FACE TRACKING USING A REGION-BASED MEAN-SHIFT ALGORITHM WITH ADAPTIVE OBJECT AND BACKGROUND MODELS

Veronica Vilaplana, David Varas

Technical University of Catalonia (UPC), Barcelona, Spain
veronica.vilaplana@upc.edu

ABSTRACT

This paper proposes a technique for face tracking based on the mean shift algorithm and the segmentation of the images into regions homogeneous in color. Object and background are explicitly modeled and updated through the tracking process. Color and shape information are used to define with precision the face contours, providing a mechanism to adapt the tracker to variations in object scale and to illumination and background changes.

1. INTRODUCTION

Face tracking plays an important role in many applications such as video indexing, visual surveillance, human computer interaction or facial expression recognition [1]. In these applications it is necessary to detect the faces, track them from frame to frame and analyze the tracks, for instance to understand their behavior. In the simplest form, a tracker estimates the face trajectory by locating its position in every frame of the sequence. While this information may be sufficient for some applications (e.g. detecting the presence of an intruder), other applications require additional data, like knowing the orientation, extension or even the precise contour of the faces at every frame (e.g. facial expression recognition).

In this paper we deal with this last kind of problem, that is, the tracking and segmentation of faces along video sequences.

We propose a strategy that combines the mean shift algorithm for tracking with a representation of the images in terms of regions homogeneous in color. As will be discussed in the following sections, mean shift can be used as a robust and flexible algorithm for tracking, requiring minimal training and computational resources. In turn, the use of regions permits a robust estimation of object and background models, as well as the precise definition of the face shape.

The proposed algorithm improves the technique presented in [2], by introducing an explicit model for the background which allows a Bayesian estimation of object probabilities and a more accurate definition of the object contours. Object and background models are continuously updated through the process, handling variations in object scale and pose, as well as illumination and background changes. The resulting algorithm tracks robustly in challenging sequences where the previous algorithm failed.

For a complete review on different techniques for object tracking the reader is referred to [3]. The organization of the paper is the following. In Section 2 we review the basic mean shift algorithm and its use for object tracking. In Section 3 we detail our region-based approach. Section 4 presents some results of the technique and a comparison with the results obtained with the CAMSHIFT algorithm [4]. Finally, Section 5 closes the paper with some conclusions.

This work has been partly supported by the projects CENIT-2006 HESPERIA and TEC2007-66858/TCM PROVEC of the Spanish Government.

2. THE MEAN SHIFT PROCEDURE

2.1. Basic formulation

Mean shift is an iterative, non parametric procedure for seeking the mode of a density distribution represented by a set of samples [5, 6].

Let S be a finite set in an n-dimensional Euclidean space X, the sample data. The sample mean with kernel K at a point $x \in X$ is defined as

$$m(x) = \frac{\sum_{s \in S} K(s-x)w(s)s}{\sum_{s \in S} K(s-x)w(s)} \quad (1)$$

where K defines an influence zone for x and $w(x)$ is a weight function. The difference $m(x) - x$ is called *mean shift*. The idea is to compute the sample means for a reduced set of points $T \subset X$ and move the points in T towards their mean, until convergence. That is, if $m(T) = \{m(t) : t \in T\}$, the mean shift procedure iterates and evolves T until it finds a fixed point $T = m(T)$.

Formally, a kernel K is a function defined in terms of its profile function $k : [0, \infty] \rightarrow R$, a non negative, non increasing and integrable function such that $K(x) = k(||x||^2)$.

The mean shift algorithm seeks the modes of the density estimate $q(x)$ computed with another kernel H which is called the shadow kernel of K:

$$q(x) = \sum_{s \in S} H(s-x)w(s) \quad (2)$$

The two kernels must satisfy the relationship $h'(r) = -ck(r)$, where h and k are the profiles of H and K, respectively, $r = ||s-x||$ and $c > 0$ is some constant. This relationship guarantees that the mean shift vector $m(x) - x$ is in the gradient direction of the density estimate $q(x)$, and has an adaptive step size, that is, it moves fast when it is far from the mode and in short steps when it is near the mode. Two kernels K typically used are the unit flat kernel and the unit Gaussian kernel, whose shadows are the Epanechnikov and Gaussian kernels, respectively [6].

2.2. Mean shift for tracking

In object tracking the goal is to track the location of an object at each frame in the sequence. The evolving set T, therefore, consists of just one point, the object centroid. In this context, a sample corresponds to the spatial coordinates of a pixel x, and has an associated sample weight $w(x)$, which defines how likely it is that the pixel x belongs to the object. The mean shift algorithm seeks the mode of the kernel density $q(x)$ computed with these weights.

Typically, weights are determined using a color-based object appearance model. For instance, [4] works with a histogram of object colors and histogram backprojection is used to assign to each pixel the likelihood associated with its color, while [7] computes the weights with a measure of histogram similarity between object and background color distributions. However, the mean shift strategy can be applied to sample weight images computed with other features besides color, like texture similarity, background substraction results, etc. [8, 9].

A particular implementation of the algorithm requires the definition of the kernel (scale and shape), a model for the object, the weight function and the shape and extension of the final tracked object.

The kernel scale is a critical parameter to the performance of the algorithm [8]. If the scale is too large, the search window may contain background points that resemble the object model, leading to an overestimation of the object size. A too large window may even make the tracker converge to an area between multiple modes instead of converging to one of them. On the other side, if the scale is too small, the shifts may move within a flat zone of likelihood around the mode, leading to poor object localization.

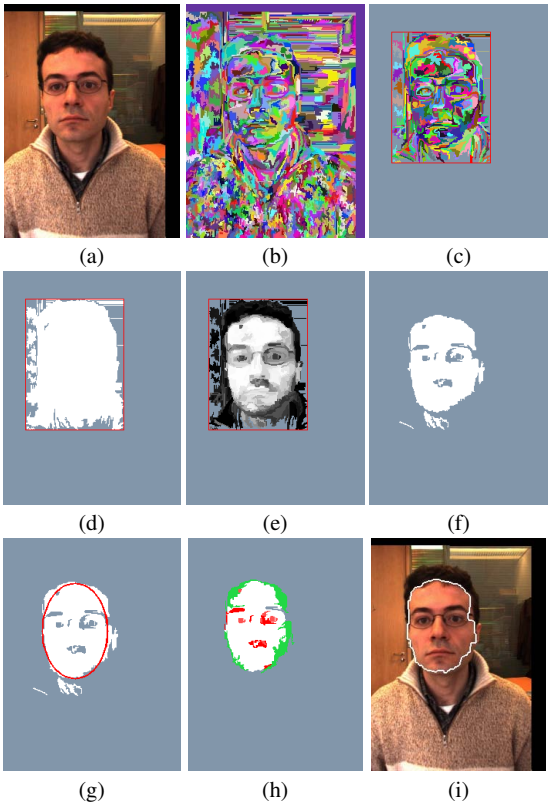

Fig. 1. (a) Original image, (b) Partition, (c) Fitted partition, (d) Kernel (fitted partition mask), (e) Weight image, (f) Initial object mask, (g) Fitted ellipse, (h) Final object mask, (i) Smoothed object contour

3. A REGION-BASED APPROACH

The proposed approach to face tracking is an extension of the basic mean shift tracking algorithm that relies on the use of a color-homogeneous image partition. Objects are represented by sets of regions, which are used to build explicit models of object and background. These models permit the Bayesian estimation of pixel probabilities, which are used first to define the weight images during the tracking iterations and at the end, together with a face shape model, to segment the final face. The precise definition of face contours provides a mechanism for adapting the kernel size while tracking faces through changes in scale, and for updating the object and background models.

3.1. Search window and kernel

The algorithm works with pixels that lie within a subimage defined by a rectangular search window $\mathcal{W}$ and a partition $\mathcal{P}$ of the image into regions homogeneous in color.

At each frame, the width and height of the search window are the width and height of the bounding box of the object found in the previous frame, scaled by a fixed factor (which is constant through the process). The window size is the same for all iterations within a frame.

The kernel is defined by all the regions R in partition $\mathcal{P}$ that are completely included in $\mathcal{W}$:

$$K(x) = \begin{cases} 1 & \text{if } x \in \{R \in \mathcal{P}/R \subset \mathcal{W}\} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Note that the kernel scale changes according to the size of the tracked object and its shape takes into account the color homogeneity observed in the image since it is defined by the regions in the partition. The fitting is performed at each iteration of the mean shift process. An example of image partition, fitted partition and kernel is presented in figures 1(b),(c) and (d) respectively, where the factor used to scale the previous object size is set to 1.4, and the image partition has 500 regions.

3.2. Object and background color models

The object color is modeled as a class conditional color distribution computed with a histogram in the YCbCr color space. Therefore, given a pixel x with color $I(x)$, the likelihood of the pixel given that it belongs to the object is $p(I(x)/\mathcal{O}) = h_{\mathcal{O}}(I(x))$, where $h_{\mathcal{O}}$ is the object histogram. This histogram is first learned from the object segmented in the first frame of the sequence which, in this case, is obtained with the method presented in [10].

The background color model is $p(I(x)/\mathcal{B}) = h_{\mathcal{B}}(I(x))$, where $h_{\mathcal{B}}$ is the background histogram. This model is first learned using a neighborhood around the object $\mathcal{O}$ segmented in the first frame.

Object and background color models are updated at every frame, combining the previous model with models derived from the object segmented in the current frame $\mathcal{O}_c$:

$$h_{\mathcal{O}_t} = \alpha h_{\mathcal{O}_{t-1}} + (1-\alpha) h_{\mathcal{O}_c} \quad (4)$$

where $\alpha \in [0, 1]$ is the learning rate. A similar expression is used to update the background model.

3.3. Weight function

During the tracking, the object and background models obtained at frame $t-1$ are used to segment the object at frame t. The object

probability of a pixel x with color $I(x)$ at frame t can be computed using the Bayes' formula:

$$p(\mathcal{O}_t/I(x)) = \frac{p(I(x)/\mathcal{O}_{t-1})p(\mathcal{O}_{t-1})}{p(I(x))} \quad (5)$$

where

$$p(I(x)) = p(I(x)/\mathcal{O}_{t-1})p(\mathcal{O}_{t-1}) + p(I(x)/\mathcal{B}_{t-1})p(\mathcal{B}_{t-1}) \quad (6)$$

$p(I(x)/\mathcal{O}_{t-1})$ is the object model and $p(\mathcal{O}_{t-1})$ is the object probability (at $t-1$). In turn, $p(\mathcal{B}_{t-1})$ is the background probability and $p(I(x)/\mathcal{B}_{t-1})$ the background color model.

These probabilities are used (i) to define the weight image and also (ii) to classify each region into object or background and define the shape of the tracked object in the current frame (see 3.4).

The weight image is computed in a region-based manner as opposed to the typical pixel-based approach. Each region R_i in the fitted partition is assigned a weight value (see figure 1.e), which is the average:

$$w(R_i) = \frac{1}{|R_i|} \sum_{x \in R_i} p(\mathcal{O}_t/I(x)) \quad (7)$$

3.4. The final shape

The final object shape is obtained in three steps. First, pixels are classified into object or background; we choose for each pixel the class with highest probability. Then a region R_i within the fitted partition is said to be an object region if the majority of its pixels are classified as object pixels, that is, if

$$\sum_{x \in R_i} \mathcal{I}(c(I(x)) = \mathcal{O}) > \frac{1}{2}|R_i| \quad (8)$$

where $c(I(x))$ is the pixel class (see figure 1.f).

The initial mask may contain background areas which present colors similar to object colors and may also lack some object areas due, for instance, to an illumination change. At this point we introduce information about the object shape to correct these problems. We assume that the face area is one connected component without holes and that its shape is approximately elliptical. The best-fit ellipse is then estimated using second order moments on the weight image (see figure 1.g). Note that to track non elliptical objects, shape matching with distance transforms can be used to fit an object shape model to the initial mask.

Finally, information from the fitted ellipse and the initial object mask are combined. The final object shape is formed by high probability regions that touch the filled fitted ellipse and regions that correspond to holes in the initial mask (see figure 1.h). Figure 1.i shows the final smoothed face contour superimposed on the original image.

3.5. Algorithm summary

The algorithm proceeds according to the following main steps:

1: Compute the object and background models using the object segmented in the first frame
2: Estimate the initial position x_0 and initial size of the search window; define a subimage where the search window will move
3: Partition the subimage into small regions homogeneous in color. Define the kernel by fitting the partition to the search window.
4: Compute object and background probabilities and the weight image.
5: Estimate the new position $x_{i+1} = m(x_i)$
6: Repeat steps 4-5 until convergence
7: Define the output, that is, the extent and shape of the tracked object by classifying regions in object or background
8: Update object and background models
9: Return to step 2 for the next frame

4. RESULTS

To check the effectiveness of the proposed method, we have tested it on a wide variety of challenging image sequences. Results comparing the performance of our system with that of CAMSHIFT (from OpenCV library) are available at our website http:// gps-tsc.upc.es / imatge / _Veronica / RBMS-ModelUpdate.html and some sample frames are presented in figure 2. The first three sequences shown here can be obtained from http:// research.microsoft.com / vision / cambridge / i2i/ DSWeb.htm

In all the examples, object and background histograms are computed in the YCbCr color space, with 32 bins per component, and the same parameters (scale = 1.4, $\alpha = 0.5$) are used.

The first sequence "Jamie1r" presents a fast moving face with large variations in scale. As the face approaches the camera, the region-based approach correctly tracks and segments the face, and when it moves backwards it recovers the correct shape in a few frames. The CAMSHIFT algorithm fails when the face moves across part of the background with a similar color (the door), and a large region of the background is mistaken for the object.

The second example, the "Ms" sequence, shows variations in face pose and an arm approaching the face. The presence of the hand is correctly managed by the proposed algorithm due to the fitting process. The CAMSHIFT tracker correctly deals with pose changes but around frame #205 the search window starts growing, including the arm and part of the background and the algorithm converges to an area between the two modes. The third and four examples show the performance of both systems for a sequence with partial occlusion and pose changes ("IUJW"), and for a sequence with illumination variations and camera motion ("Foreman").

The computational cost introduced by the use of the partition depends to some extent on the size of the object being tracked. The average number of frames processed per second is 10, for a set of 20 sequences with faces of different sizes (experiments performed on a Core 2Duo at 1.86GHz).

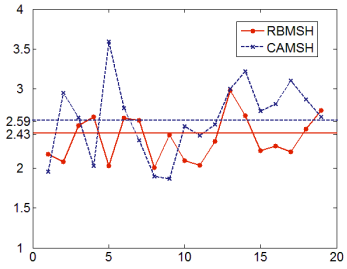

Fig. 3. Average number of iterations per frame

Figure 3 presents, for the same set of sequences, the average number of mean shift iterations per frame. The global average is 2.43 for the region-based algorithm and 2.59 for the CAMSHIFT.

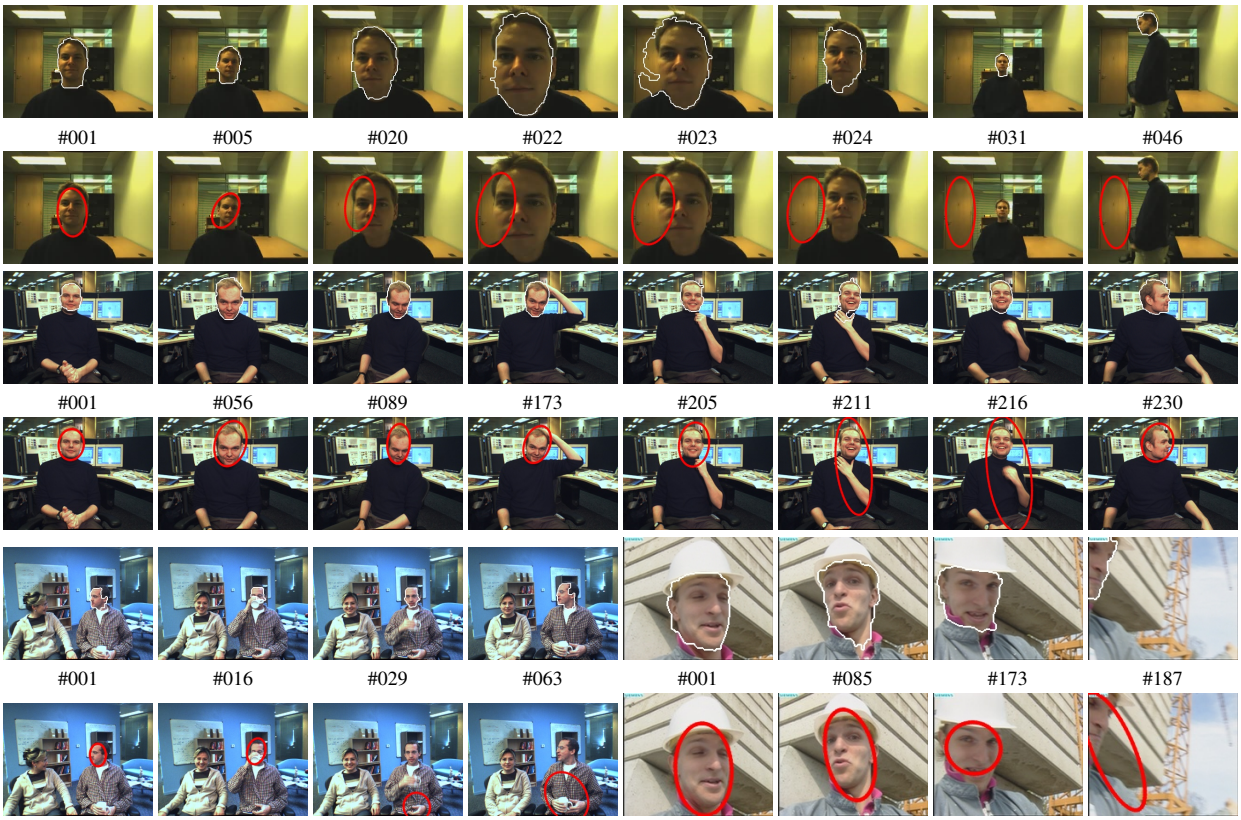

Fig. 2. Results for the sequences "Jamie1r", "MS", "IUJW" and "Foreman" with (first, third and fifth rows) the proposed algorithm and (second, fourth and sixth rows) the CAMSHIFT algorithm

5. CONCLUSIONS

We have presented an extension of the mean shift algorithm for face tracking and segmentation that relies on the use of an image partition and explicit color models for object and background, which are updated through the tracking process. The algorithm has been tested on many different sequences with very good results in difficult situations such as scale and pose variations, moving cameras and changing scenarios. Future work will examine the use of shape information to avoid leakages in the final shape contour and to adapt faster to changes in scale.

6. REFERENCES

[1] R. C. Verma, C. Schmid, and K. Mikolajczyk, "Face detection and tracking in a video by propagating detection probabilities," *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1215–1228, October 2003.

[2] V. Vilaplana and F. Marques, "Region-based mean shift tracking: application to face tracking," in *International Conference on Image Processing*, Los Angeles, USA, October 2008.

[3] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, December 2006.

[4] G. Bradsky, "Computer vision face tracking for use in a perceptual user interface," in *IEEE Workshop on Applications of Computer Vision*, Princeton, N.J., 1998, pp. 214–219.

[5] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function with applications in pattern recognition," *IEEE Trans. on Information Theory*, vol. 21, pp. 32–40, 1975.

[6] Y. Cheng, "Mean shift, mode seeking and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, August 1995.

[7] 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, May 2003.

[8] R. Collins, "Mean-shift blob tracking through scale space," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[9] R. Stolkin, I. Florescu, and G. Kamberov, "An adaptive background model for camshift tracking with a moving camera," in *Int. Conference on Advances in Pattern Recognition*, Calcutta, India, January 2007.

[10] V. Vilaplana and F. Marques, "Face detection and segmentation on a hierarchical image representation," in *European Signal Processing Conference*, Sept. 2007.