VIDEO SEGMENTATION USING FAST MARCHING AND REGION GROWING ALGORITHMS

E. Sifakis, I. Grinias, and G. Tziritas
Dept. of Computer Science, University of Crete,
P.O.Box 2208, Heraklion, GREECE
Tel: +30 81 393517; fax: +30 81 393501
e-mail: {sifakis, grinias, tziritas}@csd.uoc.gr

ABSTRACT

The algorithm presented in this paper was proposed for comparisons using the COST 211 data set. It is comprised of three main stages: (1) classification of the image sequence, and parametric motion estimation in case of a moving camera, (2) change detection having as reference a fixed frame, an appropriately selected frame or a displaced frame, and (3) object localisation using local colour features. The image sequence classification is based on statistical tests on the frame difference. The change detection module uses the two-label fast marching algorithm. Finally, the object localisation uses a region growing algorithm based on the colour similarity.

1 INTRODUCTION

Video segmentation is a key step in image sequence analysis and its results are extensively used for determining motion features of scene objects, as well as for coding purposes to reduce storage requirements. The development and wide-spread use of the international coding standard MPEG-4 [11], which relies on the concept of image/video objects as transmission elements, has raised the importance of these methods. Moving objects could also be used for content description in MPEG-7 applications.

Various approaches have been proposed for video or spatio-temporal segmentation. An overview of segmentation tools, as well as of region-based representations of image and video, are presented in [6]. The video object extraction could be based on change detection and moving object localisation, or on motion field segmentation, particularly when the camera is moving. Our approach is based exclusively on change detection. The costly and potentially inaccurate motion estimation process is not needed. We present here some relevant work from the related literature for better situating our contribution.

In the framework of COST 211 an Analysis Model (AM) is proposed for image and video analysis and segmentation [2]. The essential feature of the AM is its ability to fuse information from different sources: colour segmentation, motion segmentation, and change detection. Kim *et al.* [5] proposed a method using global motion estimation, change detection, temporal and spatial segmentation.

Our algorithm, after the global motion estimation phase, is mainly based on change detection. The change detection problem is formulated as two-label classification. In [8] we have introduced a new methodology for pixel labelling called *Bayesian Level Sets*, extending the *level set* method [7] to pixel classification problems. We have also introduced the *Multi-label Fast Marching* algorithm and applied it at first to the change detection problem [10]. A more recent and detailed presentation is given in [9]. The algorithm presented in this paper differs from previous work in the final stage where the boundary based object localisation is replaced by a region based object labelling.

In Section 2 the method for selecting the appropriate frame difference for detecting the moving object is presented. In Section 3 we present the multi-label fast marching algorithm, which uses the frame difference and an initial labelling for segmenting the image into unchanged and changed regions with respect to the camera, *i.e.* changes independent of the camera motion. The last step of the entire algorithm is presented in Section 4 where a region growing technique extends an initial segmentation map. Section 5 concludes the paper, commenting on the obtained results.

2 FRAME DIFFERENCE

In our approach the main step in video object segmentation is change detection. Therefore for each frame we must first determine another frame which will be retained as reference frame and used for the comparison. Three different main situations may occur: (a) a constant reference frame, as in surveillance applications, (b) another frame appropriately selected, in the case of a still camera, and (c) a computed displaced frame, in the case of a moving camera.

The image sequence must be classified according to the above categories. We use a hierarchical categorization based on statistics of frame differences. At first the hypothesis (a) is tested against the other two. We can consider there to exist a unique background reference image if, for a number of frames, the observed frame differences are negligible. A test on the empirical probability distribution is then used.

When the reference is not constant we have to determine the more appropriate reference in order to identify independently moving objects. In order to determine the reference frame, it must be decided if the camera is moving or not. The test is again based on the empirical probability distribution of the frame differences.

Before considering the two possible cases we will present the statistical model used for the frame difference, because the determination of the appropriate reference frame is based on this model. Let $D = \{d(x,y),(x,y) \in S\}$ denote the gray level difference image. The change detection problem consists of determining 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 (*unchanged pixel*), and $\Theta(x,y) =$ mobile (*changed pixel*). Let $p_{D|\mathrm{static}}(d|\mathrm{static})$ (resp. $p_{D|\mathrm{mobile}}(d|\mathrm{mobile})$) 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 zero-mean Laplacian for both hypotheses ($l = 0, 1$)

$$p(d(x,y)|\Theta(x,y) = l) = \frac{\lambda_l}{2} \; e^{-\lambda_l|d(x,y)|}. \qquad (1)$$

Let P_0 (resp. P_1) be the *a priori* probability of hypothesis H_0 (resp. H_1). Thus the probability density function is given by

$$p_D(d) = P_0 \; p_{D|0}\,(d|\mathrm{static}) + P_1 \; p_{D|1}(d|\mathrm{mobile}). \qquad (2)$$

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 [3].

In the case of a still camera, the current frame must be compared to another frame sufficiently distinct, *i.e.*, is a frame where the moving object is displaced to be clearly detectable. For that the mixture of Laplacian distributions (2) is first identified. The degree of discrimination of the two distributions is indicated by the ratio of the two corresponding standard deviations, or, equivalently, by the ratio of the two estimated parameters λ_0 and λ_1. So we search for the closest frame, which is sufficiently discriminated from the current one. The threshold (T_λ) on the ratio of standard deviations is supplied by the user, and thus is determined the frame difference.

In the case of a moving camera the frame difference is determined by the displaced frame difference of successive frames. The camera movement must be computed for obtaining the displaced frame difference. We use a three-parameter model for describing the camera motion, composed of two translation parameters and a zoom parameter. The estimation of the three parameters is based on a frame matching technique with a robust criterion of least median of absolute displaced differences. For computational complexity reasons the median is determined using the histogram of the absolute displaced frame differences.

3 CHANGE DETECTION USING FAST MARCHING ALGORITHM

3.1 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 entire COST data set $P_F = 10^{-7}$. Subsequently a series of tests is used for finding unchanged sites with high confidence, *i.e.*, with a small probability of non-detection. For these tests a series of six windows of dimension $(2w+1)^2$, $w = 2, \ldots, 7$, is considered and the corresponding thresholds are preset 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

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

for $w = 2, \ldots, 7$. 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. Finally the union of the above sets $\cup_{w=2}^{7} B_w$ determines the initial set of "unchanged" pixels.

3.2 Label propagation

A multi-label fast marching level set algorithm is then applied to all sets of points initially labelled. This algorithm is an extension of the well-known fast marching algorithm [7]. The contour of each region is propagated according to a motion field, which depends 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 (x,y), for a candidate label l and for a data vector d the propagation speed as

$$v_l(x,y) = \Pr\{l(x,y)|d(x,y)\}$$

Then we can write

$$v_l(x,y) = \frac{p(d(x,y)|l(x,y))\Pr\{l(x,y)\}}{\sum_k p(d(x,y)|k(x,y))\Pr\{k(x,y)\}}. \qquad (3)$$

Therefore the propagation speed depends on the likelihood ratios and on the *a priori* probabilities. The likelihood ratios can be evaluated according to 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. 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 (3) and (1) the two propagation velocities could be written as follows

$$v_0(x,y) = \frac{1}{1 + \frac{Q_1(x,y;0)\lambda_1}{Q_0(x,y;0)\lambda_0} e^{(\lambda_0 - \lambda_1)|d(x,y)|}}$$

and

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

where the parameters λ_0 and λ_1 have been previously estimated. We distinguish the notation of the *a priori* probabilities defined here from those given in Equation (2), because they should adapted to the conditions of propagation and to local situations. Indeed, the above velocity definition is extended in order to include the neighbourhood of the considered point

$$v_l(x,y) = \Pr\{l(x,y)|d(x,y), \hat{k}(x',y'), (x',y') \in \mathcal{N}(x,y))\},$$

where the neighbourhood may depend on the label, and may be defined on the current frame as well as on previous frames. Therefore in this case the ratio of *a priori* probabilities is adapted to the local context, as in a Markovian model. A more detailed presentation of the approach for defining and estimating these probabilities follows.

From the statistical analysis of the data's mixture distribution we have an estimation of the *a priori* probabilities of the two labels (P_0, P_1). 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:

$$\beta = \left(\frac{\hat{P}_0 P_1}{\hat{P}_1 P_0} \right)^{\beta_0 (\hat{P}_0 + \hat{P}_1)},$$

where $\hat{P}_0 + \hat{P}_1 + \hat{P}_u = 1$, $\hat{P}_u$ being the percentage of unlabelled pixels. The parameter β_0 is fixed equal to 4 if the camera is not moving, and to 2 if the camera is moving. 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

$$\frac{Q_0(x,y;1)}{Q_1(x,y;1)} = \frac{e^{\theta_1 - (\alpha(x,y) + n_1(x,y) - n_0(x,y))\zeta}}{\beta},$$

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 neighbourhood already labelled as "changed" (resp. "unchanged"). The parameter ζ is adopted from the Markovian nature of the label process and it can be interpreted as a potential characterizing the labels of a pair of points. Finally, the exact propagation velocity for the "unchanged" label is

$$v_0(x,y) = \frac{1}{1 + \beta \frac{\lambda_1}{\lambda_0} e^{\theta_0 + (\lambda_0 - \lambda_1)|d(x,y)| - n_\Delta(x,y)\zeta}} \tag{4}$$

and for the "changed" label

$$v_1(x,y) = \frac{1}{1 + \frac{1}{\beta} \frac{\lambda_0}{\lambda_1} e^{\theta_1 - (\lambda_0 - \lambda_1)|d(x,y)| - (\alpha(x,y) - n_\Delta(x,y))\zeta}}, \tag{5}$$

where $n_\Delta(x,y) = n_0(x,y) - n_1(x,y)$. In the tested implementation the parameters are set as follows: $\theta_0 = 4\zeta$ and $\theta_1 = 5\zeta + 4$.

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 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 proposed algorithm 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 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 continually 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 the speed could take into account the statistical description of the considered class.

The high-level description of the algorithm is as follows:

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

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 maintained in a common priority queue, in order to facilitate the selection of the candidacy with the smallest arrival time.

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

$$\parallel \nabla T(x,y) \parallel = \frac{1}{v(x,y)} \tag{6}$$

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.

While it may seem that for a given site trial pixels can exist for all different labels, in fact there can 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; therefore the average number of label candidacies per pixel is at most two. 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 indicate a running time no more than twice that required by the single contour fast marching algorithm.

4 MOVING OBJECT LOCALIZATION USING REGION GROWING ALGORITHM

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

$$C(t,t+1) = O(t) \cup O(t+1), \tag{7}$$

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

$$C(t-1,t) = O(t) \cup O(t-1). \tag{8}$$

It can easily be shown that the intersection of two successive change maps $C(t-1,t) \cap C(t,t+1)$ is equal to

$$O(t) \cup (O(t+1) \cap O(t-1)).$$

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

$$(O(t+1) \cap O(t-1)) \subset O(t).$$

If this is true, then

$$C(t,t+1) \cap C(t,t-1) = O(t).$$

Of course the above described situation is an ideal one, and is a good approximation only in the case of a still camera. Thus in this case, knowing also that there are 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 region growing algorithm by this map, *i.e.*, 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.

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 emphasise 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, whereas the opposite should be happen for the outer border.

For cases in which the background can 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.

The propagation speeds will be also determined by the *a posteriori* probability 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 validity 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 in a window of size 3×3 centered at the examined point. 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

$$v_b = \frac{1}{1 + c_b e^{-4\frac{(\bar{I}-\mu)^2}{\sigma^2}}} \tag{9}$$

where it is considered that the variance of $\bar{I}$ is equal to $\sigma^2/8$. According to Equation (3) 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 (possibly equal to 255). As the initial contour is more likely located on the background, P_o is given a smaller value than P_b (typically $P_b/P_o = 3$). The outer border advances with the complementary speed

$$v_o = 1 - v_b, \tag{10}$$

using the same local variance computation.

The width of the uncertainty zone is determined by a threshold on the arrival times, which depends 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. The motion indicator is equal to the number of pixels with different labels on two consecutive "change" maps reported to the number of the detected object points.

4.3 Region growing-based object localisation

The last stage of object segmentation is carried out by a seeded region growing (SRG) algorithm which was initially proposed for static image segmentation using a homogeneity measure on the intensity function [1]. It is a sequential labelling technique, in which each step of the algorithm labels exactly one pixel, that with the lowest dissimilarity. In [4] the SRG algorithm was used for semi-automatic motion segmentation.

The segmentation result depends on the dissimilarity criterion, say $\delta(\cdot,\cdot)$. The colour features of both background and foreground are unknown in our case. In addition local inhomogeneity is possible. For these reasons we first determine the connected components already labeled, with two possible labels: background and foreground. On the boundary of all connected components we place representative points, for which we compute the locally average colour vector in the *Lab* system. The dissimilarity of the candidate for labelling and region growing point from the labelled regions is determined using this feature and the euclidean distance. After every pixel labelling the corresponding feature is up-dated. Therefore, we search for sequential spatial segmentation based on colour homogeneity, knowing that both background and foreground objects may be globally inhomogeneous, but presenting local colour similarities, sufficient for their discrimination.

For the implementation of the SRG algorithm, a list that keeps its members (pixels) ordered according to the dissimilarity criterion is used, traditionally referred to as Sequentially Sorted List (SSL). With this data structure available, the complete SRG algorithm is as follows:

```
S1 Label the points of the initial sets.

S2 Insert all neighbours of the initial sets into the SSL.

S3 Compute the average local colour vector for a predetermined subset of the boundary points of the initial sets.

S4 While the SSL is not empty:

    S4.1 Remove the first point $y$ from the SSL and label it.

    S4.2 Update the colour features of the representative to which the point $y$ was associated.

    S4.3 Test the neighbours of $y$ and update the SSL:

        S4.3.1 Add neighbours of $y$ which are neither already labeled nor already in the SSL, according to their value of $\delta(\cdot,\cdot)$.

        S4.3.2 Test for neighbours which are already in the SSL and now border on an additional set because of $y$'s classification. These are flagged as boundary points. Furthermore, if their $\delta(\cdot,\cdot)$ is reduced, they are promoted accordingly in the SSL.
```

When SRG is completed, every pixel is assigned one of the two possible labels: foreground or background.

5 RESULTS AND CONCLUSION

We applied the above described algorithm to the entire COST data set. The results are given in the following web page

`http://www.csd.uoc.gr/~tziritas/cost.html`

We obtained results ranging from good to very good, depending on the image sequence. The image sequence classification was always correct. The parametric motion model was estimated with sufficient accuracy. The independent motion detection was confident in the case of camera motion. The mixture of Laplacians was accurately estimated, and the initialization of the label map was correct, except for some problems caused by shadows, reflexions and homogeneous intensity on the moving objects. The fast marching algorithm was very efficient and performant. The last stage of moving object localisation can be further improved. The modelization of local colour and texture content could be possible, leading to a more adaptive region growing, or eventually a pixel labelling procedure.

Acknowledgment: This work has been funded in part by the European IST PISTE ("Personalized Immersive Sports TV Experience") and the Greek "MPEG-4 Authoring Tools" projects.

References

[1] R. Adams and L. Bischof, "Seeded Region Growing," IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol. 16, pp. 641–647, June 1994.

[2] A. A. Alatan, *et al.*, "Image Sequence Analysis for Emerging Interactive Multimedia Services–The European COST 211 Framework," IEEE Trans. on Circuits and Systems for Video Technology. Vol. 8, pp. 802–813, Nov. 1998.

[3] R. Duda and P. Hart, Pattern Classification and Scene Analysis. New York: Willey-Interscience, 1973.

[4] I. Grinias and G. Tziritas, "A semi-automatic seeded region growing algorithm for video object localization and tracking," Signal Processing: Image Communication. 2001 (to appear).

[5] M. Kim, *et al.*, "A VOP Generation Tool: Automatic Segmentation of Moving Objects in Image Sequences Based on Spatio-Temporal Information," IEEE Trans. on Circuits and Systems for Video Technology. Vol. 9, pp. 1216–1226, Dec. 1999.

[6] P. Salembier and F. Marques, "Region-based Representations of Image and Video: Segmentation Tools for Multimedia Services," IEEE Trans. on Circuits and Systems for Video Technology. Vol. 9, pp. 1147–1169, Dec. 1999.

[7] J. Sethian, "Theory, algorithms, and applications of level set methods for propagating interfaces," Acta Numerica. pp. 309–395, 1996.

[8] E. Sifakis, C. Garcia, and G. Tziritas, "Bayesian level sets for image segmentation," Journal of Visual Communication and Image Representation. 2001 (to appear).

[9] E. Sifakis and G. Tziritas, "Moving object localisation using a multi-label fast marching algorithm," Signal Processing: Image Communication. 2001 (to appear).

[10] E. Sifakis and G. Tziritas, "Fast marching to moving object location," Proceedings of the 2nd Intern. Conf. on Scale-Space Theories in Computer Vision, Corfou, Greece, 1999, pp. .

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