Fast Marching to Moving Object Location

E. Sifakis and G. Tziritas

Institute of Computer Science - FORTH,
P.O. Box 1385, Heraklion, Greece
and,
Department of Computer Science, University of Crete
P.O. Box 1470, Heraklion, Greece
E-mails: tziritas@csi.forth.gr sifakis@csd.uch.gr

Abstract. In this paper we address two important problems in motion analysis: the detection of moving objects and their localization. Statistical and level set approaches are adopted in order to formulate these problems. For the change detection problem, the inter-frame difference is modeled by a mixture of two zero-mean Laplacian distributions. At first, statistical tests using criteria with negligible error probability are used for labeling as many as possible sites as changed or unchanged. All the connected components of the labeled sites are seed regions, which give the initial level sets, for which velocity fields for label propagation are provided. We introduce a new multi-label fast marching algorithm for expanding competitive regions. The solution of the localization problem is based on the map of changed pixels previously extracted. The boundary of the moving object is determined by a level set algorithm, which is initialized by two curves evolving in converging opposite directions. The sites of curve contact determine the position of the object boundary. For illustrating the efficiency of the proposed approach, experimental results are presented using real video sequences.

1 Introduction

Detection and localization of moving objects in an image sequence is a crucial issue of moving video [20], as well as for a variety of applications of Computer Vision, including object tracking [5], fixation and 2-D/3-D motion estimation. For MPEG-4 video object manipulation [18], the video object plane extraction could be based on change detection and moving object localization. For videoconferencing applications this motion analysis techniques could be used in the place of "blue-screening" techniques. Moving objects could be used for content description in MPEG-7 applications. In traffic monitoring, tracking of moving vehicles is needed, and in other cases visual surveillance is used for detecting intruding objects.

In the case of a static camera, detection is often based only on the inter-frame difference. Detection can be obtained by thresholding, or using more sophisticated methods taking into account the neighborhood of a point in a local or global decision criterion. In many real world cases, this hypothesis is not valid because of the existence of ego-motion (*i.e.*, visual motion due to the movement of the camera). This problem can be solved by computing the camera motion and creating a compensated sequence. In this work only the case of a static scene is considered.

This paper deals with both problems, change detection and moving object localization. Indeed, complete motion detection is not equivalent to temporal change detection. Presence of motion usually causes three kinds of "change regions" to appear. They correspond to (1) *the uncovered static background*, (2) *the covered background*, and (3) *the overlap of two successive object projections*. Note also that regions of third class are difficult to recover by a temporal change detector, when the object surface intensity is rather uniform. This implies that a complementary computation must be performed after temporal change detection, to extract specific information about the exact location of moving objects.

Simple approaches to motion detection consider thresholding techniques pixel by pixel [8], or blockwise difference to improve robustness against noise [21]. More sophisticated models have been considered within a statistical framework, where the inter-frame difference is modeled as a mixture of Gaussian or Laplacian distributions [20]. The use of Kalman filtering for certain reference frames in order to adapt to changing image characteristics has been investigated also [11]. The use of first order Markov chains [6] along rows and of two-dimensional causal Markov fields [9] has also been proposed to model the motion detection problem.

Spatial Markov Random Fields (MRFs), through Gibbs distribution have been widely used for modeling the change detection problem [1], [2], [3], [11], [14] and [19]. These approaches are based on the construction of a global cost function, where interactions (possibly nonlinear) are specified among different image features (*e.g.*, luminance, region labels). Besides, multiscale approaches have been investigated in order to reduce the computational complexity of the deterministic cost minimization algorithms [14] and to get estimates of improved quality.

In [16] a motion detection method based on a MRF model was also proposed, where two zero-mean generalized Gaussian distributions were used to model the inter-frame difference. For the localization problem, Gaussian distribution functions were used to model the couple of the intensities at the same site in two successive frames. In each problem, a cost function was constructed based on the above distributions along with a regularization of the label map. Deterministic relaxation algorithms were used for the minimization of the cost function.

On the other hand approaches based on contour evolution [12] [4], or on partial differential equations are also proposed in the literature. In [7] a three step algorithm is proposed including a contour detection, an estimation of the velocity field along the detected contours and finally the moving contours are determined. In [15], the contours to be detected and tracked are modeled as geodesic active contours. For the change detection problem a new image is generated, which exhibits large gradient values around the moving area. The problem of object tracking is posed in a unified active contour model including both change detection and object localization.

In this paper we propose a new method based on level set approaches. An innovative idea here is that the propagation speed is label dependent. Thus for the problem of change detection, where two labels are characterizing image sites, an initial statistical test gives seeds for performing the contour propagation. The propagation of the labels is implemented using an extension of the fast marching algorithm, named multi-label fast marching algorithm. The change detection maps are used for initializing another level set algorithm, based on the spatial gradient, for tracking the moving object boundary. For more accurate results and for having an automatic stopping criterion, two fronts are propagated in converging opposite directions, and they are designed for contact on the object boundary, where the spatial gradient is maximum.

The remainder of this paper is organized as follows. In Section 2 we consider the motion detection problem and we propose a method for initially labeling sites with high confidence. In Section 3 a new algorithm based on level set approach is introduced for propagating the initial labels. In Section 4, we present the moving object localization problem, as well as a fast marching algorithm for locating the object's boundary. In order to check the efficiency and the robustness of the proposed method, experimental results are presented on real image sequences. Results illustrating the different methods are provided in all the above described sections, and in Section 5, where the final conclusions are given too.

2 Detection of moving objects

2.1 Problem position

Let $D = \{d(x,y), (x,y) \in S\}$ denote the gray level difference image with

$$d(x,y) = I(x,y,t+1) - I(x,y,t) \tag{1}$$

The change detection problem consists of a "binary" label $\Theta(x,y)$ for each pixel on the image grid. We associate the random field $\Theta(x,y)$ with two possible events, $\Theta(x,y) = static$ *(or unchanged pixel)*, if the observed difference $d(x,y)$ supports the hypothesis for static pixel (H_0), and $\Theta(x,y) = mobile$ *(or changed pixel)*, if the observed difference supports the alternative hypothesis (H_1), for mobile pixel. Under these assumptions, for each pixel it can be written

$$\begin{array}{c} H_0 : \Theta(x,y) = static \\ H_1 : \Theta(x,y) = mobile \end{array} \tag{2}$$

Let $p_{D|static}(d|static)$ (resp. $p_{D|mobile}(d|mobile)$) be the probability density function of the observed inter-frame difference under the H_0 (resp. H_1) hypothesis. These probability density functions are assumed to be homogeneous, *i.e.* independent of the pixel location, and usually they are under Laplacian or Gaussian law. We use here a zero-mean Laplacian distribution function to describe the statistical behavior of the pixels for both hypotheses, thus the conditional probability density function of the observed difference values is given by

$$p(d(x,y)|\Theta(x,y) = l) = \frac{\lambda_l}{2} \quad e^{-\lambda_l |d(x,y)|} \tag{3}$$

Let P_{static} (resp. P_{mobile}) be the *a priori* probability of hypothesis H_0 (resp. H_1). Observed difference values are assumed to be obtained by selecting a label $l \in \{static, mobile\}$ with probability P_l and then selecting an inter-frame difference d according to the probability low $p(d|l)$. Thus the probability density function is given by

$$p_D(d) = P_{static}\; p_{D|static}(d|static) + P_{mobile}\; p_{D|mobile}(d|mobile) \qquad (4)$$

In this mixture distribution $\{P_l, \lambda_l; l \in \{static, mobile\}\}$ are unknown parame-

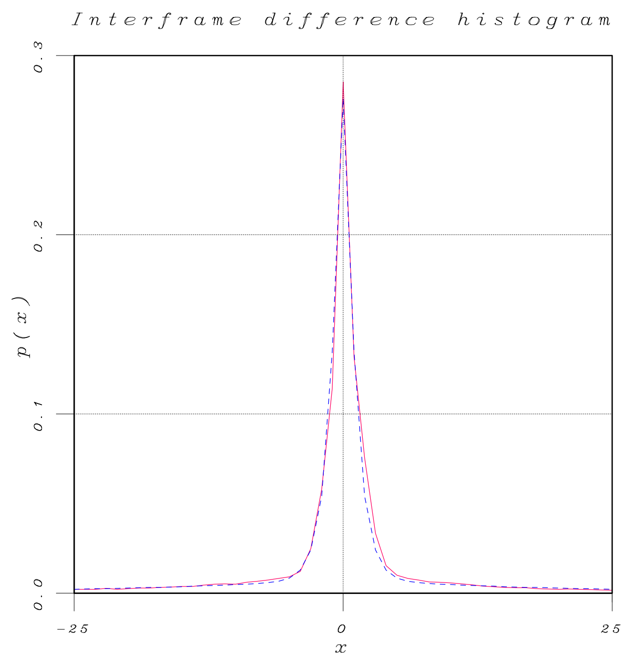


Fig. 1. Mixture decomposition in Laplacian distributions for inter-frame difference for *Trevor White*

ters. The principle of Maximum Likelihood is used to obtain an estimate of these parameters ([10] and [13]). The unknown parameters are iteratively estimated using the observed distribution of gray level inter-frame differences. An initial estimate is calculated using first, second and third order moments of the variable considered. In Figure 1, the histogram and the approximated probability density function (dashed line) for a test couple of frames is given [16].

2.2 Initial labeling

An initial map of labeled sites is obtained using statistical tests. The first test detects changed sites with high confidence. The false alarm probability is set to a small value, say P_{FA}. For the Laplace distribution used here, the corresponding threshold is

$$T_1 = \frac{1}{\lambda_0} \ln \frac{1}{P_{FA}} \tag{5}$$

Then a series of tests are used for finding unchanged sites with high confidence, that is with small probability of non-detection. For these tests a series of six windows of dimension $(2w+1)^2, (w = 1, \ldots, 6)$, are considered and the corresponding thresholds are preset as a function of λ_1.

Results of the initial processing are given in Fig. 2 for two different amounts of motion and for two couples of frames of the *White Trevor* image sequence. In these images "black" color means "unchanged" site, "white" color means "changed" site, and "gray" color means "unlabeled" site. As the amount of motion is more important, the discrimination is clearer. If the amount of motion is small, the number of ambiguous sites is important.

Fig. 2. Initial sets for two couples of frames of the *Trevor White* sequence

2.3 Label propagation

A multi-label fast marching level set algorithm, which is presented in the next section, is then applied for all sets of points initially labeled. This algorithm is an extension of the well-known fast marching algorithm [17]. The contour of each region propagates according to a motion field which depends on the label and on the absolute inter-frame difference. The exact propagation velocity for the "unchanged" label is

$$v_0(x,y) = \frac{1}{1 + e^{\beta_0(|d(x,y)| - n\zeta - \theta_0)}} \tag{6}$$

and for the "changed" label

$$v_1(x,y) = \frac{1}{1 + e^{\beta_1(\theta_1 - |d(x,y)| - (n+\alpha)\zeta)}} \tag{7}$$

where n is the number of the neighbouring pixels already labeled with the same candidate label, and α takes a positive value if the pixel at the same site of the previous label map is an interior point of a "changed" region, else it takes a zero value. The parameters $\beta_0, \beta_1, \theta_0, \theta_1$ and ζ are adapted according to the initial label map and the features characterizing the data (P_l, λ_l).

We use the fast marching algorithm for advancing the contours towards the unlabeled space. The dependence of the propagation speed only on the pixel properties, and not on contour curvature measures, is not a strong disadvantage here. In addition, the above defined propagation speed has an interesting neural interpretation. In fact, in both cases and with different parameters, the speed function is a sigmoid function with an input ($|d(x,y)|$, the inter-frame difference), and some synaptic links activated only if the states of the neurons are the same.

3 Multi-label fast marching algorithm

The fast marching level-set algorithm introduced by Sethian [17] computes a constructive solution to the stationary level set equation

$$|\nabla T(x,y)| = \frac{1}{v(x,y)} \tag{8}$$

where $v(x,y)$ corresponds to the velocity of the moving front while $T(x,y)$ is a map of crossing times. The curves are advanced monotonically according to the propagation speed field. The position of the curve at a given time t_0 is given by the equation $T(x,y) = t_0$. In the case of a suitably defined gradient estimation, Eq.(8) could be solved without iterative techniques, but by building the solution pixel by pixel from smallest values of the arrival time $T(x,y)$ to largest.

The proposed multi-label version of the fast marching algorithm solves the same problem for the case of any number of independent contours propagating with possibly different velocities, which are supposed to "freeze", when they cross over each other. In this approach two properties of each pixel are calculated: the arrival time and the region or contour that first reached the specific pixel.

Our algorithm takes advantage of the fact that the fast marching algorithm sweep the pixels in a time-advancing fashion in order to limit redundant recalculations only to the pixels of contact between contours. For each pixel a list of contour (or label) candidacies is maintained. A candidacy can only be introduced by a neighboring pixel being fixated to a certain label. It follows that no more than four candidacies may coexist per pixel, regardless of the number of labels contained in the whole image. Additionally, multiple candidacies can occur in pixels belonging to the border between two labels only, which illustrates the fact that multiple recalculations of arrival times are rather scarce. For the final decision map, the label carrying the smallest arrival time is selected for every pixel.

We now present the new multi-label fast marching level set algorithm.

```
Initialize
    For each pixel p in decision map
        If decision exists for p
            Set arrival time to zero for p
            For each neighboring pixel q lacking a decision
                - add label of pixel p to list of label candidacies for q,
                - mark it as trial,
                - give an initial estimate of the arrival time
        Else
            Set arrival time to infinity for p
Propagate
    While trial non alive label candidacies exist
        Select trial candidate c with smallest arrival time
        Mark c as an alive label candidacy
        If no decision exists for pixel p owning c
            Decide for p the label and arrival time of c
        For each undecided neighboring pixel q lacking a candidacy
        for the label of p
            - add label of pixel p to list of label candidacies for q,
            - mark it as trial
        For each neighboring pixel q containing a trial candidacy d
        for the label of c
            Recalculate arrival time of d
```

For the efficient location of the candidacy with the smallest arrival time a priority queue is utilized. This allows for insertion and deletion of pixels to/from the candidacy list at a cost proportional to the logarithm of its size. Pixel candidacies themselves, being up to four, are keeped in a linked list for ease of implementation. The above facts indicate an execution cost of order $N \log N$ over the uninitialized pixels. Moreover, in practice it is expected to run in no more than twice the time of the traditional fast marching algorithms regardless of the actual number of labels used.

4 Moving Object Localization

4.1 Initialization

The change detection stage could be used for initialization of the moving object tracker. The objective now is to localize the boundary of the moving object. The ideal change area is the union of sites which are occupied by the object in two successive time instants

$$C(t, t+1) = \{O(i, j, t)\} \cup \{O(i, j, t+1)\} \tag{9}$$

Let us also consider the change area

$$C(t, t-1) = \{O(i, j, t)\} \cup \{O(i, j, t-1)\} \tag{10}$$

It can easily be shown that

$$C(t,t+1) \cap C(t,t-1) = \{O(i,j,t)\} \cup (\{O(i,j,t+1)\} \cap \{O(i,j,t-1)\}) \quad (11)$$

This means that the intersection of two successive change maps is a better initialization for moving object localization, than each of them. In addition sometimes

$$(\{O(i,j,t+1)\} \cap \{O(i,j,t-1)\}) \subset \{O(i,j,t)\}$$

If this is true, then

$$\{O(i,j,t)\} = C(t,t+1) \cap C(t,t-1)$$

Fig. 3. Detection of Moving Objects: *Trevor White*

Knowing that there exist some errors in change detection and that sometimes under some assumptions the intersection of the two change maps gives the object location, we propose to initialize a level set contour search algorithm by this map. This search will be performed in two stages: first, an area containing the object's boundary is extracted, and second, the boundary is detected. The description of these stages follows.

In Fig. 3 we give the initial position of the moving contours for the first frame of the two couples given in Fig. 2.

4.2 Extraction of the uncertain area

The objective now is to determine the area which contains the object's boundary with extremely high confidence. Because of errors resulting from the change detection stage, and also because of the fact that the initial boundary is, in principle, placed outside the object, as shown in the previous subsection, it is needed to find an area large enough to contain the object's boundary. The task is simplified, if some knowledge about the background is acquired. In absence of knowledge concerning the background, the initial boundary could be relaxed in both directions, inside and outside, with a constant speed, which may be different for the two directions. In this area will then the photometric boundary be searched.

For cases where the background could be easily described, a level set approach extracts the zone of object's boundary. Let us suppose that the image intensity on the backround could be described by a Gaussian random variable with mean value, μ, and variance, σ^2. This model could be locally adapted. For the *White Trevor* sequence used here for illustrating results, a global backgound distribution is assumed.

The speed of uncertain area propagation is dependent on the label given by the initialization, and defined for the inner border as

$$v_o = c_o + \frac{d_o}{1 + e^{\frac{(\bar{I}-\mu)^2}{\sigma^2} - 1}} \tag{12}$$

where $\bar{I}$ is the mean value of the intensity in a 3×3 window centered at the examined point. For the outer border the speed is defined as

$$v_b = \frac{d_b e^{\frac{(\bar{I}-\mu)^2}{\sigma^2} - 1}}{1 + e^{\frac{(\bar{I}-\mu)^2}{\sigma^2} - 1}} \tag{13}$$

Thus for a point on the inner border, if its intensity is very different from that of the background, it is advancing with only the constant speed c_o. In contrast, the propagation of a point on the outer border is decelerated, if its intensity is similar to that of the background. The width of the uncertain zone is determined by a threshold on the arrival times, which depends on the size of the detected objects.

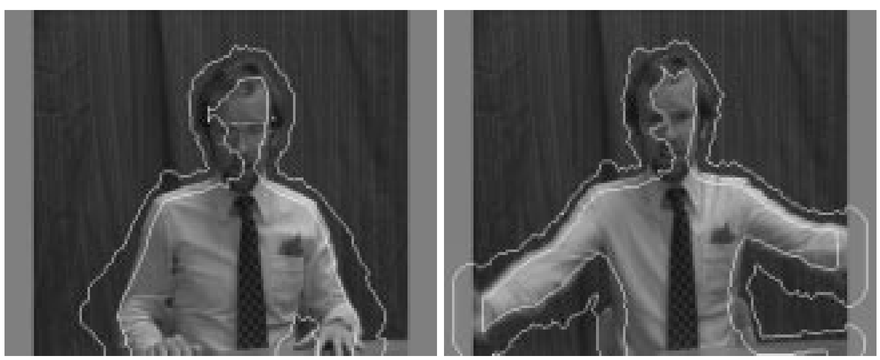


Fig. 4. Extraction of the uncertain area where the boundary is.

We give results (Fig. 4) illustrating this stage on the same image sequence and for two frames, where the initial estimate is less accurate. The initial location errors come from the amount of motion, and from its temporal change, like the change on the direction of motion.

4.3 Gradient-based object localization

The last stage involves determining the boundary of the object based on the image gradient. The two extracted boundaries are propagated in opposite directions, the inner outside and the outer inside. The boundary is determined as the place of contact of the two borders. The propagation speed for both is

$$v = \frac{1}{1 + e^{\gamma(\|\nabla I\| - \theta)}} \tag{14}$$

The parameters γ and θ are adapted to the data. An estimation of the variance of the gradient for the non-edge pixels of the undecided zone is performed. The parameter θ takes this value, and γ is the inverse of this value. Thus the two borders are propagating rapidly in the "smooth" area, and they are stopped on the boundaries of the object.

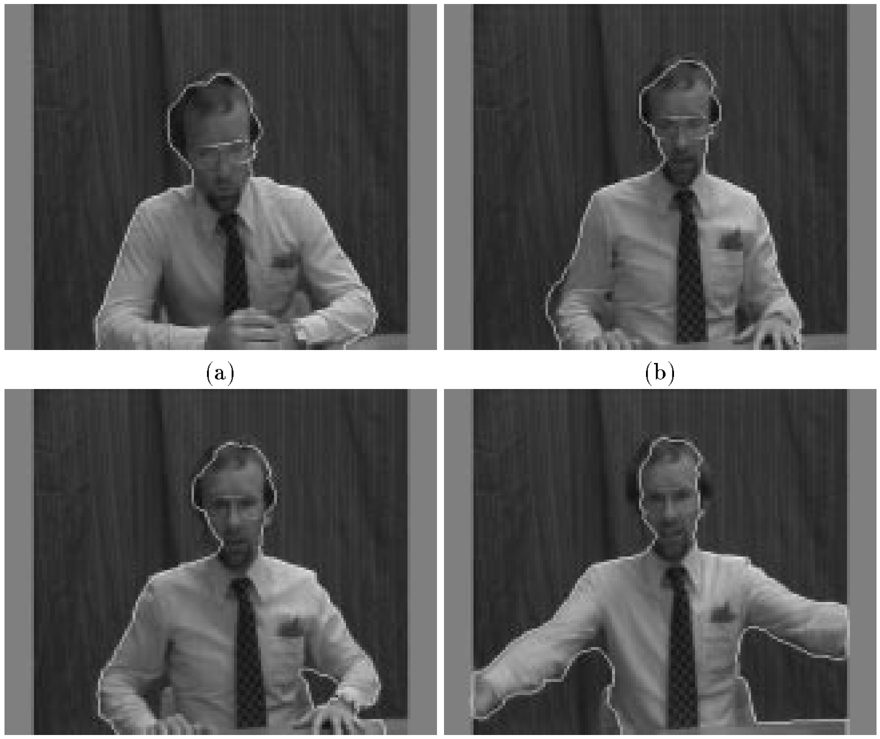
As the initial object location resulting from the change detection module sometimes may be in some distance from the real boundary, a second relaxation of the boundary may be then needed. This last relaxation takes place in a zone less large than the first one, and globally it improves the result. This is illustrated in the frame (b) of Fig. 5.

In Fig. 5 are given the same frames ((a) and (c)) as in Fig. 3 with the final result of localization, and in addition two other with some problems on the accuracy of the localization. For these two frames the uncertain area is given in Fig. 4.

5 Results and conclusions

Before concluding we would like to give some more results on real image sequences. In addition to the videoconferencing sequence *Trevor White*, which illustrates the techniques proposed and the different stages of the introduced algorithm, we have also applied the proposed methods to a telesurveillance (MPEG4 *Hall monitor*) sequence, and to a traffic monitoring sequence (*Highway*). We give results only for the last sequence in Fig. 6, because of the limited space. It is important to note here that the settings of the different parameters are exactly the same for the two image sequences.

In this article we propose at first a very interesting extension of the fast marching algorithm, in order to be able to consider multiple labels for the propagating contours. This allows to have purely automatic boundary search methods, and to obtain more robust results, as multiple labels are in competition. We have tested the new algorithm into the two stage problem of change detection and moving object localization. Of course, it is possible, and sometimes sufficient, to limit the algorithm into only one of these stages. This is the case for telesurveillance applications, where change detection with a reference frame gives the location of the moving object. In the case of a motion tracking (or rotoscopy) application, the stage of localization could be used for refining the tracking result. In any case, in this article we show that it is possible to locate a moving object without motion estimation, which, if it is added, it could improve further the already sufficiently accurate results.



(a) (b) (c) (d)

Fig. 5. Location of Moving Objects: *Trevor White*

References

1. T. Aach and A. Kaup. Bayesian algorithms for adaptive change detection in image sequences using markov random fields. *Signal Processing: Image Communication*, 7:147–160, 1995.
2. T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving video. *Signal Processing*, 31:165–180, 1993.
3. M. Bischel. Segmenting simply connected moving objects in a static scene. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:1138–1142, Nov. 1994.
4. A. Blake and M. Isard. *Active contours.* Springer, 1998.
5. P. Bouthemy and P. Lalande. Detection and tracking of moving objects based on a statistical regularization method in space and time. *Proc. of ECCV*, pages 307–311, 1990.
6. C. Cafforio and F. Rocca. Methods for measuring small displacements of television images. *IEEE Trans. on Information Theory*, IT-22:1973–1979, 1976.
7. V. Caselles and B. Coll. Snakes in movement. *SIAM Jour. on Numerical Analysis*, 33:2445–2456, Dec. 1996.
8. N. Diehl. Object-oriented motion estimation and segmentation in image sequences. *IEEE Trans. on Image Processing*, 3:1901–1904, February 1990.

Fig. 6. Location of Moving Objects: *Highway*

9. J. Driessen, J. Biemond, and D. Boekee. A pel-recursive segmentation algorithm for motion compensated image sequence coding. *Proceedings of ICASSP*, pages 1901–1904, April 1989.
10. R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* New York: Willey-Interscience, 1973.
11. K. Karmann, A. Brandt, and R. Gerl. Moving object segmentation based on adaptive reference images. *European Signal Processing Conf.*, pages 951–954, 1990.
12. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *Intern. Jour. of Computer Vision*, 1:321–332, Jan. 1988.
13. G. McLachlan, D. Peel, and W. Whiten. Maximum likelihood clustering via normal mixture model. *Signal Processing: Image Communication*, 8:105–111, 1996.
14. J-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Visual Communication and Image Representation*, pages 348–365, Dec. 1995.
15. N. Paragios and R. Deriche. Detecting multiple moving targets using deformable contours. *Intern. Conf. on Image Processing*, 1997.
16. N. Paragios and G. Tziritas. Adaptive detection and localization of moving objects in image sequences. *Signal Processing: Image Communication*, 14:277–296, Feb. 1999.
17. J. Sethian. Theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numerica*, pages 309–395, 1996.
18. T. Sikora. The mpeg-4 video standard verification model. *IEEE Trans. on Circuits and Systems for Video Technology*, 7:19–31, Feb. 1997.
19. Z. Sivan and D. Malah. Change detection and texture analysis for image sequence coding. *Signal Processing: Image Communication*, 6:357–376, 1994.
20. G. Tziritas and C. Labit. *Motion analysis for image sequence coding.* Elsevier, 1994.
21. O. Wenstop. Motion detection from image information. *Proc. of Scandinavian Conf. on Image Analysis*, pages 381–386, 1983.