MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan

3-2

Object tracking with spatio-temporal blob

C. Achard, G. Mostafaoui, M. Milgram
Laboratoire des Instruments et Systèmes d'Ile de France
4, place Jussieu, Case 252, 75252 Paris Cedex 05
achard{maum}@ccr.jussieu.fr, ghiles.mostafaoui@lis.jussieu.fr

Abstract

We propose to develop a tracking algorithm of objects or humans, based on kinematics, with a fixed monochromatic camera, without any knowledge on the sequence: size, shape or number of objects are unknown and can evolve with time. For this purpose, we first make a motion detection, then, as we suppose that people move locally in a consistent way and thus draw a regular trajectory in the spatio-temporal space, we modelize locally the 3D moving points by a mixture of Gaussians where each Gaussian represents a trajectory. These points described only by their coordinates (x, y, t) are clustered with the Classification Expectation-Maximization (CEM) algorithm. We will shown on tracking results, that this method based only on kinematics, manages the number of objects to track, occlusions and poor segmentations (over- or under-segmentation).

1 Introduction

In this paper, we address the problem of segmenting, detecting, and tracking several targets using a monochromatic fixed camera and by considering only the kinematics. Usual approaches consist in tracking the objects time after time, using the result of the previous iteration. We can mention as example the JPDAF algorithm [1], methods based on particles filters [2] or methods based on layer representation [3]. They are robust but have several drawbacks relative to their initialization: where are the targets in the first image and how many are they? How manage an evolutionary number of objects? How deal with over- or under-segmentations?

In our approach, to deal with occlusions and bad segmentations, spatio-temporal points detected in movement are considered in a global way, during several frames. Among all these pixels, we look for those that form a coherent shape in the spatio-temporal space and group them: pixels that represent physical points describing a steady trajectory, form a regular volume in the space (the section, at a given time, represents the shape of the target). In our approach, each volume is modelized by an ellipsoid with its center and its covariance matrix and is obtained with the Classification Expectation-Maximization (CEM) algorithm [4], which is a modified version of the usual Expectation-Maximization (EM) algorithm.

Several authors have already used this EM algorithm in this context. Tao et al. [3] use the EM algorithm and a dynamic layer representation to jointly estimate object motion, ownership and appearance. In these works, an external module makes the initialization and the number of targets is fixed during the sequence.

We can also mention C. Bregler [5] who recognizes human dynamics in video sequences. To do that, the first image is segmented in blobs of coherent motion and color. This is done using the EM algorithm (image is seen as a mixture of coherent blobs). Next images are segmented using the same method expected that blob parameters are initialized using the previous frames: based on parameters of the Gaussian distribution at time *t-1* (θ_{t-1}), a Kalman filter computes the predicted mean and covariance of θ_t, which are used as priors for the new E.M. iterations. This method can be seen as propagating a multinomial distribution (mixture of Gaussians) of the system state θ_t through time. So, segmentation and tracking are treated as the same problem.

Raja et al. [6] have a similar approach: color densities of the tracked object are approximated by Gaussian mixtures that are dynamically updating with time. In [7], Heisele et al. segment the first image of the sequence in blobs based on color. For each new image, a parallel k-means clustering algorithm adapts clusters of the previous iteration.

These three last methods suppose that the number of objects is known and constant along the sequence. They track targets time after time and don't have a global approach by integrating several frames of the sequence.

In [8], Greenspan et al. extract coherent space-time regions using Gaussian mixtures to represent and index video. The feature space possesses six dimensions : color information (L, a, b), spatial information (x, y) and time (t). So, they analyze video input as a single entity as opposed to a sequence of separate frames.

We use here the same approach but only on spatio-temporal data (*x,y,t*) in order to develop a tracking algorithm based on kinematics. This last one, for which any initialization is required, will be achieved without knowledge about the shape of objects, their size or their number. Moreover, the object number can evolve during the sequence (appearance or disappearance of objects). Moreover, we wish to deal with occlusion and under or over-segmentations. The algorithm is presented in section 2 and results will be shown in section 3 before a conclusion.

2 Presentation of the method

For the tracking at the time t, the algorithm presented figure 1, works on a sliding temporal window centered on *t* and comprising *lw* frames. All images between $t - lw/2$ and $t + lw/2$ are considered but the decision is made only for the time *t*. For each image, a motion detection is achieved by subtracting the current image from a reference one. Pixels are introduced in a Markovian relaxation [9] to improve an initial coarse thresholding. A labeling of connected components is then performed to obtain regions. As we suppose that people move locally in a consistent way, points of the temporal window detected in motion form regular volumes in the spatio-temporal space. These 3D moving points are assigned to a mixture of Gaussians (each Gaussian representing a trajectory) with the Classification Expectation-Maximization (CEM) algorithm. It is a modified version of the EM algorithm, which accelerates the computing time by considering binary ownerships. To update the tracking at the time *t*, we consider all points between $t - lw/2$ and $t + lw/2$.

Among them,

- points before the time *t* have already been assigned to Gaussians in a sure way (these decisions will not be updated).
- points between *t* and *t+lw/2-1* have also been assigned to Gaussian in the previous iteration. These assignments could evolve during this step of the CEM algorithm.
- Points at *t=t+lw/2* are initially assigned to new Gaussians to allow entrance of new objects in the scene.

The convergence of the CEM algorithm, for a temporal window, leads to the number of Gaussians and their parameters for the central time of the window. The synoptic of the algorithm is presented figure 1. More details concerning these different steps will be given in the next paragraphs.

2.1 New Gaussian creation

For the time *t*, we are interesting by all 3D points between $t - lw/2$ and $t + lw/2$. Only points at $t + lw/2$ are new and can generate new trajectories. They come from the motion detection computed in the frame at $t = t + lw/2$ and have been labeling. A new Gaussian is created for each region in this image (a region contains often several points). The means $\mu_k = (x_k, y_k, t_k)$ of each new Gaussian can easily be computed while the covariance matrices Σ_k are initialized by heuristics to the identity matrix multiplied by 3. It corresponds to a spherical volume in the spatio-temporal space.

2.2 A CEM step

We detail below a step of the CEM algorithm. Data, which are R^3 valued vectors $\boldsymbol{x} = (x_i, y_i, t_i)$, $i \quad 1 \cdots N$ are supposed to be a sample of a mixture of density:

$$f(\boldsymbol{x}_i) = \sum_{k=1}^{K} p_k f(\boldsymbol{x}_i, \boldsymbol{a}_k)$$

where p_k are the mixing coefficients which should add up to 1, $f(\boldsymbol{x}_i, \boldsymbol{a}_k)$ denotes the 3-dimensionnal normal density function with unknown mean μ_k and covariance matrix Σ_k, $\boldsymbol{a}_k = (\mu_k, \Sigma_k)$ are Gaussian parameters to be estimated and *K* is the number of Gaussians.

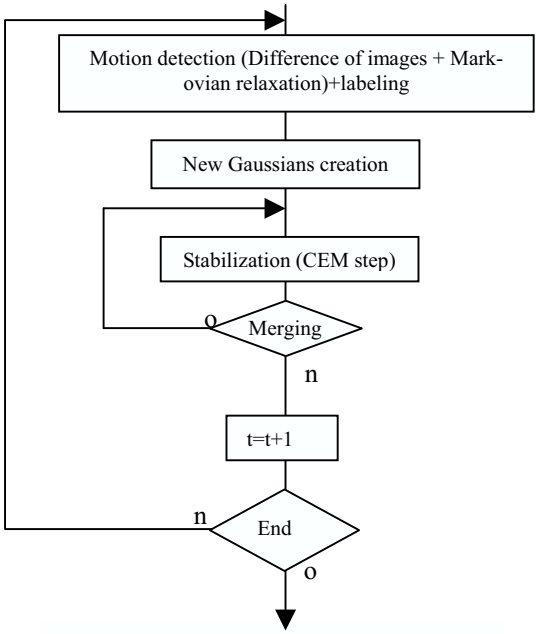


Figure 1 – Synoptic of the algorithm

The CEM algorithm computes the estimates p_k and $\boldsymbol{a}_k$ (for $k = 1 \ldots K$) and finds a partition $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K)$ of data. This algorithm, which is a classification version of the EM algorithm, incorporates a classification step between E step and M step using a *Maximum A Posteriori* (MAP) principle. It starts with an initial partition $\mathcal{P}^0$ obtained with:

- the tracking results at the previous iteration for points belonging to $[\, t - lw/2,\ t + lw/2 - 1\,]$.
- the affectation to new Gaussians for $t = t + lw/2$.

It iterates then:

E step: compute, for all points $\boldsymbol{x}_i$ such as $t_i \geq t$ and for $k=1 \ldots K$ the current posterior probabilities $h_k^m(\boldsymbol{x}_i)$ that $\boldsymbol{x}_i$ belongs to $\mathcal{P}_k$:

$$h_k^m(\boldsymbol{x}_i) = \frac{p_k^m f(\boldsymbol{x}_i, \boldsymbol{a}_k^m)}{\sum_{k'=1}^{K} p_{k'}^m f(\boldsymbol{x}_i, \boldsymbol{a}_{k'}^m)}$$

from the current parameter estimates p_k^m and $\boldsymbol{a}_k^m$.

C step: All point $\boldsymbol{x}_i$ before the time *t* preserve their affectation to Gaussian. The other points are assigned to the class which provides the maximum posterior probability $h_k^m(\boldsymbol{x}_i)$, $k=1,\ldots,K$. Let $\mathcal{P}^m$ the resulting partition.

M step: for k=1…K compute the maximum likelihood estimates (p_k^{m+1}, $\boldsymbol{a}_k^{m+1}$) using the sub-samples $\mathcal{P}_k^m$.

The computation of p_k^{m+1} is obtained by introducing higher-level information:

- The first segmentations in regions, computed in each image are employed: 3D blobs (or Gaussians) composed of points belonging to a coherent region will be more probable. A coherent region is such as its pixels belong in majority to the same blob. This concept avoids such mistakes as, for example the independent following of the bust and the legs of a person, while allowing occlusions.

- A Gaussian is more probable if its pixels are spaced out in time; a trajectory will be preferentially composed by several frames.

The mixing coefficients taking into account this *a priori* information will be:

$$p_k^{m+1} = \left(p_{k,1}^{m+1} p_{k,2}^{m+1} p_{k,3}^{m+1} \right) \Big/ \sum_{j=1}^{K} p_j^{m+1}$$

where

- $p_{k,1}^{m+1} = \dfrac{card(\mathcal{P}_k^m)}{N}$, as usually,
- $p_{k,2}^{m+1} = \dfrac{1}{NR_k} \sum_{i=1}^{NR_k} \dfrac{N_{k,i}}{N_i}$ where NR_k is the number of regions with at least one pixel in the class $\mathcal{P}_k^m$; N_i is the number of pixels of the region i and $N_{k,i}$ is the number of pixels belonging to the class $\mathcal{P}_k^m$ and to the region i.
- $p_{k,3}^{m+1} = \dfrac{l_k}{lw}$ where l_k is the number of discrete times where the class $\mathcal{P}_k$ *is present*.

The CEM algorithm is iterated while the log-likelihood varies. This latter is defined by:

$$L_K = \sum_{i=1}^{N} \ln \left\{ \sum_{k=1}^{K} p_k f(\boldsymbol{x}_i, \boldsymbol{a}_k) \right\}$$

This CEM algorithm looks like the k-means algorithm except that the latter uses Gaussian mixture with equal proportions and a common covariance matrix of the form $\sigma^2 I$.

2.3 The merging step

In order to remove false Gaussians (those that does not correspond to real trajectories and have been introduced for example by points in the last image of the temporal window), a fusion step is added. A couple of Gaussians can be merge if:

- one of the two Gaussians begins after the time t (two Gaussians beginning before the time t can not be merged because it would change results obtained before this time).
- if one of the two Gaussians starts before time t, its temporal duration must be higher than lw/4 (empirical value). Indeed, the covariance matrix of the 3D points represents their dynamic only if the points are enough.
- among all couples of Gaussians which verify the previous conditions, we look for the one that minimizes after fusion the Bayes Information Criterion (*BIC*) defined by Schwartz [10]:

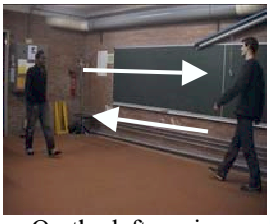
$$BIC(M) = -2L_M + Q_M \ln(N)$$

where M is the number of Gaussians, Q_M is the number of free parameters and L_M is the log-likelihood. If the fusion of this couple of Gaussians leads to: $BIC(K-1) < BIC(K)$, then, both Gaussians are merging (this last condition is not necessary if a new Gaussian includes less than three points).

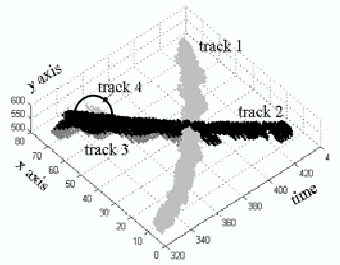
The number of Gaussians (K) is thus progressively decreased until the *BIC* criterion does not decrease any more. The number of trajectories has then been reached and the temporal window is sliding to the next time.

3 Results

Several results obtained on real sequences will now be presented: a color represents each track. Images have been sub-sampled by a factor 10, and only one image out of two has been considered, in order to reduce the computing time. Let us recall that data are composed by all the spatio-temporal points (x_i,y_i,t_i) detected in movement. For both sequences, the length of the temporal window *lw* was fixed to 50 frames.



On the left, an image of the sequence.



On the right, two views of the 3D result. All points are represented in the spatio-temporal space (x,y,t).

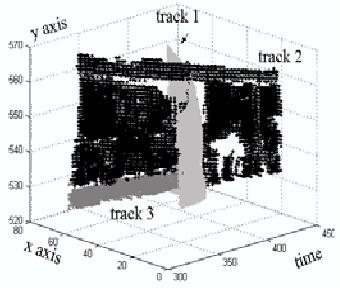


Figure 2. Result on the first sequence

The first sequence (figure 2) composed by 120 frames,

consists in the tracking of two people, who are crossing, one evolving from the left to the right, the other in the opposite direction. *17.322* points are detected in movement. Both persons are very bad segmented due for example to the pipes in the bottom of the wall (no size criterions have been introduced on region to keep the generality of the method). On average, four regions per image compose each person and lacks of detection appear often. Examples of poor motion segmentation are shown on figure 2, on the second view of the result: in the track 2, holes corresponding to lacks of detection, appear. Four tracks have been obtained. One is correct (track 1), the second has been broken in two (track 2 and 3): during the beginning of the sequence, the feet of the corresponding person are always detected as an isolated region. A track is thus created for these feet. It disappears when feet and body are detected as a same region. The only way to avoid this problem would be to introduce a model of object to track but in these conditions, the algorithm becomes dedicated to a specific application. The track 4 corresponds to noise: during few images, a shadow was detected like a moving region. To solve this problem, higher order features depending of the application have to be introduced.

It should be noted that occlusion are correctly managed on this sequence because the temporal window is large enough. Generally, this length must be at least four times higher than the length of the occlusion. If it is not the case, a new trajectory is created after the occlusion. There is a duality on the length of the temporal window: it must be large enough to cross occlusions but must remain sufficiently small to keep a steady movement.

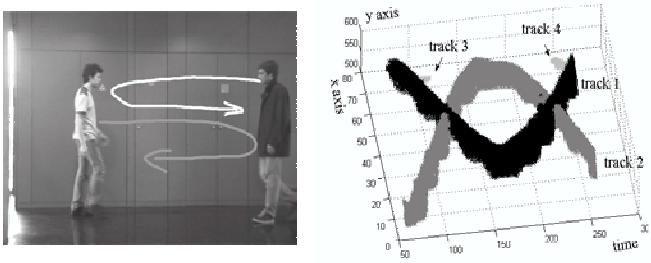


Figure 3. Result on the second sequence

The second sequence (figure 3) represents two persons who are crossing and turning back. It is composed by 226 frames and *48.899* points. The first segmentations are still very poor with over-segmentations and lacks of detection. Results show that the trajectories of the two people have been correctly determined (track 1 and 2). One can notice however that two additional tracks appear (tracks 3 and 4) which correspond to shadows detected during the segmentation. It is thus correct to not assign these points to the true trajectories. Characteristics of higher level will have to be introduced to separate the true tracks from the noise (on this sequence, a criterion of minimal length of a track would be enough to solve this problem)

4 Conclusion

An algorithm making the tracking of targets without any a priori knowledge (initialization, size, color, shape, …) is presented. It uses only kinematical information by considering that targets move in a locally steady way and thus draw regular trajectories in the spatio-temporal space. All 3D points (x_i, y_i, t_i) detected in movement during a first step are assigned to a mixture of Gaussians with the Classification Expectation Maximization (CEM) algorithm. Each Gaussian represents locally a consistent trajectory and evolves with time. As no a priori knowledge is employed, false trajectories corresponding to noise or shadows are detected; they could be removed by introducing higher-level features. In a same way, when an object is always split up in two regions during the segmentation, two tracks are created, one for each part of the object. This result is also coherent and only a model of the object to follow will be able to merge these two tracks. Results on two real sequences of moving persons who are crossing have been presented. We have shown on these sequences that this algorithm, which is totally autonomous, deals with occlusions and poor segmentations.

References

[1] Y. Bar-Shalom and T. Fortmann, "Tracking and data association", *Mathematics in Science and Engineering. Academic Press*, 1988.

[2] M. Isard, A. Blacke, "Condensation – conditional density propagation for visual tracking", *Int. J. Computer Vision*, 29, 1, 1988, pp. 5-28.

[3] H. Tao, H.S. Sawhney, R. Kumar, "Dynamic representation with application to tracking", *In Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Hilton Head, South Carolina, June 2000, pp.134-141.

[4] G. Celeux and G. Govaert, "A Classification EM algorithm for clustering and two stochastic versions", *Computational Statistics & Data Analysis*, vol. 14, 1992, pp. 315--332.

[5] C. Bregler, "Learning and recognizing human dynamics in video sequences", *In Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Puerto-Rico, 1997, pp.568-574.

[6] Y. Raja, S.J. McKenna and S. Gong, "Color model selection and adaptation in dynamic scenes", *In Proc. European Conference on Computer Vision*, Germany, 1998, pp. 460-474.

[7] B. Heisele, U. Kressel and W. Ritter, "Tracking non-rigid, moving objects based on color cluster flow", *In Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Puerto-Rico, 1997, pp.257-260.

[8] H. Greenspan, J. Goldberger, A. Mayer, "A probabilistic framework for spatio-temporal video representation and indexing", *In Proc. European Conference on Computer Vision*, Copenhagen, 2002, pp. 461-475.

[9] A.Caplier, F.Luthon and C.Dumontier, "Real-Time implementations of an MRF-Based Motion Detection Algorithm", *RealTimeImg* (4), No. 1, February 1998, pp. 41-54.

[10] G. Schwarz, "Estimating the dimension of a model", *The Annals of statistics* (6), 1978, pp. 461-4.