Fast Marching Techniques for Image Segmentation

Eftychis Sifakis, Georgios Tziritas

ABSTRACT A new region growing method is proposed for segmenting images. The region boundaries are formulated as level sets and the pixel labeling process is implemented using a new multi-label fast marching algorithm. The region contours are propagated with a velocity proportional to the *a posteriori* probability of the respective label. Statistical tests are performed to generate the initially labeled sets. Any image feature, given it is semantically relevant, can be considered for the segmentation process. Illustrations are given for combined luminance, chrominance and texture classification and segmentation in natural scenes. Moving object extraction based on change detection is also considered, which is performed as a two-label classification.

1 Introduction

Image segmentation is a vital component in any system involving image content analysis and computer vision. In the MPEG-4 standard, segmentation plays an important role in coding performance and object manipulation [27]. In the MPEG-7 standard a spatio-temporal locator is needed for the description of visual content [18]. A continuous effort has been made by the research community to solve the segmentation problem. The numerous existing approaches may be classified into two main categories: boundary-based and region-based.

Edge detection is the earliest of the boundary-based methods, based on local gradients [4]. Active contours [3], based on local gradients as well, have been introduced for tracking deformable moving objects [10], by minimizing a functional whose local maximum lies on the object boundary. Nevertheless, active contours are relatively noise sensitive. Moreover, their result depends on the initialization and they are not sufficiently topologically adaptive. Some progress has been made with the balloon model [7], where the external force applied to the curve is modified in order to make the active contour less sensitive to weak edges and spurious isolated edge points.

In the region-based approaches, techniques such as seeded region growing [1] or split-and-merge [17] were introduced. The labeling problem can also be globally formulated using Markov random field modeling. The final

solution is obtained by minimizing an energy function, where stochastic relaxation [9] may be used, but deterministic relaxation [2, 6] is often preferred, being less computationally expensive.

Efforts have also been made towards the unification of the contour- and the region-based approaches. Zhu and Yuille [32] proposed a region competition method which combines the geometrical features of snakes/balloon models and the statistical techniques of region growing. Paragios and Deriche [16] introduced the concept of geodesic active regions. The active contour evolves under the influence of two forces: a boundary force, which also contains curvature constraints and a region force, which aims to move the curve in the direction that maximizes the *a posteriori* segmentation probability.

Level set theories have been used in the formulation of several region- or boundary-based approaches for image segmentation. The mapping of active contours to the level set formulation [13, 15] has lifted many of the inconveniences of active contours, while the fast marching algorithm [20, 22] provides a computationally efficient method for tracking an evolving contour. The introduction of the geodesic active contours [29] has allowed the unification of the classical active contour based on energy minimization and the geometric active contours based on the theory of curve evolution. In this last approach, the algorithm initialization and termination problems are solved and more stable boundaries are obtained, by computing a level set solution using a geometric flow. In [5, 30] level set formulations are used for the maximization of a segment uniformity criterion, defined over a given classification, in conjunction with smoothness constraints over the boundaries of the resulting segments. Furthermore, in [19] the segmentation of an arbitrary number of classes is addressed leading to the combined evolution of several level set modeled contours. The last approach is based on the minimization of a functional that enforces region uniformity, contour smoothing, and classification coherence.

In [23] a new region-based methodology for image segmentation has been introduced, where statistical approaches for modeling the different region features are applied and labeling is achieved through a novel algorithm based on level sets and the monotonical evolution of region boundaries. As multiple simultaneously propagating contours are considered, we propose an extension of the level set approach to a multi-label framework, while allowing the propagation speed to depend on the respective region label. The segmentation performance strongly depends on the description of the label content and on the capability of incorporating the label description into the propagation velocity. For that purpose, we propose to define the propagation speed as the *a posteriori* probability of the respective label. A statistical approach, where the number of labels is assumed to be known, is therefore adopted, which requires adequate models. Pattern analysis techniques are used for the identification of the corresponding models.

The rest of this chapter is organized as follows. In Section 2, we review

the fast marching algorithm and we describe how we extend it to the multi-label case for classification purposes. In Section 3, we consider the case of combined colour and texture segmentation, where the intensity and chromaticity features are mainly captured by the histogram distribution, while Gaussian assumptions are possible too. The Discrete Wavelet Analysis is performed for the description of the texture content. The very important problem of automatic feature extraction for the description of label content is also considered, as well as the initialization of the level sets. In Section 4 we consider the moving object localization problem based on intensity change detection. Finally, conclusions are drawn in Section 5.

2 The Multi-Label Fast Marching algorithm

2.1 The stationary level set equation

Level set theory provides a framework for tracking the evolution of any curve in the plane given the velocity of the curve along its normal direction. In the pioneering work by Osher and Sethian [15] the various instances of the evolving contour are embedded as level sets of a function of higher dimensionality. In this framework the velocity function is free to include terms dependent on the geometrical characteristics of the evolving contour, such as the curvature or the outward normal to the moving contour.

Given the limitation of a constantly positive (or constantly negative) velocity function, leading to a monotonical motion of the propagating front, an *arrival time function* $T(\boldsymbol{s})$ corresponding to the time point when the moving contour crosses over the point $\boldsymbol{s}$ is well defined. Under this formulation the arrival time function T satisfies the stationary level set equation

$$F|\nabla T| = 1, \tag{1.1}$$

which simply states that the gradient of the arrival time function is inversely proportional to the velocity F of the contour at any given point. The preceding formulation allows for the constructive calculation of the arrival time function T without resorting to iterative methods. The trade-off for the computational efficiency is an inherent difficulty in integrating local properties of the evolving contour, such as curvature, in the velocity function F. Under those limitations the well-known Fast Marching level set algorithm [20], constructs a solution to Equation (1.1) from initial data with an execution cost of $n \log n$.

2.2 Multiple interface extensions and Fast Marching algorithm

The original formulation of the level set technique, as given in [15], applies specifically where there exists a clear distinction between an 'outside' and

an 'inside' region, separated by the evolving contour. Nevertheless, several applications, including multiple object segmentation and clustering require the consideration of more than two regions. In the simplest of cases where the distinct regions exhibit a smooth behavior and no triple points appear as the result of interface evolution, boundaries between different regions could be formulated as different level sets of the same function. Moreover, a technique for the proper handling of triple points and other singularities induced by multi-interface propagation can be found in [21]. These methods apply to the time-dependent level set formulation.

The work presented herein is motivated by the large number of applications that could be addressed by the tracking of the monotonical evolution of distinct regions into a special *blank* or *unlabeled* region and observing the final result of the initial regions' convergence over each other. The input to this approach would consist of an initialization for the expanding regions and a rule for their expansion into the blank region, in terms of their propagation velocity. Since the proposed framework includes strictly monotonical (expanding) motion of the considered regions, the stationary level set formulation would be best suited and the utilization of the Fast Marching algorithm would yield a favorable algorithmic complexity.

In the original two-region context, most shape modeling, feature extraction or segmentation applications of the Fast Marching level set algorithm involve initializing the arrival time map with seed regions, calculating arrival times for the rest of the spatial domain considered and either explicitly selecting a proper level set or utilizing an adequate criterion for picking the most appropriate propagation instance as the segmentation result. In the proposed framework the initialization consists of high confidence representatives of the regions in question, namely the *outside* and *inside* of the object(s) to be extracted. A third region corresponding to yet *undecided* sites of the segmentation domain is considered and velocities for the propagation of either region into the undecided one are supplied. The boundaries of both propagating regions are prescribed to freeze on contact, yielding the final segmentation solution, while eliminating the need for explicit selection of a propagation instance.

A trivial way of achieving the described functionality is to use the initialization of every region as the zero level set of an independent propagation, using the Fast Marching algorithm. Upon completion of all distinct propagations the first region managing to arrive at each site would be selected to specify its label and arrival time. This approach allows for the independent definition of propagation velocity for each expanding region, a property greatly exploited in the range of applications presented herein. Nevertheless, the execution cost for this algorithm scales with the number of independent regions and it can be shown that it is also subject to morphological instability, *e.g.* two regions that are separable with a single curve upon initialization are not bound to converge onto a single interface.

2.3 Fast Marching algorithm and labeling

The new Multi-Label Fast Marching algorithm presented in this chapter is an extension of the well-known Fast Marching algorithm introduced by Sethian [20], capable of manipulating multiple propagating contours, corresponding to the boundaries of competitively expanding regions. The low computational cost of the classical Fast Marching algorithm is maintained, since it is effectively made independent of the number of distinct regions present in the initialization. The new algorithm targets applications requiring static segmentation as well as labeling and clustering problems.

The Multi-Label Fast Marching algorithm computes a constructive solution to the stationary level set Equation (1.1) given initial conditions in terms of the zero level set of the arrival time function $T(\boldsymbol{s})$. Initializations may be provided for multiple non-intersecting regions for which the propagation velocity is allowed to follow an independent definition. All distinct regions (or *labels*) are propagated simultaneously according to their respective velocity definitions with the constraint that one region may not infiltrate a region that has been swept by another. The propagating regions evolve in a competitive fashion, with the algorithm reaching a deterministic halt once all sites of the considered domain have been labeled.

For the purposes of this text we shall limit the description of the new algorithm to the case of a two-dimensional image, although the algorithm can be trivially extended to three or more dimensions. All image pixels are either idle or carry a number of *candidacies* for different labels. Candidacies can be either *trial*, *alive* or *finalized*. *Trial* candidacies for a certain label are introduced to a specific pixel lacking a finalized candidacy when a neighboring pixel acquires a finalized candidacy for the same label. *Trial* candidacies carry an arrival time estimate which is subject to adjustment according to the process of its neighboring candidacies for the same label. *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 first trial candidacy to be turned alive per pixel is considered a finalized candidacy and is used in specifying the pixel label and arrival time in the final propagation result.

A symbolic description of the Multi-Label Fast Marching algorithm is as follows

```
InitTValueMap()
InitTrialLists()
while (ExistTrialPixels()) {
    pxl = FindLeastTValue()
    MarkPixelAlive(pxl)
    UpdateLabelMap(pxl)
    AddNeighboursToTrialLists(pxl)
    UpdateNeighbourTValues(pxl)
}
```

The algorithm is supplied with a label map partially filled with classification decisions. The arrival time for the initially labeled pixels is set to zero, while for all others it is set to a special value, *e.g.* infinity. A map of pointers to linked lists of candidacies is also maintained. Candidacy lists are initially empty, with the exception of unlabeled pixels that are neighbors to initial decisions, for which a trial candidacy is introduced carrying the label of the neighboring decision and an arrival time estimate is allocated. All trial candidacies are contained in a common priority queue.

If the candidacy queue is not empty, the trial candidacy with the smallest arrival time is selected and marked alive. If no other alive candidacies exist for this pixel, the candidacy is considered finalized and copied to the final label map. For all neighbors of this pixel lacking an alive candidacy, a trial candidacy for the same label is introduced. Finally, all neighboring trial candidates update their arrival times according to the revised condition. The re-estimation of the arrival times is performed with the utilization of the stationary level set Equation (1.1). Under a common gradient approximation robust in the presence of shocks, the equation is written

$$1/F_{ij}^2 = \max(\max(D_{ij}^{-x}T, 0), -\min(D_{ij}^{+x}T, 0))^2 + \max(\max(D_{ij}^{-y}T, 0), -\min(D_{ij}^{+y}T, 0))^2. \quad (1.2)$$

Equation (1.2) is solved for the value of the function T at the specified pixel. If the quadratic equation yields more than one solution, the greatest is used.

Although it seems possible that candidacies for all available labels may occur in a single site, it should be noted that a trial candidacy is only introduced by a neighboring candidacy being finalized, limiting the number of possible candidacies per pixel to a maximum of four. In practice, trial pixels of different labels coexist only in region boundaries, giving an average of at most two label candidacies per pixel. Even in the worst case, though, it is evident that the time and space complexity of the algorithm is independent of the number of different labels. Experiments have illustrated a running time no more than twice the time required by the single contour fast marching algorithm.

2.4 Label propagation

The multi-label fast marching level set algorithm, presented in the previous subsection, is applied to all sets of points initially labeled. The contour of each region propagates according to a velocity field which depends on the label and on the distance of the considered point from the candidate class. The label-dependent propagation speed is defined according to the *a posteriori* probability. The candidate label is ideally propagated with a speed in the interval (0, 1], which is equal to the *a posteriori* probability of the candidate label at the considered point. Let us define at a site $\boldsymbol{s}$, for a

candidate label $l(\boldsymbol{s})$ and for a data vector $x(\boldsymbol{s})$ the propagation speed as

$$F_l(\boldsymbol{s}) = \Pr\{l(\boldsymbol{s})|x(\boldsymbol{s})\}.$$

Then we can write

$$F_l(\boldsymbol{s}) = \frac{p(x(\boldsymbol{s})|l(\boldsymbol{s}))\Pr\{l(\boldsymbol{s})\}}{\sum_k p(x(\boldsymbol{s})|k(\boldsymbol{s}))\Pr\{k(\boldsymbol{s})\}} = \frac{1}{1+\sum_{k\neq l}\frac{p(x(\boldsymbol{s})|k(\boldsymbol{s}))}{p(x(\boldsymbol{s})|l(\boldsymbol{s}))}\frac{\Pr\{k(\boldsymbol{s})\}}{\Pr\{l(\boldsymbol{s})\}}} \quad (1.3)$$

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 or assumed all equal.

Under several commonly adopted models the probability density function is an exponential function of a distance measure of the data

$$p(x(\boldsymbol{s})|l(\boldsymbol{s})) = e^{-d_l(x(\boldsymbol{s}))}$$

If we assume that the *a priori* probabilities are all equal, we obtain

$$F_l(\boldsymbol{s}) = \frac{1}{1+\sum_{k\neq l} e^{d_l(x(\boldsymbol{s}))-d_k(x(\boldsymbol{s}))}}. \quad (1.4)$$

This expression of the propagation speed illustrates that when the propagated label is the correct one, all the exponents in the sum are negative and the speed is therefore close to unity. On the other hand, when the propagated label is incorrect, at least one exponent is positive, and therefore the speed is biased towards zero.

In order to compare the two speeds, let us now consider the case of two equiprobable labels. The mean time for advancing one unit length, if the curve evolves with a force corresponding to the region properties (without loss of generality, assume with label 0), is

$$E\{|\nabla T(\boldsymbol{s})|;0\} = 1 + \int \frac{p(x|1)}{p(x|0)}p(x|0)dx = 2.$$

If the curve evolves in the opposite labeled region, we have

$$E\{|\nabla T(\boldsymbol{s})|;1\} = 1 + \int \frac{p(x|1)}{p(x|0)}p(x|1)dx > 2 + \int p(x|1)\ln\frac{p(x|1)}{p(x|0)}dx.$$

The right-hand term is the Kullback distance D between the two distributions, and it is always positive. Therefore the ratio of the two mean times is

$$\frac{E\{|\nabla T(\boldsymbol{s})|;1\}}{E\{|\nabla T(\boldsymbol{s})|;0\}} = 1 + \frac{D(p(x|1),p(x|0))}{2} > 1. \quad (1.5)$$

The more discriminating the two probability distributions are, the more important the ratio of the two propagation speeds is, making more confident that the evolving curves are trapped by the boundary.

To illustrate the above using an example, let us suppose that the data is scalar distributed according to the Gauss law, with identical variance and two different mean values (μ_0 and μ_1). It is straightforward to show that

$$D(p(x|1), p(x|0)) = \exp\left(\frac{(\mu_1 - \mu_0)^2}{\sigma^2}\right). \tag{1.6}$$

Clearly, the evolution of the curve in a region which is differently labeled is decelerated, and the amount of deceleration depends in general on the signal-to-noise ratio. In practice a SNR equal to 10 is sufficient for stopping the evolution.

In this work, several features are assumed to follow the generalized zero-mean Gaussian distribution

$$p(x) = \frac{c}{2\sigma\,\Gamma(\frac{1}{c})}\, e^{-\left(\frac{|x|}{\sigma}\right)^c}, \tag{1.7}$$

where the parameter σ is the standard deviation and c reflects the sharpness of the probability density function. For $c = 2$, we obtain the Gaussian distribution and for $c = 1$, the Laplacian distribution. Then we obtain

$$D(p(x|1), p(x|0)) = \frac{1}{c}\left(\ln\frac{\sigma_0^c}{\sigma_1^c} + \frac{\sigma_1^c}{\sigma_0^c} - 1\right) \tag{1.8}$$

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 disadvantage here, because the propagation speed takes into account the region properties.

3 Colour and texture segmentation

Luminance, colour and/or texture features may be used, either alone or in combination, for segmentation. In our approach luminance and colour classes are described using the corresponding empirical probability distributions. For texture analysis and characterisation a multichannel scale/orientation decomposition is performed using Wavelet Frame Analysis [11, 12].

3.1 Texture and colour description

The features for texture segmentation are derived from the discrete wavelet frames analysis [28] using a pair of translation-invariant linear filters. The image is decomposed into components corresponding to different scales and orientations. The application of the separable 2-D filter bank on a given image yields three high-frequency detail components for each analysis

level plus a low-frequency approximation component at the last level. All components are shown [12] to be uncorrelated in the case of ideal filters.

The feature vectors for the texture content considered are the variances of the $N-1=3L$ high frequency components, for L levels of analysis, calculated over the distinct texture classes present in the image. The low frequency approximation is not used. If the luminance is sufficiently discriminating, it is also used as an additional feature described by its empirical probability distribution.

Lab colour space, designed to be perceptually uniform, was used here for colour feature extraction. Because the luminance, or intensity, component is used separately, only the chromaticity components (a, b) are used. In our work the local 2-D histograms of the (a, b) components were used as features. When some model of the distribution of the (a, b) histograms is fit (*e.g.*, Gaussian or Laplacian), the parameters of the model are used as the features. Often no such modeling is feasible, in which case local histogram estimation is required, making the procedure time consuming. The histograms are smoothed with a Gauss kernel to improve statistical robustness.

3.2 Automatic feature extraction

An essential step of the whole framework consists of estimating the features associated to the different labels. The only assumption we make is that the number of labels is known.

If the adoption of a model is reliable and, in particular, if this model is tractable, the mixture analysis is preferred. Nevertheless, the use of an *a priori* model might be difficult, if not arbitrary. After the multi-channel wavelet analysis, the generalized Gaussian model is plausible, but it is difficult to obtain accurate parameters using a mixture analysis, because the distributions are all zero-mean. Additionally, in some cases, luminance or chromaticity only may be sufficient for the segmentation, but no general model is applicable. Therefore, when the model estimation is practically impossible using mixture analysis or when the adoption of a model is not plausible, a clustering technique could lead to the discrimination of the labels and to the estimation of their description. For that purpose, we use a hierarchical clustering method [8]. Any other clustering algorithm may be used as well. The clustering is applied on blocks resulting from a systematic division of the image. The blocks are hierarchically clustered using the Bhattacharya distance as a dissimilarity measure.

For continuous variables this measure is defined as

$$d^B = -\ln\left(\int_x \sqrt{p_1(x)p_2(x)}dx\right), \tag{1.9}$$

where p_1 and p_2 are probability density functions of a feature vector x of

any dimension. The discrete version of the Bhattacharya distance is

$$d^B = -\ln\left(\sum_i \sqrt{p_1(i)p_2(i)}\right), \tag{1.10}$$

where $p_j(i)$ is the probability of the i^{th} value of class j. The Bhattacharya distance is strongly linked to the minimum classification error for the two-classes case. In addition, it is a special case of the Chernoff bound of classification error probability [31].

If a model of the distribution is known, a simpler expression of the Bhattacharya distance can be deduced. The extracted features may often be assumed uncorrelated. The simplified expression assuming generalized Gaussian distribution (Equation (1.7)) and uncorrelated features is

$$d^B = \frac{1}{c}\sum_{i=1}^{N-1} \ln \frac{\sigma_{i,1}^c + \sigma_{i,2}^c}{2\sqrt{\sigma_{i,1}^c \sigma_{i,2}^c}}, \tag{1.11}$$

where $\sigma_{i,n}$ corresponds to the standard deviation of the i^{th} feature of class n. When the clustering is complete, the description of the labels is determined. If a parametric model is used, then the parameters are estimated. In the absence of a parametric model, Equation (1.10) is used for computing distances. In both cases the estimation is performed on the clustered blocks.

In order to estimate the feature vectors of the various classes present in the image, a hierarchical clustering algorithm [8] is applied to the blocks. All blocks in the image are used as the initial clusters. Each step of the algorithm merges the pair of clusters having the most similar feature vectors and the features of the new cluster are updated accordingly. The procedure terminates when the number of clusters is reduced to the number of different classes the image to be segmented is known to contain.

3.3 Initial level sets

The next step consists of determining the initial seed regions. An initial map of labeled sites is obtained using statistical tests which classify points with high confidence. The probability of classification error is set to a small value. At first, all pixels are classified according to their distance from the different labels. The distribution of the data in a window centered at each site is approximated. Then, the Bhattacharya distances from this distribution to the features of each label are computed and assigned to the site. The distances at each site are subsequently averaged in a series of windows B_w of dimension $(2w+1) \times (2w+1)$,$(w = 1, \ldots, P)$. The mean distance in each window is used for classifying the central site to one of the possible labels. The candidate label $k(\boldsymbol{s})$ of site $\boldsymbol{s}$ is selected by finding the

label which minimizes the sum of its distances from the neighbor sites p in window B_w,

$$k(\boldsymbol{s}) = \arg\min_{l} \sum_{\boldsymbol{p}\in B_w} d_l^B(\boldsymbol{s}+\boldsymbol{p}). \tag{1.12}$$

The confidence criterion for classification of site $\boldsymbol{s}$ into the candidate label $k(\boldsymbol{s})$ results from comparing the distance of the considered site from the candidate label against the distance from the nearest label among all the others, as described in the following expression,

$$V_{k(\boldsymbol{s})} = \sum_{\boldsymbol{p}\in B_w} \left(\min_{l\neq k(\boldsymbol{s})} d_l^B(k(\boldsymbol{s})+k(\boldsymbol{p})) - d_{k(\boldsymbol{s})}^B(\boldsymbol{s}+\boldsymbol{p}) \right). \tag{1.13}$$

Sites are then sorted according to their confidence measure and a specific percentage of the sites with highest confidence are retained and subsequently labeled. A small percentage is generally sufficient. Sites which are retained for each of these P window sizes are considered as forming the initial sets of labeled points. Parameter P ranges from 3 to 6 in most applications.

3.4 Label propagation

The labels of the initial level sets are then propagated according to the principle presented in Section 2.4. Assuming that the probability density function of the texture images is Gaussian, and given that the high frequency components are zero-mean, the distance of a site $\boldsymbol{s}$ represented by the vector $x(\boldsymbol{s})$ from a texture class j with variances $\sigma_{i,j}^2$ is

$$d_j(x(\boldsymbol{s})) = \frac{1}{2} \sum_{i=1}^{N-1} \left(\frac{x_i^2(\boldsymbol{s})}{\sigma_{i,j}^2} + \log \sigma_{i,j}^2 \right), \tag{1.14}$$

where $N-1$ is the number of high frequency components.

If the probability density functions of the intensity and colour features are also Gaussian, the three corresponding components are added in Equation (1.14). Otherwise, the empirical probability distribution is used for the colour and intensity features,

$$d_j(y_C(\boldsymbol{s})) = -\ln p_j(y_C(\boldsymbol{s})), \tag{1.15}$$

$$d_j(y(\boldsymbol{s})) = -\ln p_j(y(\boldsymbol{s})), \tag{1.16}$$

where $y_C(\boldsymbol{s})$ is the vector of (a,b) colour components and $y(\boldsymbol{s})$ is the intensity.

The multi-label fast marching level set algorithm is then applied to all sets of points initially labeled. The contour of each region propagates according to a velocity field which depends on the label and on the distance

of the considered point from the candidate label. The exact propagation velocity for a given label is

$$F_l(\boldsymbol{s}) = \frac{\Pr(l)}{\sum_{k \neq l} \Pr(k) e^{d_l(x(\boldsymbol{s})) - d_k(x(\boldsymbol{s}))}}, \tag{1.17}$$

where the *a priori* probabilities $Pr(k)$ are estimated from the classification of the sites against the prototype feature classes according to the Bhattacharya distance. The distance of each site from the prototype feature classes is computed using Equations (1.14)–(1.16) with the variances for a considered site being calculated in a window centered at it.

Figure 1 presents results on the *SeaStones* image in which three classes are distinguished. Two of the classes contain similar colour distributions in the *Lab* space, while two classes contain similar textures. As a result the classification is difficult with a significant error probability. In Fig. 1(c) the classification result according to the Bhattacharya distance is shown. In Fig. 1(e) the pointwise classification result according to the *maximum* likelihood criterion is shown. In Fig. 1(b) the initial labeled map with 20% of the image points labeled according to the technique of Section 3.3 is shown. In Fig. 1(d) an intermediate instance of the propagation process where 60% of the image points are labeled is shown. The final segmentation result is considered very satisfactory and is shown in Fig. 1(f).

4 Change detection

Change detection in a video sequence is an important issue in object tracking, video-conferencing and traffic-monitoring among others. The change detection problem consists of labeling each pixel $\boldsymbol{s}$ of one frame t of a video sequence as static ($\Theta(\boldsymbol{s}) = static$) or moving ($\Theta(\boldsymbol{s}) = mobile$).

4.1 Inter-frame difference modeling

In our approach [24, 25, 26], the simple inter-frame grey level difference $x(\boldsymbol{s})$ is considered:

$$x(\boldsymbol{s}) = I(\boldsymbol{s}, t+1) - I(\boldsymbol{s}, t) \tag{1.18}$$

Therefore, a pixel is an *unchanged pixel* if the observed difference $x(\boldsymbol{s})$ supports the hypothesis for static pixel, and a *changed pixel*, if the observed difference supports the alternative hypothesis, for mobile pixel. Let $p_0(x|static)$ (resp. $p_1(x|mobile)$) be the probability density function of the observed inter-frame difference under the respective 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. A zero-mean Laplacian distribution function is used here

to describe the statistical behavior of the pixels under both hypotheses. Thus the conditional probability density function of the observed temporal difference values is given by

$$p(x(\boldsymbol{s})|\Theta(\boldsymbol{s}) = l) = \frac{\lambda_l}{2} \ e^{-\lambda_l |x(\boldsymbol{s})|}. \tag{1.19}$$

If P_{static} (resp. P_{mobile}) be the two *a priori* probabilities, the probability density function of the difference is given by

$$p_X(x) = P_{static} \ p_0 \ (x|static) + P_{mobile} \ p_1(x|mobile). \tag{1.20}$$

In this mixture distribution $\{P_l, \lambda_l : l \in \{static, mobile\}\}$ are the unknown parameters. Mixture analysis aims at determining the *a priori* probabilities of the labels, P_l, and the parameters, λ_l, of the probability density functions of the data given the labels. The most frequently used method for parameter estimation uses the Maximum Likelihood principle, which results in an iterative algorithm [8, 14].

4.2 Level set-based labeling

An initial map of labeled sites is obtained using statistical tests. The initialization process is different from that described in Section 3.3, because *changed* sites should be detected in a point-wise way in order to avoid false alarms which could be propagated, while block-wise decisions are preferable for the *unchanged* initial labels. 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

$$T_1 = \frac{1}{\lambda_0} \ln \frac{1}{P_{FA}}. \tag{1.21}$$

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 six windows of dimension $(2w+1)^2, (w = 2, \ldots, 7)$, are considered and the corresponding thresholds are pre-set as a function of λ_1. Finally the union of the so defined sets determines the initial set of "unchanged" pixels.

The multi-label fast marching level set algorithm is then applied for all sets of points initially labeled. The contour of each region propagates according to a motion field, which depends on the label and on the absolute inter-frame difference.

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. At a pixel level, the decision process is highly noisy, and could be

made more robust by taking into account the known labels in the neighborhood of the considered pixel. This means that the *a priori* probabilities in Equation (1.3) are locally adapted. Finally, the exact propagation velocity for the "unchanged" label is

$$F_0(\boldsymbol{s}) = \frac{1}{1 + e^{\beta_0(|x(\boldsymbol{s})| - n\zeta + \theta_0)}} \tag{1.22}$$

and for the "changed" label

$$F_1(\boldsymbol{s}) = \frac{1}{1 + e^{\beta_1(\theta_1 - |x(\boldsymbol{s})| - (n+\alpha)\zeta)}}, \tag{1.23}$$

where n is the number of the neighboring 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). In the current implementation the parameters are set as follows:

$$\zeta = 0.1T_1, \quad \theta_0 = 4\zeta, \quad \theta_1 = 3.5\left(\zeta + \frac{1}{\lambda_0}\right), \text{ and } \beta_1 = 1.$$

The value of parameter β_0 must take into account the initial label map. If the percentage of the pixels labeled "unchanged" is less than the estimated probability of this label, parameter β_0 is given a value less than 1, because this means that the thresholds used were probably relatively low. Otherwise the value given is greater than 1. In Figure 2 the two speeds are mapped as functions of the absolute inter-frame difference for typical parameter values near the boundary.

Figure 3 shows two different initial labeled maps and the corresponding final labeled images based on the interframe difference between the two first frames of the *Erik* sequence. The background is shown in black, the foreground in white and unlabeled points in gray. We observe that quite different initializations lead to very similar final segmentations because of the label propagation velocity. More results are given in our Web page:

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

5 Conclusion

A new level set-based framework for image segmentation was presented. The Multi-Label Fast Marching algorithm has been introduced for the propagation of high-confidence classification decisions in accordance with the *a posteriori* probability of the competing classes. Two specific segmentation applications are addressed in order to illustrate the usability of this new

algorithm as a fast, precise and generic technique for unsupervised pixel labeling.

This approach's objective of maximizing the *a posteriori* probability is shared by other techniques, such as stochastic or deterministic relaxation. Such approaches utilize an objective function whose global minimum yields the optimal segmentation map. Concerning the algorithm presented here, the absence of a global objective function can be considered as a weakness, yet the dependence of the propagation velocities on the *a posteriori* probabilities of the competing classes clearly leads the segmentation process toward the same goal, while allowing an extremely efficient noniterative implementation.

In comparison with existing level set implementations of active contours, the Multilabel Fast Marching algorithm, as presented here, clearly lacks the capability of incorporating a smoothness constraint into the propagation process, thus sometimes resulting in noisy expansion of the moving contours. Nevertheless the competition of the expanding regions and their convergence over each other significantly reduce the amount of noise in the final region boundaries, while retaining a high level of localization precision. In addition, our algorithm can handle multiple segmentation classes with a complexity that outperforms most existing multiclass level set methods.

Inherent limitations and shortcomings of the proposed framework include the strong dependence of the segmentation quality on the extracted features, which must be both sufficient and discriminant. Additionally, the initial high-confidence classification decisions impose strict constraints on the morphology of the final regions. Subsequently, narrow or small and isolated parts of a given class exhibit an inherent difficulty of detection.

6 References

[1] R. Adams and L. Bischof. Seeded region growing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-16:641–647, June 1994.

[2] J. Besag. On the statistical analysis of dirty images. *Journal of Royal Statistics Society*, 48:259–302, 1986.

[3] A. Blake and M. Isard. *Active contours*. Springer, 1998.

[4] J. F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8:679–698, 1986.

[5] T. Chan and L. Vese. An active contour model without edges. In *Proc. 2nd Int. Conf. on Scale-Space Theories in Computer Vision*, pages 141–151, Corfou, Greece, 1999.

[6] P. Chou and C. Brown. The theory and practice of bayesian image labeling. *International Journal of Computer Vision*, 4:185–210, 1990.

[7] L. Cohen. On active contours models and balloons. *CVGIP: Image Understanding*, 53:211–218, Mar. 1991.

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

[9] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis Machine Inteligence*, 6:721–741, 1984.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *Intern. Jour. of Computer Vision*, 1:321–332, Jan. 1988.

[11] S. Liapis, E. Sifakis, and G. Tziritas. Color and texture segmentation using deterministic relaxation and fast marching algorithms. In *Proc. Intern. Conf. on Pattern Recognition*, pages 621–624, Barcelone, 2000.

[12] S. Liapis, E. Sifakis, and G. Tziritas. Colour and texture segmentation using wavelet frame analysis, deterministic relaxation and fast marching algorithms. *J. of Visual Communication and Image Representation*, (accepted).

[13] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-17:158–175, Feb. 1995.

[14] G. McLachlan, D. Peel, and W. Whiten. Maximum likelihood clustering via normal mixture model. *Signal Processing: Image Communication*, 8:105–111, 1996.

[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 regions: A new framework to deal with frame partition problems in computer vision. *J. of Visual Communication and Image Representation*, 13:249–268, March/June 2002.

[17] T. Pavlidis. *Algorithms for Graphics and Image Processing.* Computer Science Press, Rockvill MD, 1982.

[18] P. Salembier. Overview of the mpeg-7 standard and of future challenges for visual information analysis. *EURASIP Journal on Applied Signal Processing*, pages 343–353, April 2002.

[19] C. Samson, L. Blanc-Feraud, G. Aubert, and J. Zerubia. A level set model for image classification. In *Proc. 2nd Int. Conf. on Scale-Space Theories in Computer Vision*, pages 306–317, Corfou, Greece, 1999.

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

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

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

[23] E. Sifakis, C. Garcia, and G. Tziritas. Bayesian level sets for image segmentation. *J. of Visual Communication and Image Representation*, 13:44–64, March/June 2002.

[24] E. Sifakis, I. Grinias, and G. Tziritas. Video segmentation using fast marching and region growing algorithms. *EURASIP Journal on Applied Signal Processing*, pages 379–388, April 2002.

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

[26] E. Sifakis and G. Tziritas. Moving object localisation using a multi-label fast marching algorithm. *Signal Processing: Image Communication*, 16:963–976, Aug. 2001.

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

[28] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. on Image Processing*, IP-4:1549–1560, Nov. 1995.

[29] R. Kimmel V. Caselles and G. Sapiro. Geodesic active contours. *Int. Jour. of Computer Vision*, pages 61–79, 1997.

[30] A. Yezzi, A. Tsai, and A. Willsky. A fully global approach to image segmentation via coupled curve evolution. *J. of Visual Communication and Image Representation*, 13:195–216, March/June 2002.

[31] T. Young and K. S. Fu. *Handbook of pattern recognition and image processing*. Acedemic Press, 1986.

[32] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes / mdl for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-18:884–900, Sept. 1996.

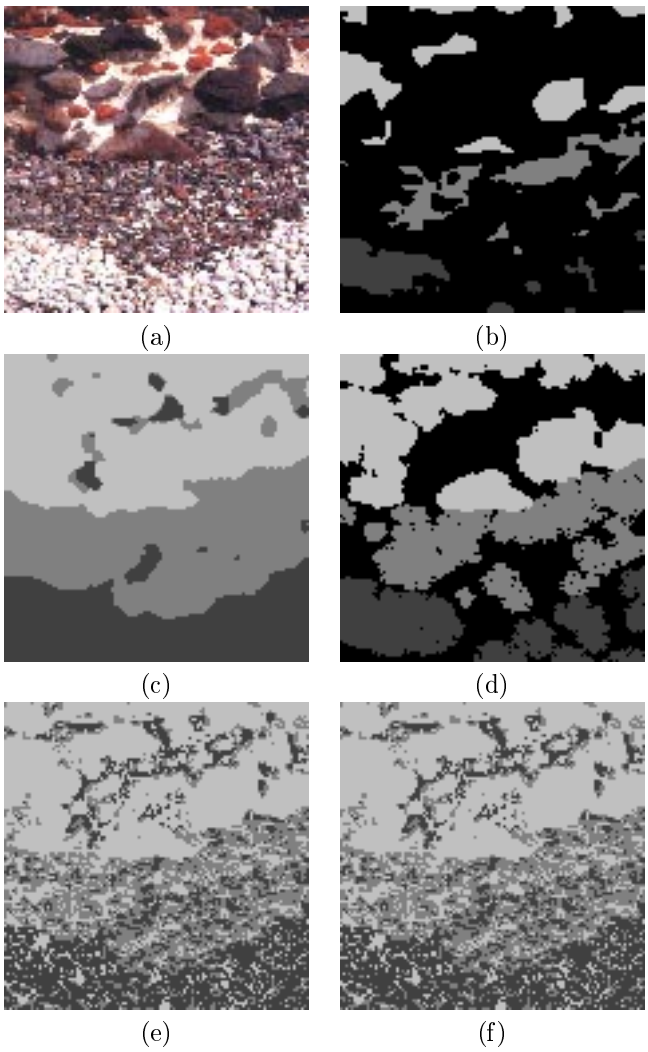

FIGURE 1. The segmentation result on the *SeaStones* image (a).

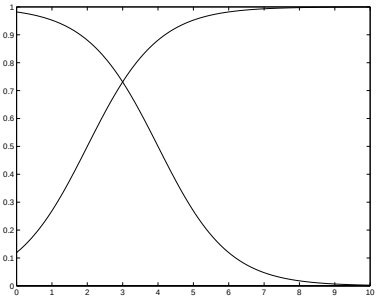

FIGURE 2. The propagation speeds of the two labels.

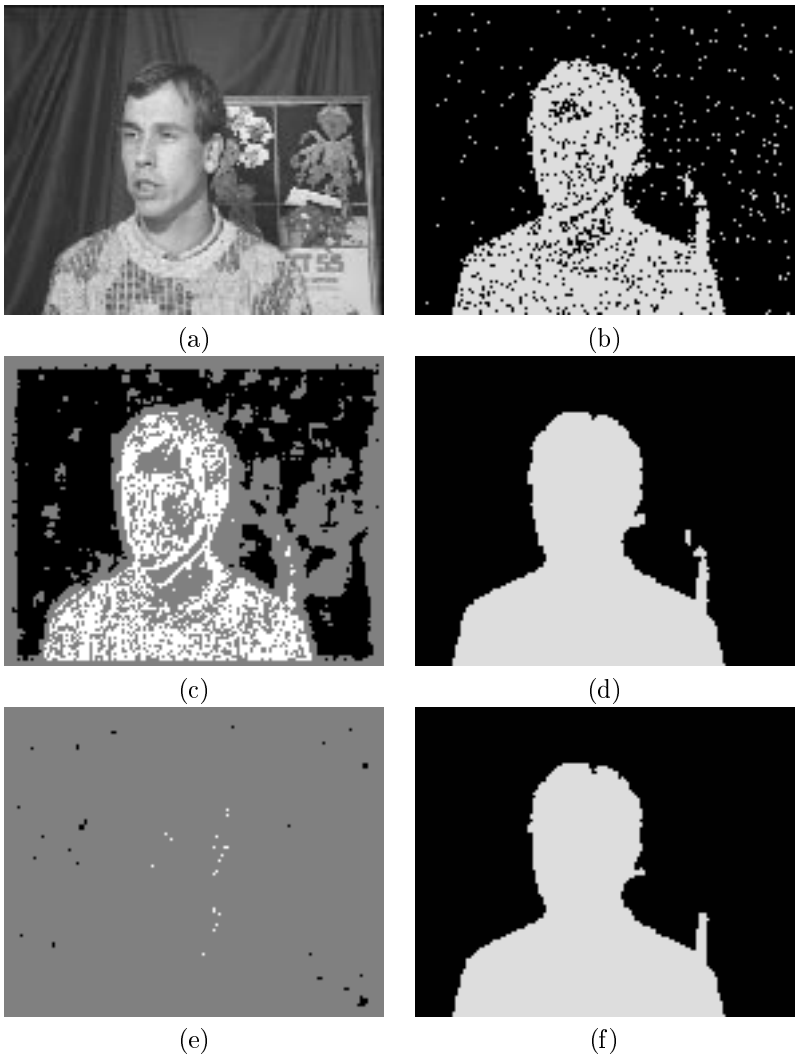

FIGURE 3. The segmentation result on the *Erik* sequence: (a) the first frame of the sequence, (b) the *maximum a posteriori* probability classification, (c) initialization of a large number of sites, (d) the corresponding segmentation, (e) initialization of a small number of sites, and (f) the corresponding segmentation.