

# Real-Time Tracking Using Multiple Target Models

Manuel J. Lucena<sup>1</sup>, José M. Fuertes<sup>1</sup>, and Nicolás Pérez de la Blanca<sup>2</sup>

<sup>1</sup> Departamento de Informatica, Escuela Politecnica Superior, Universidad de Jaen  
Campus de las Lagunillas, 23071 Jaen, Spain

{mlucena, jmf}@ujaen.es

<sup>2</sup> Departamento de Ciencias de la Computacion e Inteligencia Artificial  
ETSII. Universidad de Granada

C/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain

nicolas@ugr.es

**Abstract.** Using Comaniciu et al.'s approach as a basis, [9], this paper presents a real-time tracking technique in which a multiple target model is used. The use of a multiple model shall enable us to provide the tracking scheme with a greater robustness for tracking tasks on sequences in which there are changes in the lighting of the tracked object. In order to do so, a selection function is defined for the model to be used in the search process of the object in each frame.

## 1 Introduction

Tracking objects through the frames of an image sequence is a critical task in online and offline image-based applications such as surveillance, visual serving, gestural human-machine interfaces, video editing and compression, augmented reality and visual effects, motion capture, driver assistance, medical and meteorological imaging, etc.

Bayesian framework methods have played an important role in tracking [1][2][3]. The inclusion of a prior offline learning phase enables objects with more complicated shapes to be tracked [4][5][6]. Exemplar-based methods generate object representations from examples and then use distance measures to perform template matching.

If the objects to be tracked are non-rigid, it is advisable to represent them with probability distributions. A straightforward way to derive a distribution model is by using histogram analysis [7][8][9]. The techniques introduced independently by Bradski and Comaniciu et al. are based on the following principle: the current frame is searched for a region, a fixed-shape variable-size window, whose color content best matches a reference color model. The search is deterministic. Starting from the final location in the previous frame, it proceeds iteratively at each frame so as to minimize a distance measure to the reference color histogram. Objects are modeled using color distributions and the similarity is then measured between the target and candidate distributions using a Bhattacharyya coefficient.

A key component of a successful tracking system is the ability to search efficiently for the target, as real-time tracking is one of the main goals of our research.

Comaniciu et al.[9] propose a tracking algorithm in which a scheme for object representation and tracking is established from the definition of a single target model. The reference target model is represented by its pdf  $q$  in the feature space. The reference model can be chosen to be the color pdf of the target. In the subsequent frame, a *target candidate* is defined at location  $\mathbf{y}$ , and is characterized by the pdf  $p(\mathbf{y})$ . Both pdfs are estimated from the data. In order to satisfy the low-computational cost imposed by real-time processing discrete densities,  $m$ -bin histograms should be used.

In certain cases, when the target moves in variable lighting conditions, shadows appear which significantly alter the color distributions in the image sequence (Figure 1). A single pdf will therefore be insufficient for modeling and tracking the object reliably. Our approach is based on the use of multiple pdfs in a single target model, when lighting conditions change between frames.

This paper is organized into four sections: Section 2 presents a short review of the multiple model tracking technique; Section 3 presents some experimental results; and finally, Section 4 concludes the paper.



**Fig. 1.** Three frames of a sequence where the target presents different illumination conditions.

## 2 Tracking

### 2.1 Target Representation

In this section, we shall briefly present the main elements defined by Comaniciu et al. [9] in their tracking scheme. The pdfs defined for the target model and the target candidate will be given by  $m$ -bin histograms.

$$\begin{aligned}
 \text{target model:} \quad & \hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1\dots m} & \sum_{u=1}^m \hat{q}_u &= 1 \\
 \text{target candidate:} \quad & \hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\dots m} & \sum_{u=1}^m \hat{p}_u &= 1
 \end{aligned}$$

A target is represented by an ellipsoidal region in the image. All targets are first normalized to a unit circle.

The function  $b : R^2 \rightarrow \{1 \dots m\}$  associates to the pixel at location  $\mathbf{x}_i^*$  the index  $b(\mathbf{x}_i^*)$  of its bin in the quantized feature space. The probability of the feature  $u = 1 \dots m$  in the target model is then computed as:

$$\hat{q}_u = C \sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2) \delta[b(\mathbf{x}_i^*) - u] \quad (1)$$

where  $\delta$  is the Kronecker delta function and  $C$  is a normalization constant.

Let  $\{\mathbf{x}_i\}_{i=1 \dots n_h}$  be the *normalized* pixel locations of the target candidate, centered at  $\mathbf{y}$  in the current frame. Using the same kernel profile  $k(x)$ , but with bandwidth  $h$ , the probability of the feature  $u = 1 \dots m$  in the target candidate is given by:

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right) \delta[b(\mathbf{x}_i^*) - u] \quad (2)$$

where  $C_h$  is a normalization constant.

## 2.2 Minimization Algorithm

The similarity function defines a distance between the target model and the candidates. The distance between two discrete distributions is defined as:

$$d(\mathbf{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]} \quad (3)$$

where the similarity function will be denoted by:

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}) \hat{q}_u} \quad (4)$$

which is the sample estimate of the Bhattacharyya coefficient between  $\mathbf{p}$  and  $\mathbf{q}$  [?].

In order to find the location corresponding to the target in the current frame, the distance (3) should be minimized as a function of  $\mathbf{y}$ . This is equivalent to maximizing the Bhattacharyya coefficient  $\hat{\rho}(\mathbf{y})$ . For this, Comaniciu et al. [9] use the mean-shift algorithm with a monotone kernel.

## 2.3 Model Selection

In order to prevent losses of the target due to lighting changes, we propose a multiple model  $\mathbf{M}$ , comprising a set of  $n$  pdfs, corresponding to several different histograms of the object under typical lighting conditions:

$$\mathbf{M} = \{\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1, \dots, \hat{\mathbf{q}}_{n-1}\} \quad (5)$$

Running the target localization algorithm for each  $\hat{\mathbf{q}}_i$ , we obtain a set  $\mathbf{B}$  of Bhattacharyya coefficients,

$$\mathbf{B} = \{b_0, b_1, \dots, b_{n-1}\}$$

and a set  $\mathbf{Y}$  of image positions

$$\mathbf{Y} = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{n-1}\}$$

representing the best target location for each model and the corresponding similarity levels.

We then need to select the pdf in  $\mathbf{M}$  which best fits the observed frame. Selecting the one with the largest Bhattacharyya coefficient may increase the risk of distractions with image regions having similar histograms to the ones present in our model. In order to avoid this, we shall also take into account the position of the maximum given for each  $\hat{\mathbf{q}}_i$ , and define a probability distribution based on the difference between the position  $\mathbf{y}_i$  estimated by the tracker, and the predicted position  $\bar{\mathbf{y}}$  of the target. A value of  $\bar{\mathbf{y}}$  for each frame can be obtained by using a dynamical model of the object to be tracked.

Assuming statistical independence between  $\mathbf{B}$  and  $\mathbf{Y}$ , we can define the probability of each  $\hat{\mathbf{q}}_i$ , given  $\mathbf{B}$  as:

$$p(\hat{\mathbf{q}}_i/\mathbf{B}) = \frac{b_i \cdot p(\hat{\mathbf{q}}_i)}{\sum_j (b_j \cdot p(\hat{\mathbf{q}}_j))} \quad (6)$$

with  $p(\hat{\mathbf{q}}_i)$  being the *a priori* probability distribution for each pdf in  $\mathbf{M}$ . Additionally, the probability of each  $\hat{\mathbf{q}}_i$ , given  $\mathbf{Y}$ , is given by:

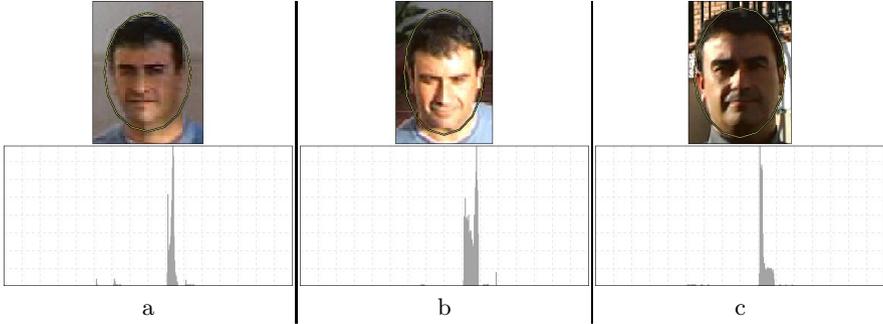
$$p(\hat{\mathbf{q}}_i/\mathbf{Y}) = \frac{p(\bar{\mathbf{y}} - \mathbf{y}_i) \cdot p(\hat{\mathbf{q}}_i)}{\sum_j (p(\bar{\mathbf{y}} - \mathbf{y}_j) \cdot p(\hat{\mathbf{q}}_j))} \quad (7)$$

In our case, we suppose that the  $\bar{\mathbf{y}} - \mathbf{y}_i$  values follow a zero-mean Gaussian distribution, i.e.  $p(\bar{\mathbf{y}} - \mathbf{y}_i) \sim N(0, \sigma)$ . Expressions (6) and (7) lead us to the probability distribution used to select the best pdf for each frame:

$$\begin{aligned} p(\hat{\mathbf{q}}_i/\mathbf{B}, \mathbf{Y}) &= \frac{p(\hat{\mathbf{q}}_i/\mathbf{B}) \cdot p(\hat{\mathbf{q}}_i/\mathbf{Y})}{p(\hat{\mathbf{q}}_i)} \\ &= \frac{b_i \cdot p(\bar{\mathbf{y}} - \mathbf{y}_i) \cdot p(\hat{\mathbf{q}}_i)}{\sum_j (b_j \cdot p(\hat{\mathbf{q}}_j)) \cdot \sum_j (p(\bar{\mathbf{y}} - \mathbf{y}_j) \cdot p(\hat{\mathbf{q}}_j))} \end{aligned} \quad (8)$$

### 3 Results

We have tested the efficiency of our method based on multiple target models by comparing it with a mean-shift tracker using single models [9] with different sequences and lighting conditions. We have used a three-component multiple model containing the simple models shown in Figure 2, and compared the obtained results. All of the experiments have been carried out on a desktop PC (Pentium IV at 2 GHz), at real-time speed (over 40 fps).



**Fig. 2.** Regions used to calculate the RGB models, with their correspondent histograms. The corresponding images belong to different sequences than the ones used for the experiments.



**Fig. 3.** Test Sequence 1, tracking with RGB histograms. a), b) and c): simple models shown in Figure 2; d): multiple model (on the upper-left corner of the images, the best model for that frame is shown).

In this paper, we show the application of the mean-shift tracker both for a simple and a multiple model, on three different sequences. In order to compute the  $m$ -bins histograms required for the tracking algorithm, two color spaces have been used: RGB quantized into  $8 \times 8 \times 8$  bins, and YUV quantized into  $16 \times 4 \times 4$ , obtaining histograms with the same number of bins, but which are more sensitive to intensity in the second case.



**Fig. 4.** Test Sequence 2, tracking with RGB histograms. a), b) and c): simple models shown in Figure 2; d): multiple model (on the upper-left corner of the images, the best model for that frame is shown).

The sequences represent a person who is moving in different directions, moving closer and farther away (scale changes) and varying the speed of the movement. As a result of the presence of shadows in two of the sequences, there are changes in the lighting of the target on entering or leaving these (see Figure 1).

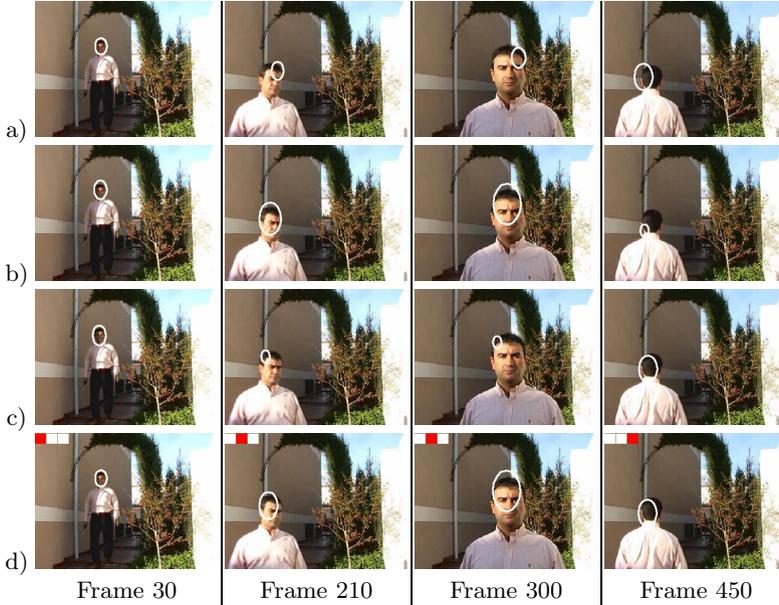
The tracking carried out in the sequences is defined on an ellipsoidal region covering the face. The model was obtained using different images, representing the target in different lighting conditions. Consequently, the target models used for the experiments do not belong to the test sequences. This is an advantage since the multiple model may be initialized offline by employing the set of images that best match the illumination conditions of the sequence.

In Figure 2, the images used to calculate the models are shown, together with their corresponding histograms, weighted with an Epanechnikov kernel of the type used in [9].

In order to predict the position  $\bar{\mathbf{y}}^{t+1}$  of the target in the next frame, a very simple dynamics has been used:

$$\begin{aligned} \mathbf{d}^{t+1} &= \lambda \cdot (\mathbf{y}^t - \mathbf{y}^{t-1}) + (1 - \lambda) \cdot \mathbf{d}^t \\ \bar{\mathbf{y}}^{t+1} &= \mathbf{y}^t + \mathbf{d}^{t+1} \end{aligned} \tag{9}$$

where  $\mathbf{y}^t$  represents the position of the target obtained by the tracking algorithm at time  $t$ , and  $\mathbf{d}^0 = 0$ . In our experiments, we have used a value for  $\lambda$  of 0.5.



**Fig. 5.** Test Sequence 3, tracking with RGB histograms. a), b) and c): simple models shown in Figure 2; d): multiple model (on the upper-left corner of the images, the best model for that frame is shown).

Due to the simplicity of the dynamical model, we have used a  $\sigma$  value in Equation (7) of 0.5. Having a more precise and less general dynamics would allow this value to be reduced, favoring measurements closer to the expected position of the target.

Although there are no significant lighting variations in the first sequence (Figure 3), the multiple model performs better when the hand and the ball occlude the target, because the dynamics gives less weight to these distracting events. The results obtained with RGB and YUV-based models are very similar.

In the second and third sequence, significant variations can be observed in the lighting conditions of the target. The simple model obtained from Image C (Figure 2) gets distracted at the beginning of the second sequence (Figure 4) because of the similarity between the histograms of the head and the ground. For the last sequence, we can see that the selection of the best model for each frame increases the tracker accuracy (Figure 5).

## 4 Conclusion

The method presented in this paper enables multiple models to be used in order to prevent loss when there are significant variations in the target's histogram, and allows real-time execution on a desktop computer.

The experimental results indicate that our method increases the robustness of tracking when faced with lighting changes in the object. By adequately selecting the samples for the multiple model, it is possible to track an object from its generic set of images.

## Acknowledgment

This work has been financed by grant TIC-2001-3316 from the Spanish Ministry of Science and Technology.

## References

1. M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proceedings of European Conference on Computer Vision*, Cambridge, UK, 1996, pp. 343–356.
2. Hedvig Sidenbladh and Michael J. Black, "Learning image statistics for bayesian tracking," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Vancouver, Canada, 2001, vol. 2, pp. 709–716.
3. Y. Wu and T.S. Huang, "," in *Proceedings of Eighth IEEE International Conference on Computer Vision*, 2001, vol. 2, pp. 26–33.
4. S. Avidan, "Support vector tracking," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2001, pp. 184–191.
5. D. Gavrila and V. Philomin, "Real-time object detection for smart vehicles," in *Proceedings of IEEE International Conference on Computer Vision*, 1999, pp. 87–93.
6. K. Toyama and A. Blake, "Probabilistic tracking in a metric space," in *Proceedings of IEEE International Conference on Computer Vision*, 2001, vol. 2, pp. 50–57.
7. G.R. Bradski, *Computer Vision Face Tracking for Use in a Perceptual User Interface*, Intel Technology Journal, 1998.
8. Tyng-Luh Liu and Hwann-Tzong Chen, "Real-time tracking using trust-region methods.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 397–402, 2004.
9. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.