3.5).

Acknowledgments

The authors would like to thank Dr. Liu and Dr. Chen for providing us with the Magnets video sequence. The test data employed in Figs. 7a-d was obtained from the European Union MUSCLE Network of Excellence (FP6-507752). This database was provided by Advanced Computer Vision GmbH - ACV.

References

Althof, R., Wind, M., Dobbins, J., 1997. A rapid and automatic image registration algorithm with subpixel accuracy. IEEE Transactions on Medical Imaging 16, 308-316.

Bertolino, P., Montanvert, A., 1996. Multiresolution segmentation using the irregular pyramid. Int. Conf. On Image Processing 1, 257-260.

Blake, A., Isard, M., Reynard, D., 1995. Learning to track the visual motion of contours. Artificial Intelligence 78, 101-134.

Brun, L., Kropatsch, W., 2003. Construction of combinatorial pyramids. E. Hancock and M. (Eds.), Graph based Representations in Pattern Recognition, LNCS 2726, Springer Verlag , 1-12.

Cavallaro, A., Steiger, O., Ebrahimi, T., 2005. Tracking video objects in cluttered background. IEEE Trans. on Circuits and Systems for Video Technology 15 (4), 575-584.

Comaniciu, D., Ramesh, V., Meer, P., 2003. Kernel-based object tracking. IEEE Trans. on Pattern Anal. and Machine Intell. 25(5), 564-577.

Dani, P., Chaudhuri, S., 1995. Automated assembling of images: Image montage preparation. Pattern Recognition 28, 431-445.

Elgammal, A., Duraiswami, R., Harwood, D., Davis, L., 2002. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. of the IEEE 90 (7), 1151-1163.

Feris, R., Krger, V., Cesar Jr., R., 2001. Effcient real-time face tracking in wavelet subspace. Proc. of the Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, Vancouver, BC, Canada .

Hang, B., Yang., C., Duraiwami, R., Davis, L. Bayesian filtering and integral image for visual tracking. Proc. of the 6th Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005), 2005.

Huart, J., Bertolino, P., 2005. Similarity-based and perception-based image segmentation. IEEE Int. Conf. on Image Processing (ICIP2005)(accepted to) .

Ho, J., Lee, K., Yang, M.-H., Kriegman, D., 2004. Visual Tracking Using Learned Linear Subspaces. Proc. of the Int. Conf. on Computer Vision and Pattern Recognition, 1, 782-789.

Irani, M., Rousso, B., Peleg., S., 1994. Computing Occluding and Transparent Motions. Int. Journal Computer Vision, 12(1), 5-16.

Jepson, A., Fleet, D., El-Maraghi, T., 2003. Robust online appearance models for visual tracking. IEEE Trans. Pattern Analysis and Machine Intelligence 25 (10), 1296-1311.

Koller, D., Danilidis, K., Nagel, H., 1993. Model-based object tracking in monocular image sequences of road tra.c scenes. Int. Journal Computer Vision 10 (3), 257-281.

Kropatsch, W., Haxhimusa, Y., 2004. Grouping and segmentation in a hierarchy of graphs. C.A. Bouman and E.L. Miller (Eds), Computational Imaging

II, SPIE , 193-204.
Krüger, V., Sommer, G., 2000. Affine real-time face tracking using gabor wavelet networks. Proc. Int. Conf. on Pattern Recognition , 3-8.
Kumar, R., Sawhney, H., Asmuth, J., Pope, A., Hsu, S., 1998. Registration of video to geo-referenced imagery. Proceedings of the Int. Conf. on Pattern Recognition (ICPR98) , 1393-1399.
Lallich, S., Muhlenbach, F., Jolion, J., 2003. A test to control a region growing process within a hierarchical graph. Pattern Recognition 36, 2201-2211.
Liu, T., Chen, H., 2003. Real-time tracking using trust-region methods. IEEE Trans. on Pattern Analysis and Machine Intelligence 26 (3), 397-402.
Marfil, R., Rodriguez, J., Bandera, A., Sandoval, F., 2004. Bounded irregular pyramid: a new structure for colour image segmentation. Pattern Recognition 37 (3), 623-626.
Marfil, R., Bandera, A., Rodriguez, J., Sandoval, F., 2004b. Real-time template-based tracking of non-rigid objects using bounded irregular pyramids, Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 1, 301-306.
Martin, J., Devin, V., Crowley, J., 1998. Active hand tracking. Proc. 3rd IEEE Int. Conf. Automatic Face and Gesture Recognition, 573-578.
Molina-Tanco, L., Bandera, J., Marfil, R., Sandoval, F., 2005. Real-time human motion analysis for human-robot interaction. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS05) (accepted to).
Nguyen, H., Worring, M., van den Boomgaard, R., 2001. Occlusion robust adaptive template tracking. Proc. IEEE Conf. on Computer Vision (ICCV01) 1, 678-683.
Nguyen, H., Smeulders, A., 2002. Template tracking using color invariant pixel features. Proc. of the Int. Conf. on Image Processing (ICIP02) , 569-573.
Nguyen, H., Smeulders, A., 2004. Fast occluded object tracking by a robust appearance .lter. IEEE Trans. on Pattern Anlysis and Machine Intelligence 26 (8), 1099-1104.
Nummiaro, K., Koller-Meier, E., Roth, D., Van Gool, L., 2003. Color-based object tracking in multi-camera environments. Proc. of the 25th Pattern Recognition Symposium (DAGM03) , 591-599.
Raja, Y., Mckenna, S., Gong, S., 1999. Tracking color objects using adaptive mixture models. Image Vision Computing 17, 225-231.
Rosenfeld, A., Vanderbrug, G., 1977. Coarse-fine template matching. IEEE Trans. on Systems, Man and Cybernetics 7, 104-107.
Rucklidge, W., 1997. Efficient guaranteed search for gray-level patterns. Proc. IEEE Conf. Computer Vision and Pattern Recognition , 717-723.
Scott, D., 1992. Multivariate density estimation. Wiley.
Shi, J., Tomasi, C., 1994. Good features to track. IEEE Conf. Computer Vision and Pattern Recognition , 593-600.
Shinagawa, Y., Kunii, T., 1998. Unconstrained automatic image matching using multiresolutional critical-point filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 994-1010.

Sidenbladh, H., Black, M., Fleet, D., 2000. Stochastic tracking of 3d human figures using 2d image motion. Proc. European Conf. Computer Vision 2, 702-718.

Tao, H., Sawhney, H., Kumar, R., 2000. Dynamic layer representation with applications to tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition 2, 134-141.

Tao, H., Sawhney, H., Kumar, R., 2002. Object tracking with bayesian estimation of dynamic layer representations. IEEE Trans. Pattern Analysis and Machine Intelligence 24 (1), 75-89.

Thévenaz, P., Ruttimann, U., Unser, M., 1998. A pyramidal approach to subpixel registration based on intensity. IEEE Transactions on Image Processing 7, 27-41.

Vanderbrug, G., Rosenfeld, A., 1977. Two stage template matching. IEEE Transactions on Computers 26, 384-393.

Wong, R., Hall, E., 1978. Sequential hierarchical scene matching. IEEE Transactions on Computers 27, 359-366.

Figure captions

Fig. 1. Schematic representation of the BIP associated to an object. Pixels of the object have been marked at level 0. See text for details
Fig. 2. Illustration of the tracking algorithm.
Fig. 3. a) Original image; b) Segmented image with the chosen target marked in red and the ROI marked in blue.
Fig. 4. Template hierarchical representation of the hand extracted from Fig. 3.
Fig. 5. Template hierarchical representation of a face.
Fig. 6. Updating object template: a) sequence frames of a moving hand; and b) updated template.
Fig. 7. a-c) Dot tracking results: real trajectories have been marked as light gray points and generated trajectories have been marked as black gray points.
Fig. 8. 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.
Fig. 9. Face tracking results by the proposed method with appearance changes and partial occlusions.
Fig. 10. Two objects tracking results by the proposed method.
Fig. 11. Tracking results by the proposed method with a fast moving.
Fig. 12. (a) sequence frames of a moving hand; (b) ground truth; (c) tracked targets with the proposed method; (d) error pixels.
Fig. 13. Comparison between the proposed method and by the mean-shift based approach by Comaniciu et al. (2003).

Table captions

Table 1. Processing times of different pyramids.
Table 2. Pixel errors (in numbers of pixels) in Fig. 12.

Table 1

	Processing times (sec)		
	t_{min}	t_{ave}	t_{max}
ClaIPyr (Bertolino and Montanvert (1996))	2.51	3.96	7.68
LocIPyr (Huart and Bertolino (2005))	1.71	2.78	6.13
MorIPyr (Lallich et al. (2003))	2.43	3.47	4.47
BIP (Marfil et al. (2004))	0.65	0.76	0.84
HieIPyr (Kropatsch and Haxhimusa (2004))	4.07	4.29	4.91
ComIPyr (Brun and Kropatsch (2003))	1.32	2.88	12.8

Table 2

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

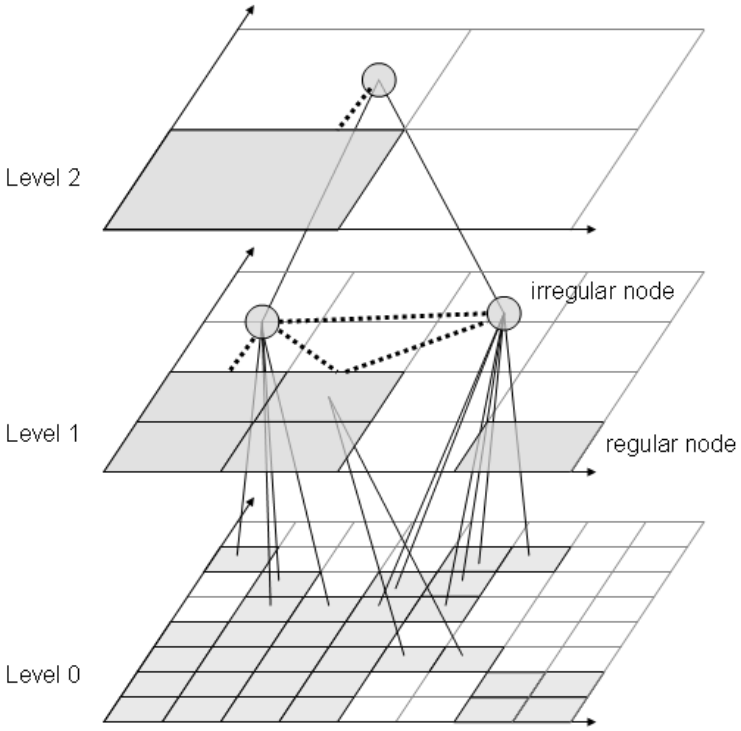

Level 2
irregular node
Level 1
regular node
Level 0

Fig. 1.

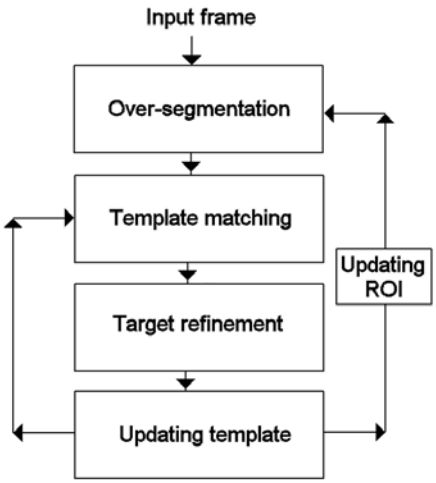
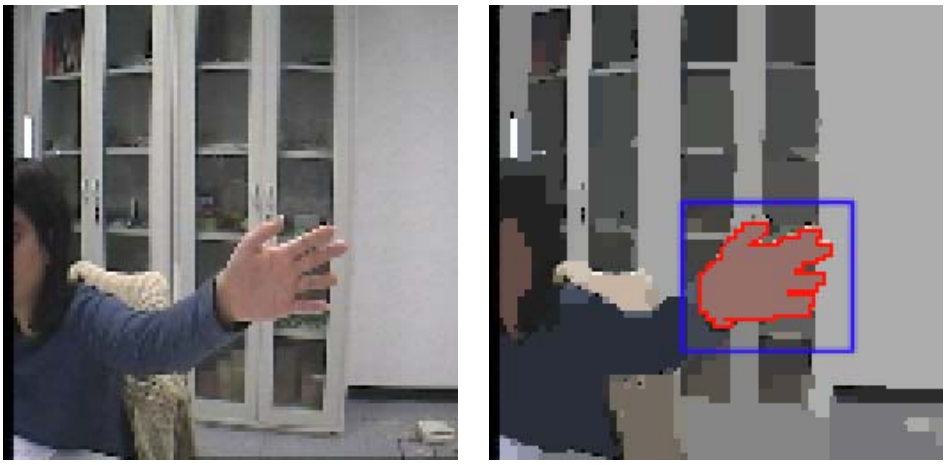
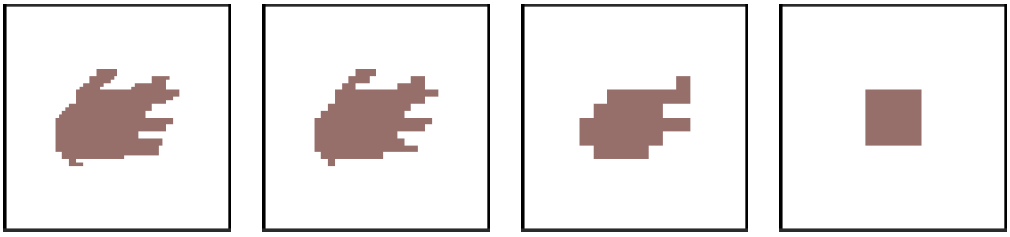

Input frame
Over-segmentation
Template matching
Target refinement
Updating template
Updating ROI

Fig. 2.



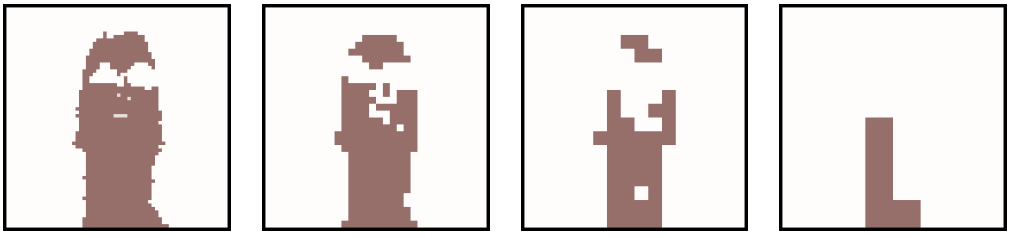
(a) (b)

Fig. 3.



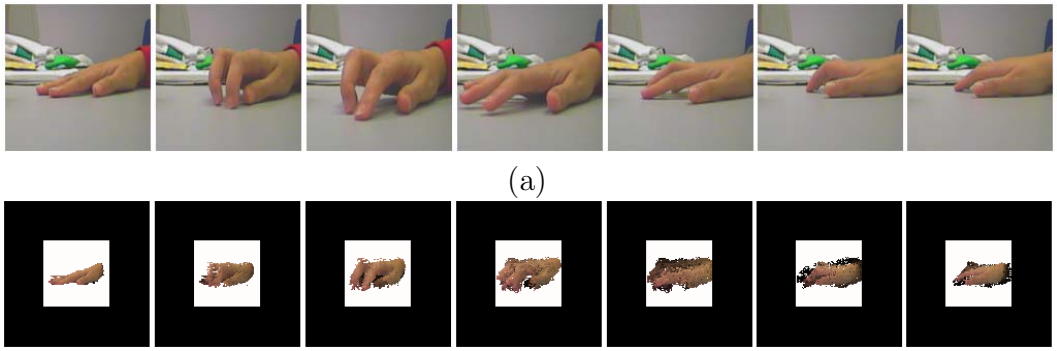
Level 0 Level 1 Level 2 Level 3

Fig. 4.



Level 0 Level 1 Level 2 Level 3

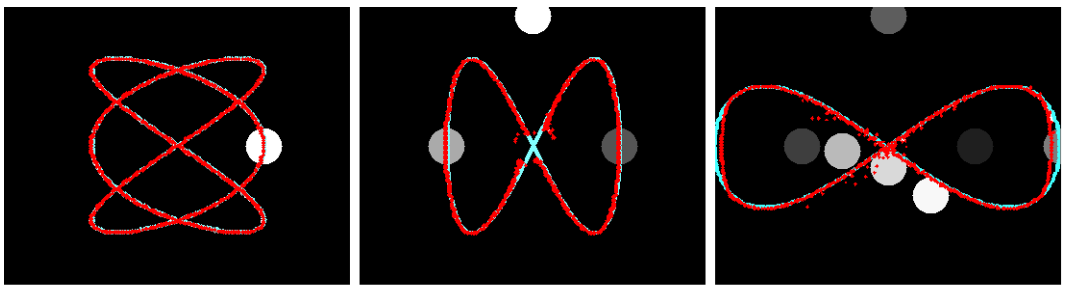
Fig. 5.



(a)

(b)

Fig. 6.



(a) (b) (c)

Fig. 7.

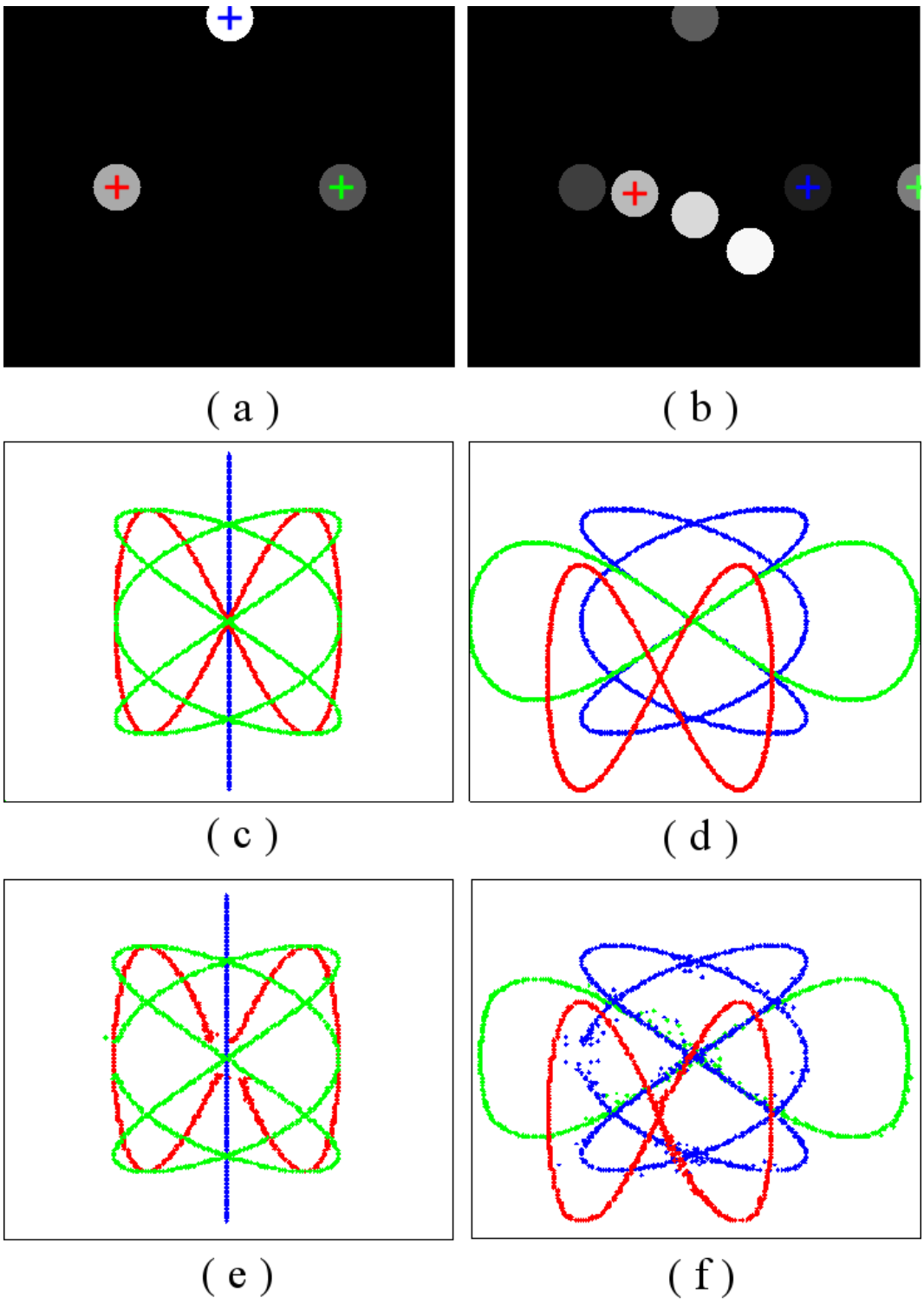


Fig. 8.

frame 1 frame 13 frame 14 frame 80 frame 84

frame 120 frame 124 frame 131 frame 150 frame 160

Fig. 9.

frame 50 frame 100 frame 108 frame 115 frame 120 frame 175

frame 185 frame 250 frame 335 frame 360 frame 400 frame 480

Fig. 10.

frame 10 frame 14 frame 26 frame 29 frame 40

Fig. 11.

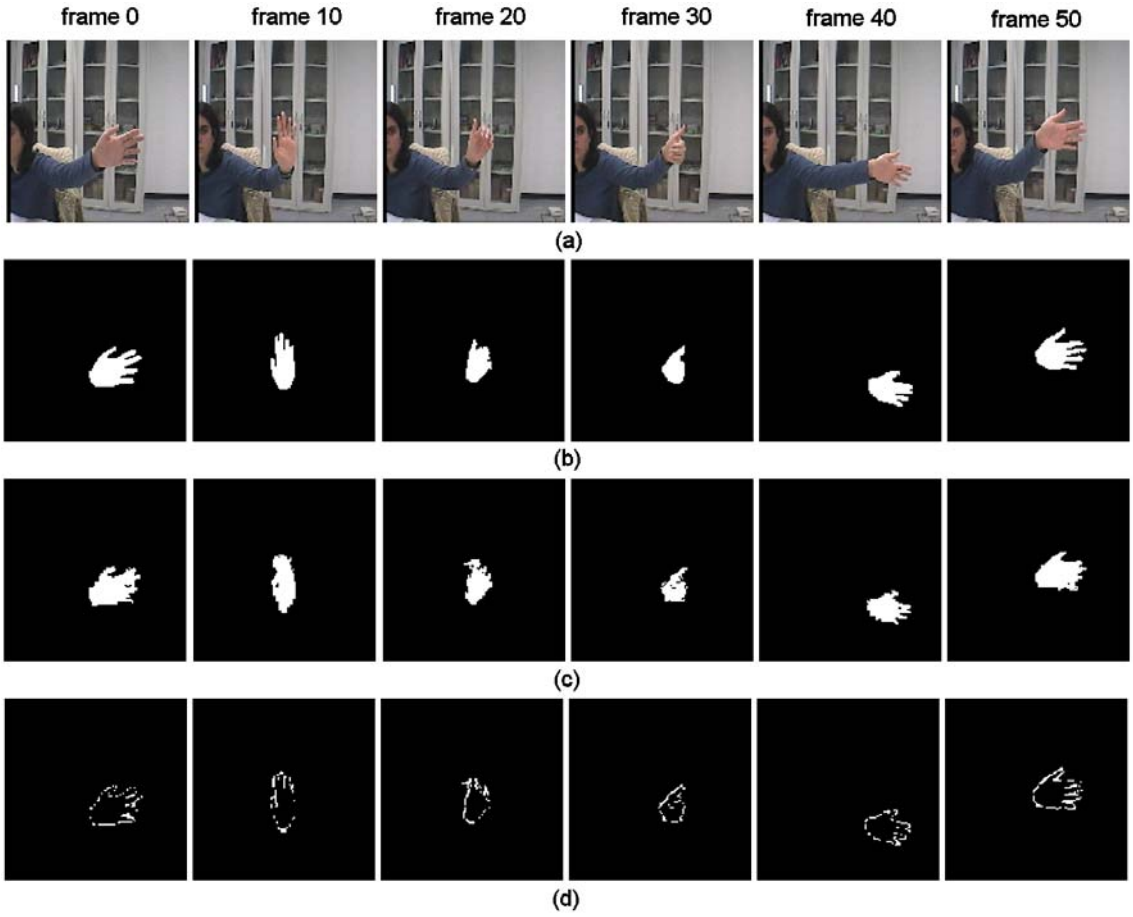
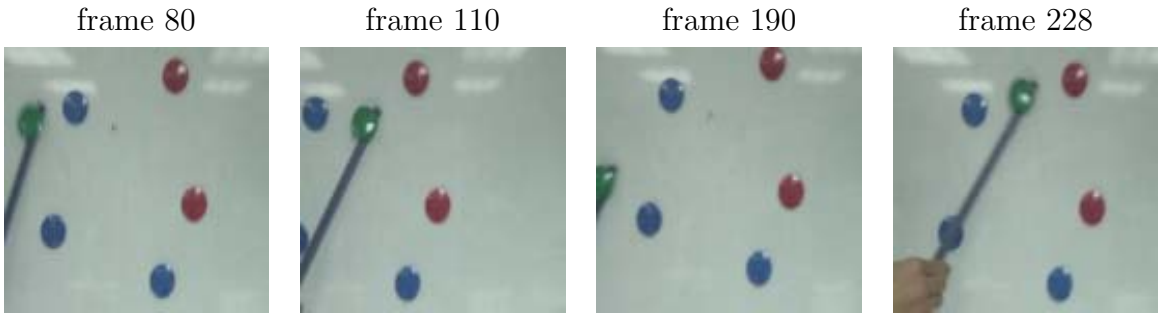
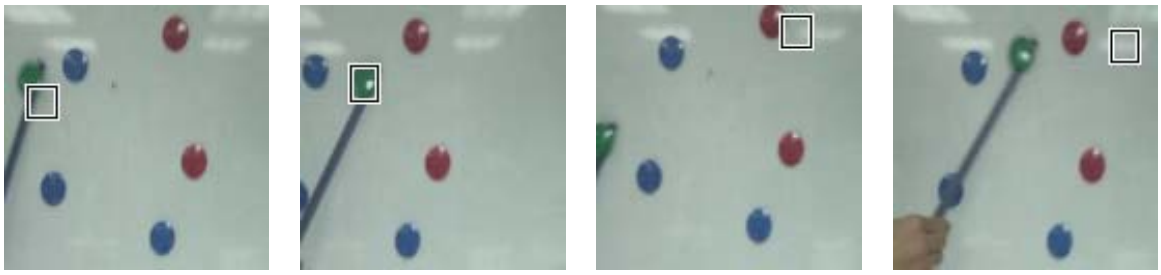


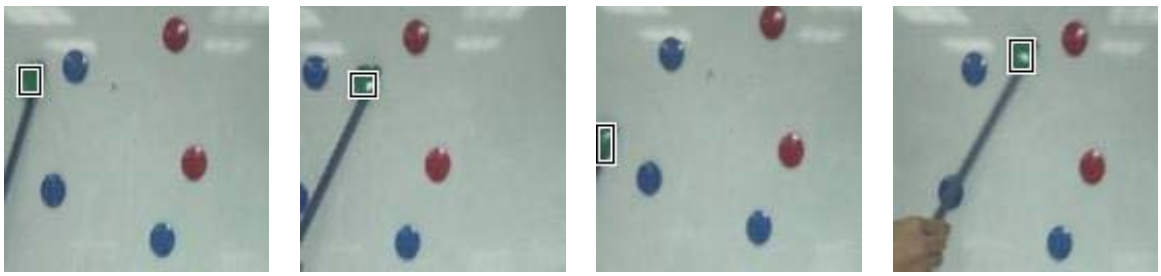
Fig. 12.



(a) some frames of the Magnets video sequence



(b) results obtained with the mean-shift based approach by Comaniciu et al. (2003)



(c) results obtained with the method proposed in this paper

Fig. 13.