Moving Object Localisation Using a Multi-Label Fast Marching Algorithm

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 2208, Heraklion, Greece
E-mails: {tziritas,sifakis}@csd.uoc.gr

July 14, 2000

Abstract

In this paper we address two problems crucial to motion analysis: the detection of moving objects and their localisation. Statistical and level set approaches are adopted in formulating these problems. For the change detection problem, the inter-frame difference is modelled by a mixture of two zero-mean Laplacian distributions. At first, statistical tests using criteria with negligible error probability are used for labelling as changed or unchanged as many sites as possible. All the connected components of the labelled sites are used thereafter as region seeds, 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 localisation 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 initialised by two curves evolving in converging opposite directions. The sites of curve contact determine the position of the object boundary. Experimental results using real video sequences are presented, illustrating the efficiency of the proposed approach.

1 Introduction

Detection and localisation of moving objects in an image sequence is a crucial issue for the analysis of moving video [25], 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 [23], the video object plane extraction could be based on change detection and moving object localisation. For videoconferencing applications these motion analysis techniques could be used in 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 neighbourhood of a point in a local or global decision criterion. In many real world cases, this hypothesis is not valid because of the presence of ego-motion (*i.e.*, visual motion caused by the camera's movement). This problem can be eliminated 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 localisation. Indeed, complete motion detection is not equivalent to temporal change detection. The 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 type are difficult to identify 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 [26]. More sophisticated models have been considered within a statistical framework, where the inter-frame difference is modelled as a mixture of Gaussian or Laplacian distributions [25]. The use of Kalman filtering for certain reference frames in order to adapt to changing image characteristics has also been investigated [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 the Gibbs distribution have been widely used for modelling the change detection problem [1], [2], [3], [11], [14] and [24]. These approaches are based on the construction of a global cost function, where interactions (possibly non-linear) are specified among different image features (*e.g.*, luminance, region labels). Multi-scale approaches have also been investigated in order to reduce the computational overhead of the deterministic cost minimization algorithms [14] and to improve the quality of the field estimates.

In [17] a motion detection method based on a MRF model was proposed, where two zero-mean generalised Gaussian distributions were used to model the inter-frame difference. For the localisation problem, Gaussian distribution functions were used to model 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, consisting of contour detection, estimation of the velocity field along the detected contours and finally the determination of moving contours. In [16], the contours to be detected and tracked are modelled 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 localisation.

In this paper we propose a new method based on level set approaches. The level set methodology was introduced by S. Osher and J. Sethian [15] and can handle a contour evolution, while naturally allowing changes in the topology of the segmented regions. A thorough presentation of the level set method is given in [19]. The fast marching level set algorithm introduced by Sethian [18] computes a constructive solution to the stationary level set equation

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

where $v(x,y)$ corresponds to the propagation speed at point (x,y) of the evolving front, while $T(x,y)$ is a map of crossing times. At any given time the location of the evolving active contour can be determined. The resulting segmentation is interpreted by means of the velocity field used. Given the limitation of a constantly positive velocity function and a suitable discrete gradient definition, the fast marching algorithm can construct a solution $T(x,y)$ which satisfies Eq. (1) all over the image without resorting to iterative methods. The running time of the fast marching algorithm is of order $n \log n$ over the image size, classifying it as a very efficient segmentation technique. A review of the fast marching algorithm appears in [20].

An innovative idea here is that the propagation speed can be made label-dependent. Thus for the problem of change detection, where image sites are characterized by two labels, 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, called the *multi-label fast marching algorithm*. The change detection maps are used for initialising another level set algorithm, based on the spatial gradient, for tracking the moving object boundary. For more accurate results and in order to have an automatic stopping criterion, two fronts are propagated in converging opposite directions, so as to meet on the object boundary, where the spatial gradient is maximum.

The remainder of this paper is organised as follows: In Section 2 we consider the motion detection problem and propose a method for initially labelling sites with high confidence. In Section 3 a new algorithm based on level set approaches is introduced for propagating the initial labels. In Section 4 we present the moving object localisation problem, as well as a fast marching algorithm for locating the object's boundary. In order to assess 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 each of the above sections, as well as in Section 5, where the final conclusions are given.

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

$$d(x,y) = I(x,y,t+1) - I(x,y,t). \tag{2}$$

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, $\Theta(x,y) = \text{static}$ (or *unchanged pixel*), if the observed difference $d(x,y)$ supports the hypothesis (H_0) for static pixel, and $\Theta(x,y) = \text{mobile}$ (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

$$\begin{array}{ll} H_0: & \Theta(x,y) = \text{static} \\ H_1: & \Theta(x,y) = \text{mobile} \end{array} \tag{3}$$

Let $p_{D|\text{static}}(d|\text{static})$ (resp. $p_{D|\text{mobile}}(d|\text{mobile})$) 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 Laplacian or Gaussian. We use here a zero-mean Laplacian distribution function to describe the statistical behaviour of the pixels for both hypotheses, thus the conditional probability density function of the observed difference values is given by

$$p(d(x,y)|\Theta(x,y) = l) = \frac{\lambda_l}{2} \; e^{-\lambda_l |d(x,y)|}. \tag{4}$$

In what follows we shall also use index '0' for *static* label and index '1' for *mobile* label. Let P_0 (resp. P_1) be the *a priori* probability of hypothesis H_0 (resp. H_1). Observed difference values are assumed to be obtained by selecting a label $l \in \{\text{static}, \text{mobile}\}$ 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

$$p_D(d) = P_0 \; p_{D|0}\,(d|\text{static}) + P_1 \; p_{D|1}(d|\text{mobile}). \tag{5}$$

In this mixture distribution $\{P_l, \lambda_l; l \in \{0,1\}\}$ are unknown parameters. The principle of Maximum Likelihood is used to obtain an estimate of these parameters ([10], [13]). The unknown parameters are iteratively estimated using the observed distribution of grey 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 pair of frames is shown [17].

2.2 Initial labelling

An initial map of labelled 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_F. For the Laplace distribution used here, the corresponding threshold is

$$T_1 = \frac{1}{\lambda_0} \ln \frac{1}{P_F}. \tag{6}$$

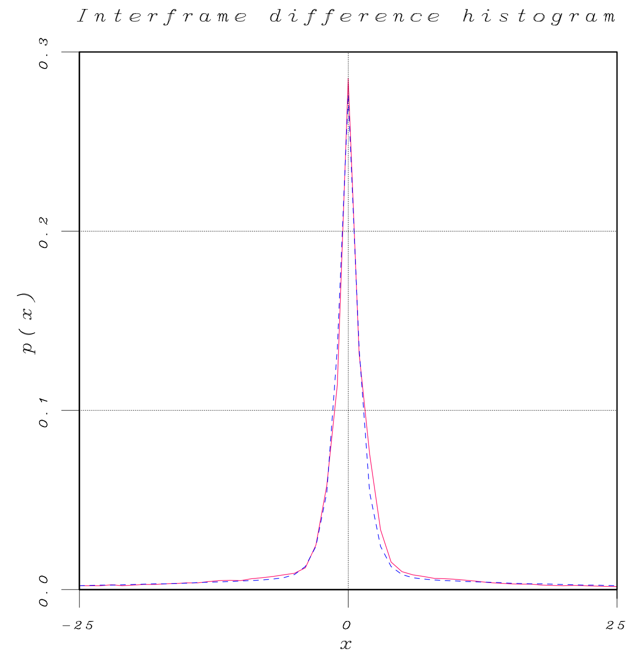


Figure 1: Mixture decomposition in Laplacian distributions for inter-frame difference for *Trevor White*.

Subsequently a series of tests is used for finding unchanged sites with high confidence, that is with small probability of non-detection. For these tests a series of five windows of dimension $(2w+1)^2$, $w = 2, \ldots, 6$, are considered and the corresponding thresholds are pre-set as a function of λ_1. Let us denote by B_w the set of pixels labelled as unchanged when testing window indexed by w. We set them as follows

$$B_w = \{(x,y) : \sum_{k=-w}^{w} \sum_{l=-w}^{w} |d(x+k, y+l)| < \frac{\gamma_w}{\lambda_1}\}, w = 2, \ldots, 6.$$

The probability of non-detection depends on the threshold γ_w, while λ_1 is inversely proportional to the dispersion of $d(x,y)$ under the "changed" hypothesis. As the evaluation of this probability is not straightforward, the numerical value of γ_w is empirically fixed. In the following table we give the values used in our implementation:

w	2	3	4	5	6
γ_w	0.4	1.6	3.5	7.0	12.0

Finally the union of the above sets $\cup_{w=2}^{6} B_w$ determines the initial set of "changed" pixels.

Results of the initial processing are given in Fig. 2 for two different amounts of motion and for two pairs of frames taken from the *Trevor White* image sequence. In these images black represents an "unchanged" site, white a "changed" site, and grey an "unlabelled" site. As expected in the right-hand map (frames 38–39), where the amount of motion is more prominant, the discrimination is clearer. If the amount of motion is small (left-hand map for frames 23–24), there is a significant number of ambiguous sites.

2.3 Label propagation

A multi-label fast marching level set algorithm, presented in the next section, is then applied for all sets of points initially labelled. This algorithm is an extension of the well-known fast marching algorithm [19]. The contour of each region is propagated according to a motion field which depends

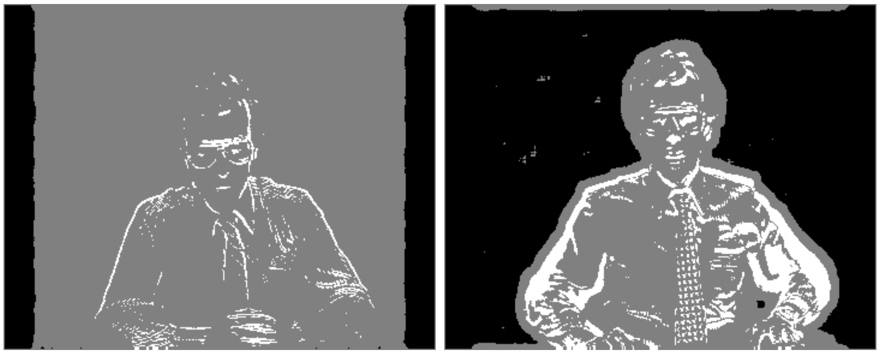


Figure 2: Initial sets for two pairs of frames of the *Trevor White* sequence.

on the label and on the absolute inter-frame difference. The label-dependent propagation speed is set according to the *a posteriori* probability principle. As the same principle will be used later for other level set propagations and for their respective velocities, we shall present here the fundamental aspects of the definition of the propagation speed. The candidate label is ideally propagated with a speed in the interval [0, 1], equal in magnitude to the *a posteriori* probability of the candidate label at the considered point. Let us define at a site s, for a candidate label l and for a data vector d the propagation speed as

$$v_l(s) = \Pr\{l(s)|d(s)\}$$

Then we can write

$$v_l(s) = \frac{p(d(s)|l(s))\Pr\{l(s)\}}{\sum_k p(d(s)|k(s))\Pr\{k(s)\}} = \frac{1}{1 + \sum_{k \neq l} \frac{p(d(s)|k(s))}{p(d(s)|l(s))} \frac{\Pr\{k(s)\}}{\Pr\{l(s)\}}}. \tag{7}$$

Therefore the propagation speed depends on the likelihood ratios and on the *a priori* probabilities. The likelihood ratios can be evaluated according to the assumptions on the data, and the *a priori* probabilities could be estimated, either globally or locally, or assumed all equal.

In the case of a decision between the "changed" and the "unchanged" labels according to the assumption of Laplacian distributions, the likelihood ratios are exponential functions of the absolute value of the inter-frame difference. In a pixel-based framework the decision process is highly noisy. Moreover, the moving object might be non-rigid, its various components undergoing different movements, *e.g.*, the head and the arms of *Trevor White*. In regions of uniform intensity the frame difference could be small, while the object is moving. The memory of the "changed" area of the previous frames should be used in the definition of the local *a priori* probabilities used in the propagation process. According to Equations (7) and (4) the two propagation velocities could be written as follows

$$v_0(x,y) = \frac{1}{1 + \frac{Q_1(x,y)\lambda_1}{Q_0(x,y)\lambda_0} e^{(\lambda_0 - \lambda_1)|d(x,y)|}}$$

and

$$v_1(x,y) = \frac{1}{1 + \frac{Q_0(x,y)\lambda_0}{Q_1(x,y)\lambda_1} e^{-(\lambda_0 - \lambda_1)|d(x,y)|}},$$

where the parameters λ_0 and λ_1 have been previously estimated. We distinguish the notation of the *a priori* probabilities because they should adapted to the conditions of propagation and to local situations. For this reason a heuristic approach is adopted for estimating the *a priori* probabilities. From the statistical analysis of data's mixture distribution we have an estimation of the *a priori* probabilities of the two labels. This is an estimation and not *a priori* knowledge. However, the initially labelled points are not necessarily distributed according to the same probabilities, because the initial detection depends on the amount of motion, which could be spatially and temporally variant. We define a parameter β measuring the divergence of the two probability distributions as follows:

$$\beta = \left(\frac{\hat{P}_0 P_1}{\hat{P}_1 P_0} \right)^{4(\hat{P}_0 + \hat{P}_1)},$$

where $\hat{P}_0 + \hat{P}_1 + \hat{P}_u = 1$, $\hat{P}_u$ being the percentage of unlabelled pixels. Then β will be the ratio of the *a priori* probabilities. In addition, for $v_1(x,y)$ the previous "change" map and local assignements are taken into account, and we define

$$\frac{Q_0(x,y)}{Q_1(x,y)} = \frac{e^{\theta_1 - (\alpha(x,y) + n_1(x,y) - n_0(x,y))\zeta}}{\beta},$$

where $\alpha(x,y) = \ln(2\delta(x,y) - 1)$, with $\delta(x,y)$ the distance of the (interior) point from the border of the "changed" area on the previous pair of frames, and $n_1(x,y)$ (resp. $n_0(x,y)$) the number of pixels in neighborhood already labelled as "changed" (resp. "unchanged"). Finally, the exact propagation velocity for the "unchanged" label is

$$v_0(x,y) = \frac{1}{1 + \beta \frac{\lambda_1}{\lambda_0} e^{(\lambda_0 - \lambda_1)|d(x,y)| + \theta_0 - (n_0(x,y) - n_1(x,y))\zeta}} \tag{8}$$

and for the "changed" label

$$v_1(x,y) = \frac{1}{1 + \frac{1}{\beta} \frac{\lambda_0}{\lambda_1} e^{\theta_1 - (\lambda_0 - \lambda_1)|d(x,y)| - (\alpha(x,y) + n_1(x,y) - n_0(x,y))\zeta}}. \tag{9}$$

In our current implementation the parameters are set as follows: $\zeta = 0.1T_1$ (see Eq. 6), $\theta_0 = 4\zeta$ and $\theta_1 = 5\zeta + 4$. In Figure 3 the two speeds are mapped as functions of the absolute inter-frame difference for typical parameter values near the boundary.

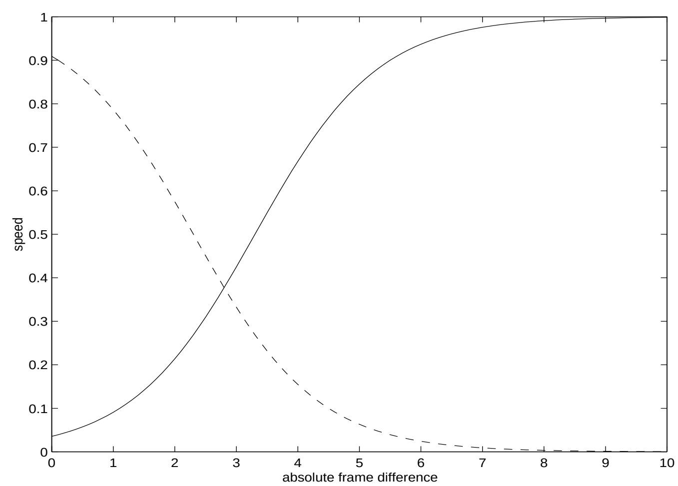


Figure 3: The propagation speeds of the two labels; solid line: "changed" label, dashed line: "unchanged" label.

We use the fast marching algorithm for advancing the contours towards the unlabelled space. Often in level set approaches constraints on the boundary points are introduced for in order to obtain a smooth and regularised contour and so that an automatic stopping criterion for the evolution is available. Our approach differs in that the propagation speed depends on competitive region properties, which both stabilise the contour and provide automatic stopping for the advancing contours. Only the smoothness of the boundary is not guaranteed. Therefore the dependence of the propagation speed on the pixel properties alone, and not on contour curvature measures, is not a strong disadvantage here. The main advantage is the computational efficiency of the fast marching algorithm.

The curve evolution for the initial map given in the left-hand picture of Fig. 2 is presented in Fig. 4. Topological changes occur as the labels are propagated, and the speeds seem to be well adapted to the considered detection problem.

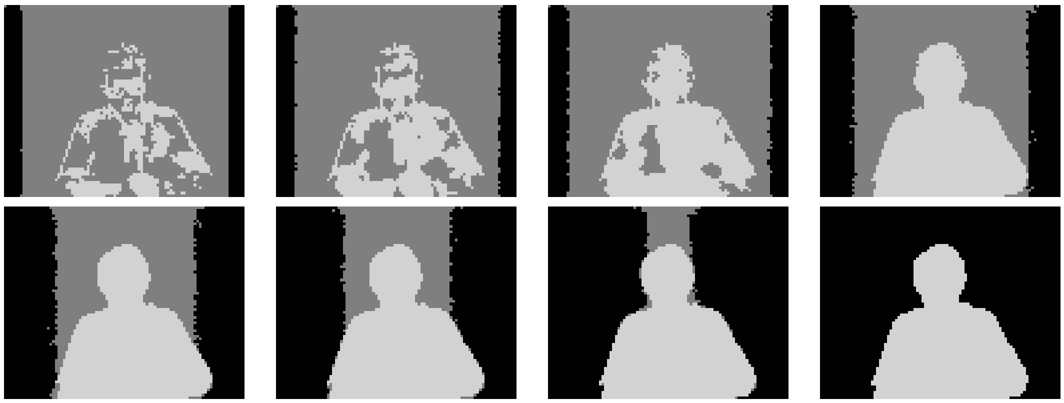


Figure 4: The evolution of the labelled curves for the left-hand initial map of Fig. 2.

3 Multi-label fast marching algorithm

The proposed algorithm, first presented in [22], is a variant of the fast marching algorithm which, while retaining the properties of the original, is able to cope with multiple classes (or labels). The same algorithm has also been used for other image segmentation tasks formulated as labelling problems [21]. The execution time of the new algorithm is effectively made independent of the number of existing classes by handling all the propagations in parallel and dynamically limiting the range of action for each label to the continuously shrinking set of pixels for which a final decision has not yet been reached. The propagation speed may also have a different definition for each class and, as seen in the previous section, the speed could take into account the statistical description of the considered class.

The algorithm as described below assumes the existence of an initialization for the crossing times $T(x, y)$, specifically its zero level set (*cf.* Fig. 2). Each pixel may carry several label candidacies. An "alive" candidacy represents a fixated arrival time, while that of a "trial" candidacy is subject to change. Alive candidacies are selected from the set of trial candidacies according to a minimum arrival time criterion and have their arrival time estimate fixated. The result of the algorithm is not only the crossing time at each point (x, y), but the corresponding labelling as well. The symbolic description of the algorithm follows:

The algorithm is supplied with a label map partially filled with decisions. A map with pointers to linked lists of trial pixel candidacies is also maintained. These lists are initially empty except for sites neighbouring initial decisions. For those sites a trial pixel candidacy is added to the corresponding list for each different label of neighbouring decisions and an initial arrival time is assigned. The arrival time for the initially labelled sites is set to zero, while for all others it is set to infinity. Apart from their participation in trial lists, all trial candidacies are independently contained in a common priority queue.

While there are still unresolved trial candidacies, the trial candidacy with the smallest arrival time is selected and turned alive. If no other alive candidacy exists for this site, its label is copied to the final label map. For each neighbour of this site a trial candidacy of the same label is added, if it does not already possess one, to its corresponding trial list. Finally, all neighbouring trial pixels of the same label update their arrival times according to the stationary level set equation (1).

While it may seem that trial pixels can exist per site for all different labels, in fact there can

InitTValueMap()
InitTrialLists()
while (ExistTrialPixels())
  {
   pxl = **FindLeastTValue()**
   MarkPixelAlive(pxl**)**
   UpdateLabelMap(pxl**)**
   AddNeighborsToTrialLists(pxl**)**
   UpdateNeighborTValues(pxl**)**
  }

be at most four, since a trial candidacy is only introduced by a finalised decision of a neighbouring pixel. In practice trial pixels of different labels coexist only in region boundaries, giving an average of label candidacies per pixel of two at most. Even in the worst case, it is evident that the time and space complexity of the algorithm is independent of the number of different labels. Experiments have indicated a running time no more than twice the time required by the single contour fast marching algorithm.

4 Moving object localisation

4.1 Initialisation

The change detection stage could be used for initialisation 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

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

where $\{O(i,j,t)\}$ is the set of points belonging to the moving object at time t. Let us also consider the change area

$$C(t-1,t) = \{O(i,j,t)\} \cup \{O(i,j,t-1)\}. \tag{11}$$

It can easily be shown that

$$C(t,t+1) \cap C(t,t-1) = \{O(i,j,t)\} \cup (\{O(i,j,t+1)\} \cap \{O(i,j,t-1)\}). \tag{12}$$

This means that the intersection of two successive change maps is a better initialisation for moving object localisation than either 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

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

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, that is the intersection of two successive change maps. 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. 5 is shown the initial position of the moving contours for the first frame of the two pairs given in Fig. 2.

Figure 5: Detection of moving objects: *Trevor White*.

4.2 Extraction of the uncertainty area

The objective now is to determine the area that 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 necessary to find an area large enough to contain the object's boundary. This task is simplified if some knowledge about the background is available. In the 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. Within this area then we search for the photometric boundary.

The objective is to place the inner border on the moving object and the outer border on the background. We insist here that *inner* means inside the object and *outer* means outside the object. Therefore if an object contains holes the inner border corresponding to the hole includes the respective outer border, in which case the inner border is expanding and the outer border is shrinking. In any case the object contour is expected to be between them at every point and under this assumption it will be possible to determine its location by the gradient-based module described in the next subsection. Therefore, the inner border should advance rapidly for points on the background, and slowly for points on the object. The opposite should be happen for the outer border.

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

The propagation speeds will be also determined by the likelihood principle. If, as assumed, the intensity on the background points is distributed according to the Gaussian distribution, the local average value of the intensity should also follow the Gaussian distribution with the same mean value and variance proportional to σ^2. The likelihood test on the goodness of this hypothesis is based on the normalised difference between the average and the mean value

$$\frac{(\bar{I}-\mu)^2}{\sigma^2}$$

where $\bar{I}$ is the average value of the intensity. A low value means a good fit with the background. Therefore the inner border should advance more rapidly for low values of the above statistics, while the outer border should be decelerated for the same values.

On the other hand it is almost certain that the border resulting from the previous stages is located on the background. Thus the probability of being on the background is much higher than the probability of being on the object. For the outer border the speed is defined as

$$v_b = \frac{1}{1 + c_b e^{-4\frac{(\bar{I}-\mu)^2}{\sigma^2}}} \tag{13}$$

where $\bar{I}$ is the mean value of the intensity in a 3×3 window centered at the examined point. Therefore it is considered that the variance of $\bar{I}$ is equal to $\sigma^2/8$. According to Equation (7) the constant c_b is

$$c_b = \frac{P_b}{P_o} \frac{\Delta}{\sigma\sqrt{2\pi}},$$

where P_b and P_o are the *a priori* probabilities of being on the background or on the moving object, respectively. We have assumed that in the absence of knowledge the intensity on the object is uniformly distributed in an interval whose the width is Δ. As the initial contour is more likely located on the background, P_o is given a smaller value than P_b. The outer border advances with the complementary speed

$$v_o = 1 - v_b, \tag{14}$$

using the same local variance computation. The plots of the two borders' speeds are shown in Fig. 6 as a function of the normalised difference between the mean and the average values with $c_b = 5$. The width of the uncertainty zone is determined by a threshold on the arrival times, which depends

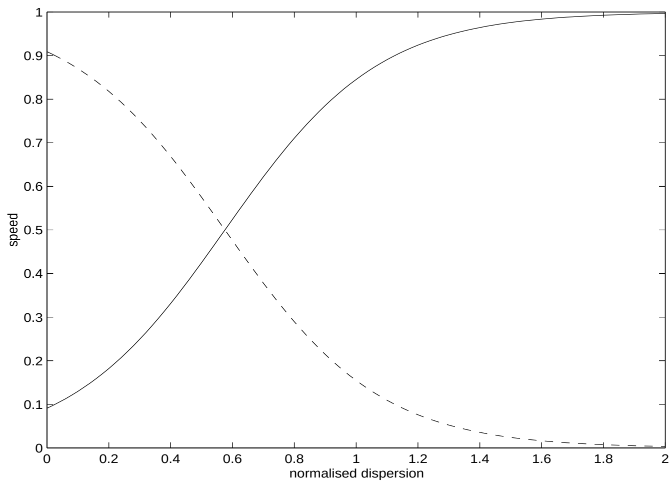


Figure 6: The propagation speeds of the borders defining the uncertainty area; solid line: v_b, dashed line: v_o.

on the size of the detected objects and on the amount of motion and which provides the stopping criterion. At each point along the boundary the distance from a corresponding "center" point of the object is determined using a heuristic technique for fast computation. The uncertainty zone is a fixed percentage of this radius modified in order to be adapted to the motion magnitude. However, motion is not estimated, and only a global motion indicator is extracted from the comparison of the consecutive changed areas.

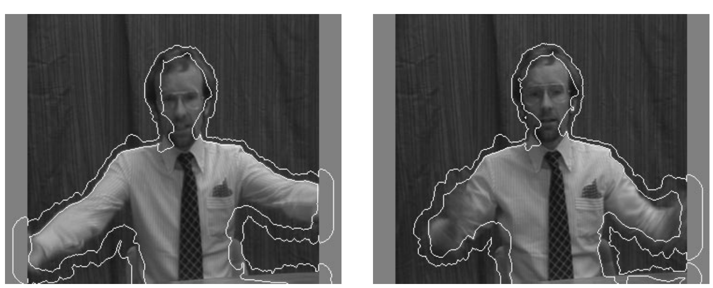


Figure 7: Extraction of the uncertainty area on two frames of the *Trevor White* sequence.

We present results (Fig. 7) illustrating this stage on the same image sequence and for two frames, where the initial estimate is less accurate. The initial location errors arise from the amount of motion, and from its temporal change, such as changes in the direction of motion.

4.3 Gradient-based object localisation

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

$$v_g = \frac{1}{1 + c_g \frac{\|\nabla I\|}{\sigma_n} e^{\frac{\|\nabla I\|}{\sigma_n}}}. \tag{15}$$

The parameter σ_n is adapted to the data. An estimation of the variance of the gradient for the non-edge pixels of the undecided zone is performed. The object's boundary is expected to be situated in the undecided zone. Therefore a bimodal distribution of the gradient magnitude should be observed on the data. Then a robust estimation method can discriminate between the two categories: edge and non-edge pixels. At first the global variance is estimated, and then points with gradient magnitude less than three times this variance are classified as non-edge pixels, and the parameter σ_n is estimated from only these points. In the above formulation of the gradient-based propagation speed we have assumed that the distribution for non-edge pixels is

$$p(\| \nabla I \|) = \frac{1}{\sigma_n} e^{-\frac{\|\nabla I\|}{\sigma_n}}$$

and for edge pixels the distribution is

$$p(\| \nabla I \|) = \frac{\| \nabla I \|}{\sigma_g^2} e^{-\frac{\|\nabla I\|}{\sigma_g}},$$

where $\sigma_g > \sigma_n$. The speed is plotted in Fig. 8 as a function of the normalised gradient ($\frac{\|\nabla I\|}{\sigma_n}$), with $c_g = 0.1$. Thus the two borders are propagating rapidly in the "smooth" area, and they are stopped

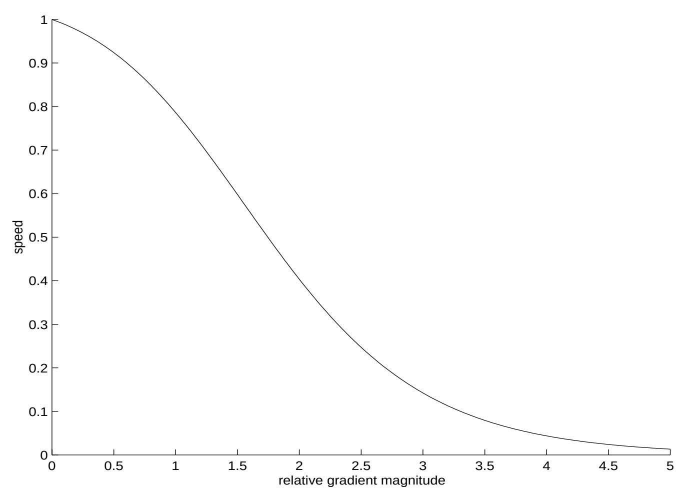
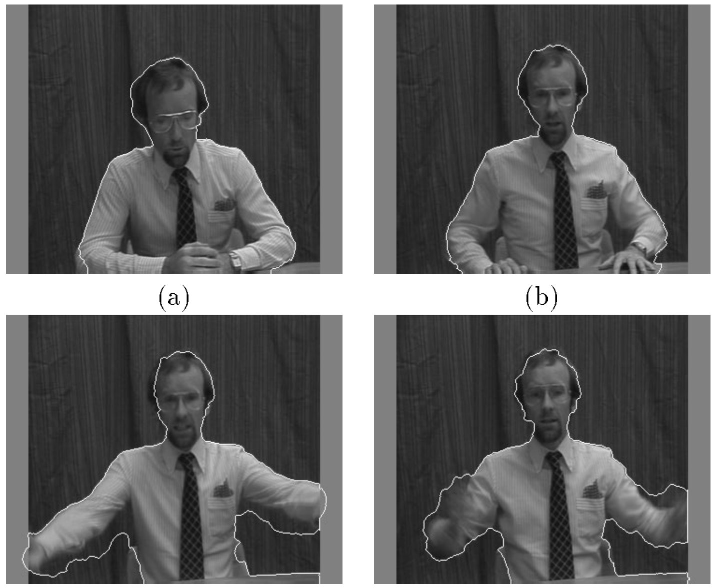


Figure 8: The gradient-based propagation speed of the borders converging to the object boundary.

on the boundaries of the object, since the propagation speeds for both curves are practically zero as the gradient is relatively high at such sites.

In Fig. 9, (a) and (b), the same frames presented in Fig. 5 are again shown, this time with the final result of localisation. (c) and (d) show the result for the frames of Fig. 7, which presents some problems for the accuracy of the localisation.



(a) (b)

(c) (d)

Figure 9: Location of moving objects: *Trevor White*.

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 sequence (MPEG4 *Hall monitor*, Fig. 10), and to a traffic monitoring sequence (*Highway*, Fig. 11). The results of these two sequences are somehow influenced from shadows. These problems are expected as intensity changes are also detected in the shadowed areas. Sometimes a photometric discontinuity could be detected in the edge of the shadow, but often the transition is gradual. It is important to note here that the settings of the different parameters are exactly the same for the two image sequences, as for the *Trevor White* sequence.

Figure 10: Detection of a moving human on *Hall monitor* sequence.

The extension to the fast marching algorithm proposed in this article is capable of dealing with multi-labelled propagating contours. This allows the use of purely automatic boundary search meth-

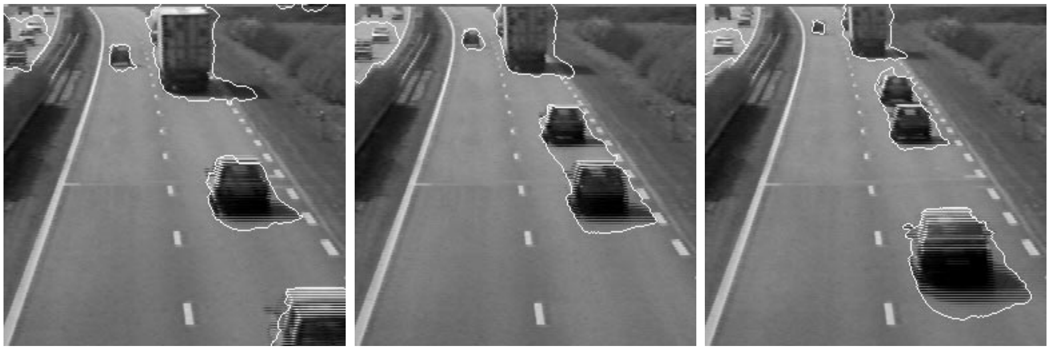


Figure 11: Location of moving objects: *Highway*.

ods, and furthermore yields more robust results, as multiple labels are in competition. In addition, we proposed a new approach for defining the propagation speeds of the labeled contours. The propagation speed is based on the statistical description of the propagated classes. An approximation of the *a posteriori* probability of the label determines, in different uses of the multi-label fast marching algorithm, the propagation speed of the label.

We have applied the new algorithm to the two-stage problem of change detection and moving object localisation. Of course, it is possible, and sometimes sufficient, to limit the algorithm to 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 localisation could be used for refining the tracking result. In any case, in this article it is shown that it is possible to locate a moving object without motion estimation, which, if added, could further improve the already quite accurate results.

Acknowledgment: This work has been funded in part under the European ESPRIT NEMESIS (New Multimedia Services using Analysis Synthesis) and the greek PENED "MPEG-4 Authoring Tools" projects.

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*, PAMI-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. In *Proc. of European Conf. on Computer Vision*, 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.

[9] J. Driessen, J. Biemond, and D. Boekee. A pel-recursive segmentation algorithm for motion compensated image sequence coding. In *Proc. IEEE Conf. on Acooustics, Speech, and Signal Processing*, Glasgow, 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. In *European Signal Processing Conf.*, 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] S. Osher and J. Sethian. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[16] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-22:266–280, March 2000.

[17] N. Paragios and G. Tziritas. Adaptive detection and localization of moving objects in image sequences. *Signal Processing: Image Communication*, 14:277–296, Feb. 1999.

[18] J. Sethian. A marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci.*, 93:1591–1595, 1996.

[19] J. Sethian. Theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numerica*, pages 309–395, 1996.

[20] J. Sethian. Fast marching methods. *SIAM Review*, 41:199–235, 1999.

[21] E. Sifakis, C. Garcia, and G. Tziritas. Bayesian level sets for image segmentation. *Journal of Visual Communication and Image Representation*, 2000 (to appear).

[22] E. Sifakis and G. Tziritas. Fast marching to moving object location. In *Proc. 2nd Intern. Conf. on Scale-Space Theories in Computer Vision*, Corfou, Greece, 1999.

[23] T. Sikora. The MPEG-4 video standard verification model. *IEEE Trans. on Circuits and Systems for Video Technology*, 7:19–31, Feb. 1997.

[24] Z. Sivan and D. Malah. Change detection and texture analysis for image sequence coding. *Signal Processing: Image Communication*, 6:357–376, 1994.

[25] G. Tziritas and C. Labit. *Motion analysis for image sequence coding*. Elsevier, 1994.

[26] O. Wenstop. Motion detection from image information. In *Proc. of Scandinavian Conf. on Image Analysis*, 1983.