# Real-time Template-based Tracking of Non-rigid Objects using Bounded Irregular Pyramids

R. Marfil, A. Bandera, J.A. Rodríguez, F. Sandoval
Departamento de Tecnología Electrónica
E.T.S.I. Telecomunicación, Universidad de Málaga
Campus de Teatinos, 29071-Málaga, Spain
Email: rebeca@dte.uma.es

***Abstract*—In object tracking, change of objects aspect is the most important cause of failure. This paper proposes a modified template matching approach to solve this problem without a priori learning of object views. This method permits 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. The capability of the tracking system to handle partial occlusions and target distortions is demonstrated for several video sequences.**

## I. Introduction

Real-time object tracking is a critical task in many computer vision applications such as surveillance, object-video compression and driver assistance. Typically, a visual tracking system can be divided into two mayor components: target characterization and localization, and filtering and data association [1]. The first component is mostly a bottom-up process, which must be capable to deal successfully with the changes in the appearance of the target, meanwhile the second is usually a top-down process dealing with the dynamic of the object movement and the evaluation of different assumptions.

The proposed tracking system integrates both components into the same framework. In these situations, the way the two components are combined and weighted depends on the final application, having a great influence in the robustness and efficiency of the tracking process. In our case, we propose a system to track objects in cluttered scenes. Therefore, the tracking system relies more on target representation and localization than on target dynamics [1].

Basically, the algorithm works in four consecutive stages. Firstly, it performs an over-segmentation of the input image region where is more likely that the target will be. This segmentation stage increases the stability and the robustness of the tracking process, because it permits to take into account neighborhood information in the tracking of each pixel. To avoid the excessive computational cost associated with a segmentation task, this segmentation is performed in a hierarchical way by using a bounded irregular pyramidal structure [6]. Secondly, the target is searched by means of a template matching procedure. After the target has been correctly localized in the current frame, a refinement step improves the appearance of the localized target. Finally, the template is updated by using 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 makes possible that the whole process runs in real time (25  30 Hz in a Pentium 850 MHz PC).

The paper is organized as follows: Section 2 describes the target and template representation. Section 3 presents the hierarchical tracking algorithm. Section 4 shows several experiments and, finally, section 5 gives some conclusions.

## II. Target and template representation

In order to represent the target to track, a feature space must be chosen. The 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 [10]. Therefore, colour distributions have been used to track nonrigid objects like heads [11] or hands [7]. A variety of statistical techniques have been used to model the colour distribution [4]. Thus, Raja et al [11] modelled the colour distribution of an object using a mixture of Gaussians fitted using the EM 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 [12], it has been successfully used to track hands [7] or other non-rigid objects against cluttered backgrounds [1]. Besides, the colour histograms can be easily quantized into a minor number of bins to satisfy the low-computational cost imposed by real-time processing. One of the main drawback 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 [1]. In order to avoid it, the similarity function can be regularized by masking the objects with an isotropic kernel in the spatial domain [4].

Other techniques try to merge colour and shape information to characterize objects. Thus, Gevers and Smeulders [5] combined colour and shape invariants for object recognition based on geometric algebraic invariants computed from colour co-occurrences. Although the algorithm was efficient, its discriminative power decreases by the amount of invariance. In

[3], the colour and shape information was combined in one dimensional vector that provided better results than colour-based or shape-based methods. In these two methods, the object matching is performed using appearance features invariant to viewpoint (invariant-based methods). However, they likely fail in case of severe changes of viewpoint, when a completely unseen side of an object moves into view. To handle this drawback, the view-based methods use considerably more a priori knowledge on the object [2]. The disadvantage of these methods is that they need an a priori trained appearance model which is not always available in practice.

This paper is concerned with tracking objects in image sequences using template matching. In fact, it aims for robust tracking under severe changes of viewpoints in the absence of an a priori model. In order to use a template matching process, it should be able to solve its two main drawbacks:

- the template does not represent the current object appearance,
- the target is partially or totally occluded.

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. Thus, the template could be computed as a weighted average between the previous template and the current localized target [13]. In [8], the template is estimated by robust and adaptive Kalman filters. 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 [9], where photometric invariant colour features are used. These approaches are pixel-based, and they do not take into account the colour of neighboring pixels. To solve this problem, we propose to simultaneously perform a segmentation step which is combined with the template-based tracking method to enhance the robustness of the tracking procedure.

### *A. Target and Template Hierarchical Models*

In order to reduce the computational cost of the template matching process and to achieve a real-time system, the proposed method uses a Bounded Irregular Pyramid (BIP) [6] to represent the target and the template. A pyramid is a hierarchical structure where the base is the original image and each pyramid level is recursively obtained by processing its underlaying level. There are a set of links that connect the nodes between levels, creating son-father relationships among them. In the literature, we can find several types of pyramids depending on the way that a level is processed to generate a new level. Specifically, the bounded irregular pyramid is a 4 to 1 structure where each level is generated by reducing the resolution of the previous one by a factor of four, Thus, a node of a new level $l$ is generated by averaging the colour of the four nodes immediately below at level $l-1$. But, on the contrary that other 4 to 1 structures, the BIP has valid nodes and non-valid ones. A valid node is generated when the four nodes below have similar colour. The non-valid nodes are not taken into account and they are removed from the structure.

Each pyramidal node n is identified by $(i,j,l)$ where $l$ represents the level and $(i,j)$ are the (x,y) coordinates within the level. 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. To build the different levels of the pyramid, each node has associated five parameters:

- Homogeneity, $Hom(i,j,l)$. $Hom(i,j,l)$ is set to 1 if the four nodes immediately underneath have colour difference values below a threshold $T_C$ and their homogeneity values are equal to 1. Otherwise, it is set to 0.
- Chromatic phasor, $S_H(i,j,l)$. The chromatic phasor is composed of the saturation (S) and the hue (H) values of the HSV colour space. If the cell is homogeneous, $S_H(i,j,l)$ is equal to the average of the chromatic phasors of the four cells immediately underneath. If the cell is not homogeneous, $S_H(i,j,l)$ is set to a null value.
- Intensity, $V(i,j,l)$. If the cell is homogeneous, $V(i,j,l)$ is equal to the average of the intensity values associated to the four nodes immediately underneath. Otherwise, it is set to a null value.
- Area, $A(i,j,l)$. It is equal to the sum of the areas of the four nodes immediately underneath.
- Parent link, $(X,Y)(i,j,l)$. If$Hom(i,j,l)$ is equal to 1, the values of the parent link of the four cells immediately underneath are set to $(i,j)$. Otherwise, these four parent links are set to a null value.

It must be noted that only nodes presenting an homogeneity value equal to 1 are valid nodes. Each valid node is linked to a homogeneous region at the base.

Thus, in the proposed system, the target $T$ and the template $M$ are represented by using 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 pyramidal structures corresponding to the template and the target in the frame $t$ respectively. Each level of the template is made up of a set of valid nodes $m^{(t)}(i,j,l)$. And, each level of the target is made up of a set of valid nodes $q^{(t)}(i,j,l)$.

Fig. 1 presents an example of template representation using a 4 levels pyramid. The base of the pyramid (level 0) presents $64 \times 64$ pixels. Fig. 1 shows how pixels at level $l$ are arranged into sets of $2 \times 2$ elements to create a node at level $l+1$. It must be noted that nodes related to non homogeneous sets (white nodes in the figure) are removed from the structure and they are not taken into account in the tracking procedure.

## III. Algorithm Description

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 image segmentation also based on a bounded irregular pyramid structure [6]. After choosing the target, the algorithm extracts its hierarchical representation. This hierarchical structure is the first template $T^{(0)}$ and its spatial position is the first region of interest (ROI), i.e. the portion of the current frame where the target is more likely placed.

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

### A. 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 it is described in subsection III-E. The hierarchical structure is built as it is explained in section II.A, and it can be represented in each level as:

$$ROI^{(t)}(l) = \bigcup_{ij} p^{(t)}(i,j,l) \tag{3}$$

being $p$ a node of the bounded irregular pyramid built over the ROI.

It must be noted that, after this structure generation, valid nodes without parent are regarded as roots of trees defined by their links to lower level nodes. Thus, they perform an over-segmentation of $ROI^{(t)}$ by defining classes at the base of the structure ($ROI^{(t)}(0)$). This over-segmentation permits to take into account neighborhood information in the tracking of each pixel.

### B. Template Matching

After the hierarchical representation of $ROI^{(t)}$ has been obtained, the algorithm looks for the target $T^{(t)}$. To do that, the algorithm uses a hierarchical template matching approach. Thus, 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 $l_w^{(t)}$ in the current frame the higher level where this matching can be correctly achieved. It permits 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:

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

That is, $l_w$ is the highest level whose template area is at least a TA% 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. Anyway, it is not a critical parameter of the algorithm. In our case, a threshold value of TA=50 has demonstrated to be adequate for all experiments.

**Target localization.** The process to localize the target in the current frame t is a top-down process which starts at the working level $l^{(t)}w$ and stops in 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 whole covered. If $ROI_{(t)}(l)$ was whole 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 $d^{(t)}k = (d^{(t)}k(i), d^{(t)}k(j))$, being $d^{(t)}0$ the first displacement. $d^{(t)}f$ the final displacement. $d^{(t)}f$ 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 $d^{(t)}0 = d^{(t-1)}f$. In order to localize the target and obtain $d^{(t)}f$, the overlap $O_{(t)_{d^{(t)}k}}$ between $M^{(t)}(l)$ and $ROI^{(t)}(l)$ in 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 it is explained in section III.D. $\xi$ is the subset of pixels 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 TC the colour threshold employed in the pyramid generation. $f(m^{(t)}(i,j,l_w^{(t)}), a(t))$ 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 $p^{(t)}(i + d(t)k(i), j + d_k^{(t)}(j), l^{(t)}w).a(t)$ denotes the parameter vector of the transformation, which is specific for the current frame. (6) is satisfied when a match occurs.

Basically, (5) and (6) means that if there is a match between a pixel of the template and a pixel of the ROI the overlap is incremented in a value equal to the weight of the pixel of the template. We consider that the target has been found in a position if the overlap in that position is higher than 70%. All the ROI pixels that match with pixels of the template are marked as pixels of the target in the whole structure $ROI^{(t)}$. Thus, the hierarchical representation of the target $T^{(t)}$ is obtained.

### C. Target Refinement

In order to refine the target appearance, its hierarchical representation is rearranged level by level following a top-down scheme. This process is applied to all valid nodes of the ROI which have not been marked as pixels of the target in the template matching process. Each of these nodes $p^{(t)}(i,j,l) \notin T^{(t)}(l)$ searches for all valid neighbor nodes $p^{(t)}n(i,j,l)$ in a $3 \times 3$ vicinity which satisfy the following conditions:

- $Hom(p_n^{(t)}(i,j,l)) = 1$
- $p_n^{(t)}(i,j,l) \in T^{(t)}(l)$

- $g(p^{(t)}(i,j,l), p_n^{(t)}(i,j,l)) < T_C$

That is, each valid node of the ROI that does not belong to the target searches for all valid neighbor nodes, in a 3 × 3 vicinity, that belong to the target and have a colour similar to it. Among the set of candidates the studied node is linked to the most similar to it.

### D. Updating Template

As sequences present severe viewpoint changes, the object template must be updated constantly to follow up varying appearances. In this type of situations, the most current template values tend to reflect the state of the process better than the rest of the template values. However, a excessively fast updating is sensitive to sudden tracking error. Therefore, the updated template should be a compromise between the latest template and the new data. In our case, we associate a probability value with each valid node of the template model. This value places more importance to more recent data and it permits to forget older data 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 probability value or weight associates to each valid node $m^{(t)}(i,j,l)$ of the template $M^{(t)}$ in the current frame $t$.

The whole template is updated at each sequence frame:

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

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

where the superscript $(t)$ denotes the current frame and the forgetting constant, $\alpha$, is a predefined coefficient that belongs to the interval [0, 1]. This constant dictates the degree of forgetting, i.e., how strong the forgetting action will be. Equation 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 via the inverse coordinate transformation $f^{-1}(q^{(t)}(i,j,l), a(t))$, explained in section III.B, if there is match between template and target. Equation 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 $\alpha$ 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 (6) is satisfied.

Fig. 3 presents an example of weighted template updating. In order to show the older data forgetting, the intensity value of the template has been multiplied for its associated weight.

### E. Updating Region of Interest

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**

it consists in obtaining the level 0 of the new region of interest taking into account the position where the target is placed in the original image of the frame $t$. Thus, firstly, the algorithm calculates the bounding-box of $T^t(0)$. This bounding-box can be defined as the smallest rectangle which includes the target in the original frame $t$:

$$BB(T^{(t)}(0)) = [(i_{min}, j_{min}), (i_{max}, j_{max})] \quad (9)$$

with

$$\begin{array}{c} i_{min} min\{i \ / p^{(t)}(i,j,0) \in T^{(t)}(0)\} \\ \mathrm{j}_{min} min\{j \ / p^{(t)}(i,j,0) \in T^{(t)}(0)\} \\ \mathrm{i}_{max} max\{i \ / p^{(t)}(i,j,0) \in T^{(t)}(0)\} \\ \mathrm{j}_{max} max\{j \ / p^{(t)}(i,j,0) \in T^{(t)}(0)\} \end{array}$$

Then, $ROI^{t+1}(0)$ will be made up of the pixels of the next frame $p^{(t+1)}(i,j,l)$ which are included in the bounding box (9) plus the pixels included in an extra border $\epsilon$ of the bounding box. This extra border assures that the target in the next frame will be placed in the new ROI.

$$ROI^{(t+1)}(0) = \bigcup_{ij} p^{(t+1)}(i,j,0) \quad (10)$$

with

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

This step is performed at the end of the tracking process $t$.

**2) Over-segmentation of** $ROI^{(t+1)}(0)$

it consists in building the hierarchical structure $ROI^{(t+1)}$. This step is performed at the beginning of the tracking process $t+1$ and it has been previously explained in section III.A.

## IV. Experimental Results

We have tested the algorithm for several movie clips. Here, we just present some representative results. The targets were usually human faces, hands and bodies, although we also show a sequence where a ball is tracked. As we commented in section 3, templates were always initialized manually at the beginning of the tracking process.

In the first example, we applied the proposed tracking system for real-time face tracking. Fig. 4 shows the capability of the tracker to handle scale changes, rotations of the face, partial occlusions and changes of illumination. The algorithm runs at 30 fps on a 850 MHz PC.

In Fig. 5, 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. Anyway, it can be observed that the ball is reliably tracked over the whole sequence.

## V. Conclusion

The target representation and localization is a central component in visual object tracking. In this paper, a new approach for target representation and localization using template matching is proposed. This approach permits to track non-rigid objects without a previous learning of different objects views. To do that, the proposed method uses a weighted template which follows up the viewpoint and appearance changes of the object to track. This weighted template and the way it is

updated allow the algorithm can handle successfully partial 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 permits to run the whole system in real time.

## ACKNOWLEDGMENTS

This work was partially supported by the Spanish Ministerio de Ciencia y Tecnología (MCYT) and FEDER funds, project No. T1C2001–1758.

## REFERENCES

[1] D. Comaniciu, V. Ramesh and P. Meer, Kernel-based object tracking, IEEE Trans. on Pattern Anal. and Machine Intell., 25(5), pp. 564-577, 2003.
[2] T.F. Cootes, G.V. Wheeler, K.M. Walker and C.J. Taylor, View-based active appearance models, Image and Vision Computing, 20 (9-10), pp. 657-664, 2002.
[3] A. Diplaros, Th. Gevers and I. Patras, Color-shape context for object recognition, IEEE Workshop on Color and Photometric Methods in Computer Vision, 2003.
[4] A. Elgammal, R. Duraiswami, D. Harwood and L.S. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proc. of the IEEE, 90 (7). pp. 1151-1163, 2002.
[5] Th. Gevers and A.W.M. Smeulders, Image indexing using composite color and shape invariant features, Int. Conf. on Computer Vision, pp. 234-238, 1998.
[6] R. Marfil, J.A. Rodríguez, A. Bandera and F. Sandoval, Bounded irregular pyramid: a new structure for color image segmentation, Pattern Recognition, 37(3), pp. 623-626, 2004.
[7] J. Martin, V. Devin and J. Crowley, Active hand tracking, Proc. 3rd IEEE Int. Conf. Automatic Face and Gesture Recognition, pp. 573-578, 1998.
[8] H.T. Nguyen, M. Worring and R. van den Boomgaard, Occlusion robust adaptive template tracking, Proc. IEEE Conf. on Computer Vision (ICCV'01), 1, pp. 678-683, 2001.
[9] H.T. Nguyen and A.W.M. Smeulders, Template tracking using color invariant pixel features, Proc. of the Int. Conf. on Image Processing (ICIP'02), pp. 569-573, 2002.
[10] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth and L. Van Cool, Color-based object tracking in multi-camera environments, Proc. of the 25th Pattern Recognition Symposium (DACM'03), pp. 591-599, 2003.
[11] Y. Raja, S.J. Mckenna and S. Gong, Tracking color objects using adaptive mixture models, Image Vision Computing, 17, pp. 225-231, 1999.
[12] D.W. Scott, Multivariate density estimation, Wiley, 1992.
[13] H. Tao, H. Sawhney and R. Kumar, Dynamic layer representation with applications to tracking, Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00), 2, pp. 134-141, 2000.