Bayesian Level Sets for Image Segmentation

Eftichis Sifakis, Christophe Garcia and Georgios Tziritas

Department of Computer Science, University of Crete, P.O. Box 2208, Heraklion, Greece; and Institute of Computer Science - FORTH, P.O. Box 1385, Heraklion, Greece

E-mail: sifakis@csd.uoc.gr; cgarcia@csi.forth.gr; tziritas@csd.uoc.gr

Received ; accepted

This paper presents a new general framework for image segmentation. A level set formulation is used to model the boundaries of the image regions and a new Multi-Label Fast Marching is introduced for the evolution of the region contours towards the segmentation result. Statistical tests are performed to yield an initial estimate of high-confidence subsets of the image regions. Furthermore, the velocities for the propagation of the region contours are defined in accordance with the *a posteriori* probability of the respective regions, leading to the **Bayesian Level Set** methodology described in this paper. Typical segmentation problems are considered and experimental results are given to illustrate the robustness of the method against noise and its performance in precise region boundary localization.

1. INTRODUCTION

Image segmentation is a vital component of any system of image content analysis. In the MPEG-4 standard, segmentation plays an important role in coding performance and object manipulation [23]. 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 is located at 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 advance has been obtained 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 firstly introduced. The labeling problem is 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 region-based approaches. Zhu and Yuille [28] proposed a region competition method which combines the geometrical features of snakes/balloon models and the statistical techniques of region growing. In [16], the concept of geodesic active regions is introduced. 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 [15, 13] has raised many of the inconveniences of active contours, while the fast marching algorithm [19, 21] provides a computationally efficient method for the tracking of a monotonically evolving contour. Natural handling of morphological transitions and the possibility of motivating the evolution of the moving front on intrinsic curve properties give rise to several applications, as extensively presented in [20]. The introduction of the geodesic active contours [25] 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, 26] level set formulations have been 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. In addition, the latter suggests a generalization for the case of more than two segmentation classes, although the adopted formulation inflicts dimensionality constraints over the input features to allow for such extensions. Furthermore, in [18] the segmentation of an arbitrary number of classes is addressed, using a system of coupled partial differential equations, leading to the combined evolution of several level set modeled contours. The last approach is based on the minimization of a functional which enforces region uniformity, contour smoothing and classification coherence.

In this paper, we introduce a new region-based methodology for image segmentation, where statistical approaches are applied for modeling the different regions and a novel level set based algorithm is used for labeling through the evolution of the region boundaries. Since different simultaneously propagating region 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 capacity 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 suitable models. Pattern analysis techniques are used for the identification of the corresponding models.

This paper is organized as follows. In Section 2, we review the level set formulation and the fast marching algorithm and we describe how we extend the latter to the multi-label case. In Section 3, we consider the very important problem of automatic feature extraction for the description of region content, as well as the initialization of the level sets. In Section 4, we consider different problems of image segmentation, in order to illustrate the general applicability of our framework. We first consider the classical case of level sets driven by the image gradient. We show how a multi-label propagation, i.e, the propagation of two contours in opposite directions, determines the actual location of the region boundaries. Then, we consider the case of luminance segmentation, where any difference in the probability distribution of the corresponding signal provides features for content description. In that case, a model-based approach, using Gaussian assumptions and an approach based on histogram distribution are provided. The application of our method to the segmentation of a field of two chromaticity components is then considered and described by a two-dimensional probability distribution. When this description appears to be insufficient, segmentation based on texture features is considered. The Discrete Wavelet Analysis is performed for the description of the texture content. Another interesting case arises in motion detection, which is a powerful feature for object segmentation. The detection of temporal changes of the luminance signal is performed. The segmentation of the two-dimensional motion field is finally considered using a Gaussian model. Various experimental results illustrate each of the segmentation problems. Finally, conclusions are drawn.

2. LEVEL SET THEORY AND ALGORITHMS

2.1. Fundamental level set formulations

Level set theory provides a framework for tracking the evolution of a closed curve of the plane given the velocity of the curve along its normal direction. The parametric representation of the curve $\mathcal{C}(t)$ used in some other methods is unsuitable for many applications since morphological changes of the moving contour, such as splitting and merging, are extremely difficult, if not impossible, to manipulate.

In the pioneering work by Osher and Sethian [15] the various locations of the evolving contour are embedded as level sets of a function of higher dimensionality. Consider a $N-1$ dimensional hyper-surface $\Gamma(t)$ and let $\phi(\boldsymbol{s}, t=0)$, where $\boldsymbol{s}$ is a point in $\mathcal{R}^N$, be defined by

$$\phi(\boldsymbol{s}, t=0) = \pm d \tag{1}$$

where d is the distance from $\boldsymbol{s}$ to $\Gamma(t=0)$ and the the sign is chosen according to the point $\boldsymbol{s}$ being outside or inside the initial hyper-surface $\Gamma(t=0)$. Thus the initial location of the moving contour is given by

$$\Gamma(t=0) = (\boldsymbol{s} | \phi(\boldsymbol{s}, t=0) = 0). \tag{2}$$

The derivations described in [15] yield the time-dependent level set equation for the embedding function ϕ

$$\phi_t + F|\nabla\phi| = 0 \tag{3}$$

where F is the propagation velocity in the normal direction to the moving contour and the initial conditions are given by Equation (1). Explicit finite differences may be used in the numerical solution of the above equation using forward differences for the time derivative and a suitable approximation for the spatial gradient.

In the above formulation the velocity function F is free to include terms dependent on the curvature or the outward normal to the moving contour, measures that are easily expressed by means of the evolving function ϕ. Moreover, in [19], an extremely computationally efficient variant of the above method was proposed. Given the limitation of a constantly positive (or constantly negative) velocity function F, 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 the above formulation we have that $\phi(\boldsymbol{s}, T(\boldsymbol{s}) = 0)$ and the location of the moving contour at time t_0 is given by

$$\Gamma(t = t_0) = (\boldsymbol{s} | T(\boldsymbol{s}) = t_0). \tag{4}$$

The arrival time function T satisfies the stationary level set equation

$$F|\nabla T| = 1 \tag{5}$$

which simply states that the gradient of the arrival time function is inversely proportional to the velocity of the contour at any given point. The preceding formulation allows the constructive calculation of the arrival time function T, without resorting to iterative methods. The tradeoff 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, introduced in [19], constructs a solution to Equation (5) from initial data with a $n \log n$ execution cost.

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 [20].

The methods mentioned apply to the time-dependent level set formulation, based on Equation 3, allowing for the tracking of the moving region interfaces against time. The motive of the work presented herein is that a substantial number of applications could be covered by the tracking of the monotonical evolution of distinct regions into a special *blank* or *unlabeled* region and the observation of the final result of the initial regions' convergence over each other. The input to this approach would be constituted 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 an initialization of the arrival time map with seed regions, the calculation of arrival times for the rest of the spatial domain considered and either an explicit selection of a proper level set or the utilization of 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 a hard segmentation solution while discarding 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 in this paper. 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. The Multi-Label Fast Marching algorithm

The new Multi-Label Fast Marching algorithm presented in this paper is an extension of the well-known Fast Marching algorithm introduced by Sethian [19], capable of manipulating in parallel multiple propagating contours that correspond to the boundaries of competitively expanding regions. The computational complexity 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 limitations of the original Fast Marching algorithm, namely the requirement of a constant sign velocity and the absence of a contour smoothness term, are also present in the Multi-Label variant. The new algorithm targets applications of 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 (5) 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 limitation of one region being unable to infiltrate a region having 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 this presentation we shall limit the description of the new algorithm to the case of a two-dimensional image, although the algorithm trivially expands 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 in a specific pixel lacking a finalized candidacy when a neighboring pixel acquires an alive 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 comparative symbolic description of the classical Fast Marching algorithm [19] and the Multi-Label extension introduced herein can be seen in figure 1.

```
InitTValues()
InitNarrowBand()
while (ExistTrialPixels()) {
  pxl = FindLeastTValue()
  MarkPixelAlive(pxl)
  AddFarawayNeighbors(pxl)
  UpdateNeighbourTValues(pxl)
}
```

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

FIG. 1. Pseudo-code for the single and multi label fast marching algorithms

The Multi-Label Fast Marching algorithm is supplied with a label map partially filled with decisions. The arrival time for the initially labeled pixels is set to zero, while for all others it is set to infinity. A map of pointers to linked lists of candidacies is also maintained. Candidacy lists are initially empty, with the exception of unlabeled pixels being next 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 also contained in a common priority queue.

Until no more trial candidacies exist, the trial candidacy with the smallest arrival time is selected and turned 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 (5). Under the most common gradient approximation used, the equation is written

$$1/F_{ij}^2 = \max(\max(D_{ij}^{-x}T, 0), -\min(D_{ij}^{+x}T, 0))^2 + \max(\max(D_{ij}^{-y}T, 0), -\min(D_{ij}^{+y}T, 0))^2. \quad (6)$$

Equation (5) 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, bringing 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 label candidacies per pixel of two at most. 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 for 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 maximum *a posteriori* probability principle. 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 s, for a candidate label $l(s)$ and for a data vector $x(s)$ the propagation speed as

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

Then we can write

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

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 or assumed all equal.

Very often 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

$$F_l(\boldsymbol{s}) = \frac{1}{1+\sum_{k\neq l} e^{d_l(\boldsymbol{s})-d_k(\boldsymbol{s})}}. \tag{8}$$

This expression of the propagation speed illustrates that when the propagated label is the right one, all the exponents of the sum are negative and the speed is close to unity. In the opposite case, when the propagated label is not correct, at least one exponent is positive and, therefore, the speed is biased towards zero.

For measuring the difference of 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 (say, without loss of generality, 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

$$\frac{E\{|\nabla T(\boldsymbol{s})|;1\}}{E\{|\nabla T(\boldsymbol{s})|;0\}} = 1 + \frac{D(p(x|1), p(x|0))}{2} > 1. \tag{9}$$

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

For illustrating 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

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

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

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. SEED REGIONS AND FEATURE EXTRACTION

An essential step of the whole framework consists of estimating the features associated to the different labels and in determining the initial seed regions. The only assumption we make is that the number of labels is known. We describe hereafter the stages of automatic feature extraction and of determination of the zero level sets.

3.1. Automatic feature extraction

As the number of labels is assumed to be known, two approaches are possible for their discrimination: mixture analysis and unsupervised clustering. In section 3 presenting image segmentation applications, both approaches are used according to the particular problem to be solved.

If the adoption of a model is reliable and, in particular, if this model is tractable, the mixture analysis is preferred. This is often the case of luminance segmentation

where a Gaussian distribution is plausible. This is also the case for change detection where a generalized Gaussian or a Laplacian model are often adopted. Furthermore, the motion vectors resulting from optical flow computation are assumed to be Gaussian variables. The distribution mixture is expressed as the weighted sum of the individual distributions of the different labels. Mixture analysis aims at determining the *a priori* probabilities of the labels and the parameters 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]).

Nevertheless, there exist interesting applications where the use of an *a priori* model might be difficult, if not arbitrary. For example, in texture segmentation, after a multi-channel analysis, the Gaussian model is plausible but it would be difficult to obtain accurate parameters using a mixture analysis because the distributions may be all zero-mean (see section 4.4). In addition, there are cases where no general model is applicable and luminance or chromaticity only may be sufficient for the segmentation. 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, 12]). 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.

The continuous definition of this measure is

$$d^B = -\ln(\int_x \sqrt{p_1(x)p_2(x)}dx) \tag{10}$$

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

$$d^B = -\ln(\sum_i \sqrt{p_1(i)p_2(i)}) \tag{11}$$

where $p_j(i)$ is the probability of i^{th} feature of class j. This measure has the advantage of being designed to compare features for the two-class case. In addition, it corresponds to the Chernoff bound of error probability [27].

If a model of the distribution is known, a simpler expression of the Bhattacharya distance can be deduced. In this work, we assume that some features follow the generalized Gaussian distribution

$$p(x) = \frac{c}{2\sigma\,\Gamma(\frac{1}{c})}\,e^{-\left(\frac{|x|}{\sigma}\right)^c} \tag{12}$$

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. The extracted features may often be assumed uncorrelated. The simplified expression assuming Gaussian distribution

and uncorrelated features is

$$d^B = \frac{1}{c}\sum_{i=1}^{N} \ln \frac{\sigma_{i,1}^c + \sigma_{i,2}^c}{2\sqrt{\sigma_{i,1}^c \sigma_{i,2}^c}} \tag{13}$$

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 absence of a parametric model, equation (11) 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 different classes present in the image, a hierarchical clustering algorithm [8] is applied to the blocks. All the different blocks in the image are used as the initial clusters. Each step of the algorithm, merges the two clusters with the nearest feature vectors and the features of the new cluster are updated accordingly. The procedure terminates when the number of clusters becomes equal to the number of the different classes in the image to be segmented.

3.2. Initial level sets

An initial map of labeled sites is obtained using statistical tests. These tests 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 D_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 $k(s)$ of site s is selected in the following way

$$k(s) = \arg\min_{l} \sum_{p \in D_w} d_l^B(s+p) \tag{14}$$

by finding the label l which minimizes the sum of its distances from the neighbor sites p in window D_w.

The confidence criterion for classification of site s into the candidate label $k(s)$ results from the comparison of the distance of the considered site from the candidate label against the distance from the nearest label among all the others, as described in Equation 15.

$$\sum_{p \in D_w} \left(\min_{l \neq k(s)} d_l^B(s+p) - d_{k(s)}^B(s+p) \right) \tag{15}$$

Sites are then sorted according to their confidence measure and a specific percentage of the sites with highest confidence are retained and labeled. A small percentage, a typical value being 5%, 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.

4. SEGMENTATION

In this section, we describe the application of our general framework to typical segmentation problems, such as luminance, chromaticity, texture and motion segmentation. In each case, statistical approaches are used for obtaining an initial high-confidence set of pixel labels constituting the initial regions. The propagation velocities for the expanding regions are defined according to the considered segmentation problem. The multi-label fast marching level set algorithm is subsequently applied to give the segmentation solution.

4.1. Gradient-based segmentation

Boundary-based segmentation methods often rely on the gradient of the image, where local maxima are expected to define object boundaries. It should be noted that the gradient may be computed from intensity images, color images or more generally from multi-dimensional images. In some applications where no specific descriptors of object region are available, users may roughly define the area of interest, by defining two contours, one being contained inside the object (referred to as *inner contour*) and the other one containing the object (referred to as *outer contour*). A labeled map is therefore built: pixels belonging to the inner area are assigned to one label and pixels belonging to the outer area are assigned to another label. The algorithm propagates the user initialized two contours, in opposite directions, the inner outside and the outer inside, towards locations demonstrating high spatial contrast, which correspond to the edges of the object to be segmented. The algorithm uses a velocity field calculated from the gradient image to march the contours in the allowed direction, and build the arrival map of each contour. The object boundary is recovered as the place of contact of the two contours. It therefore provides an automatic stopping criterion for the propagation since the two contours converge over each other. The propagation speed of the evolving contours

FIG. 2. Gradient, initial contours and several propagation instances for the *Olaf* image.

should be maximum for zero-value gradient and close to zero for important gradient values. A suitable propagation speed F, being a generalization of the one used in [20], has been chosen for each contour.

$$F = \frac{1}{1 + \left(\frac{(|\nabla I|)}{\theta}\right)^{\gamma}} \tag{16}$$

The parameter θ may be interpreted as a threshold, defining the lowest gradient value that may influence the contour propagation. The parameter γ may be considered as controlling the sensibility of the front propagation with regards to normalized gradient values. Figure 2 illustrates the gradient-based results of the multi-label fast marching algorithm applied on the color image *Olaf*, where the gra-

dient magnitude image computed in *Lab* color space, the original image with the initial inner and outer contours, some intermediate instances of propagation and the final segmented image, are presented.

4.2. Luminance segmentation

A grey scale image may be segmented using its intensity value distribution. Each region content may be described by either modeling the luminance distribution, using for instance a Gauss probability density function, or estimating the empirical probability density function. Both approaches are considered here.

In the *a priori* model-based approach, without loss of generality and for purpose of illustration, we consider the case of two labels $l \in \{0,1\}$, with respective distributions assumed to be Gaussian with the same variance (σ^2), and two different mean values (μ_0 and μ_1). The propagation speed according to Equation (7) is

$$F_l(s) = \frac{1}{1+\exp\left(\frac{(\mu_k-\mu_l)(2x(s)-\mu_0-\mu_1)}{2\sigma^2}\right)} \qquad k \neq l, k \in \{0,1\}, l \in \{0,1\}, \quad (17)$$

in the case of equiprobable labels. It may be noticed that both propagation speeds depend on the signal to noise ratio ($\frac{|\mu_0-\mu_1|}{\sigma}$). They are drawn in Figure 3 for a signal to noise ratio equal to one, as functions of the normalized distance ($\frac{x(s)-(\mu_0+\mu_1)/2}{\sigma}$) to the global mean value. A synthetic example, built to illustrate the efficiency

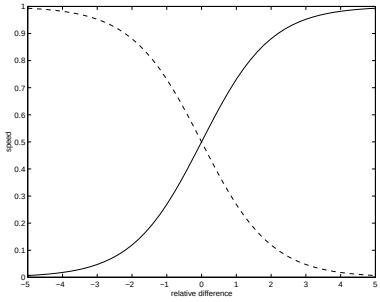

FIG. 3. The propagation speeds of the two labels for a signal to noise ratio equal to one (dashed line for label 0 and solid line for label 1)

of our approach, is presented in Figure 4, where a light disk is drawn on a darker background. Pixels of the disk follow a Gaussian distribution of mean $\mu_0 = 170$ and standard deviation $\sigma = 30$, while pixels of the background follow a Gaussian distribution of mean $\mu_1 = 140$ and standard deviation $\sigma = 30$. The signal to noise ratio is 1, corresponding to a very poor image quality. From top-left to bottom-right, Figure 4 shows the original noisy image, the initial label map, some instances of the propagation and the final segmentation result. One may notice that, even in a case of a very noisy image, with a few initial labeled sites, the algorithm performs very well.

In the general case where no *a priori* model is known, only luminance histograms are used for segmentation, with $h_l(i)$ being the histogram concerning the label l. The label is propagated with the following speed:

$$F_l(s) = \frac{1}{1+\sum_{k\neq l}\frac{h_k(x(s))}{h_l(x(s))}}. \quad (18)$$

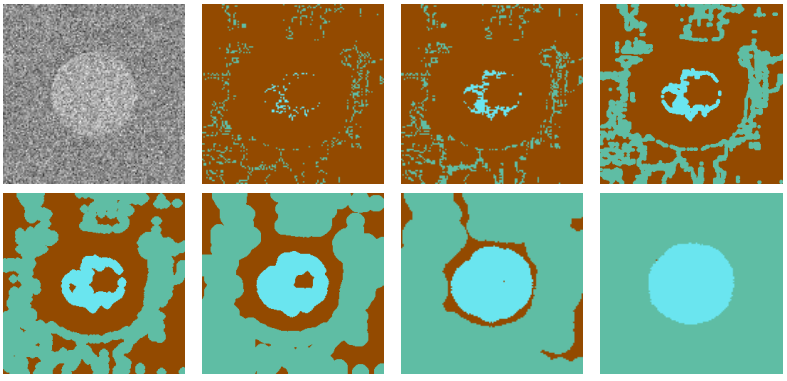

FIG. 4. Luminance-based segmentation of the *disk* image

The luminance histogram-based segmentation of the real textured image *zebra* is presented in Figure 5, where two classes (the zebra and the grass) are considered, leading to a segmentation with precise border estimation.

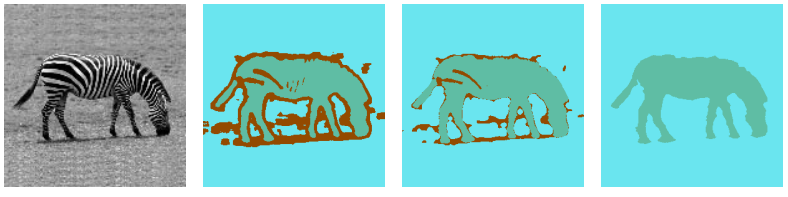

FIG. 5. Luminance histogram-based segmentation of the *zebra* image

4.3. Chromaticity segmentation

As the luminance component is considered in the previous paragraph, only chromaticity components are taken into account here. The *Lab* color space is used for feature extraction. In our approach, the extracted features are the local 2-D histograms of the (a, b) components in a squared block centered at each site. No modeling of the distribution of the (a, b) histograms is performed (*e.g.* Gauss), because in most blocks, histograms do not follow the same distribution. The propagation speed is the same as in the luminance histogram-based case (see (18)) .

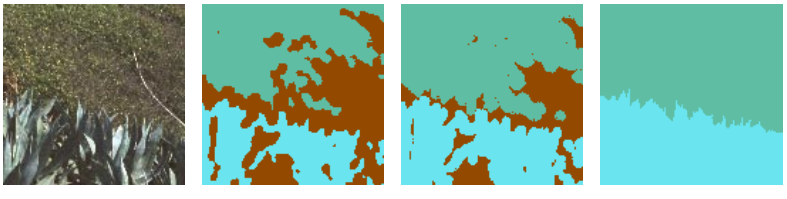

FIG. 6. Chromaticity histogram-based segmentation of the *GrassPlants* image

Figure 6 presents our results on the natural scene of *GrassPlants* (MIT Media Vistex data set), with some intermediate instances of propagation and the final segmented image. In this case, the approach performs well by using only the histograms of the chromaticity components (a, b).

4.4. Texture segmentation

Segmenting images according to textural information may overcome some limitations of the previous luminance or chromaticity-based approaches. Our method for texture segmentation is based on the Discrete Wavelet Analysis for describing the texture content, as presented in detail in [12].

4.4.1. Texture analysis and characterization

The features for texture segmentation are derived from the discrete wavelet frames analysis [24]. The input signal is split into components corresponding to disjoint frequency ranges via recursive bisection of the low-frequency component. The resulting wavelet frames carry important textural characteristics of the original signal such as periodicity and translation invariance, with respect to different scales. The formulation can trivially be extended to 2-D by forming the corresponding filter bank from the 1-D filters using separability. The application of the 2-D filter bank on a given image yields three high-frequency *detail* components per each analysis level plus the low-frequency *approximation* component at the last level. All components can be shown to be uncorrelated in the case of ideal filters.

The chosen feature vectors for texture description are the variances of the $N-1$ high frequency components and the mean of the low frequency approximation, calculated over the distinct texture classes present in the image. The low frequency approximation is used only if it is sufficiently discriminating. An initial estimate of the texture classes is obtained by the hierarchical clustering algorithm described in section 3.1. The block sets obtained by the algorithm are used for the calculation of the prototype feature vectors, used in initial labeling and label propagation.

4.4.2. Label propagation

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 s represented by the vector $x(s)$ from a texture class l with variances $\sigma_{i,l}^2$ and mean value μ_l of the approximation component is determined as follows:

$$d_l(x(s)) = \frac{1}{2}\left(\sum_{i=1}^{N-1}\left(\frac{x_i^2(s)}{\sigma_{i,l}^2} + \log\sigma_{i,l}^2\right) + \frac{(x_N(s)-\mu_l)^2}{\sigma_{N,l}^2} + \log\sigma_{N,l}^2\right) \tag{19}$$

where $\sigma_{N,l}^2$ is the variance of the low frequency approximation of class l.

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

$$F_l(s) = \frac{\Pr(l)}{\sum_{k\neq l}\Pr(k)e^{d_l(x(s))-d_k(x(s))}} \tag{20}$$

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

The top row of Figure 7 shows the original *four-regions* image, the block-set image for prototype vector calculation and the preliminary pixel classification images according to the Bhattacharya distance and the maximum likelihood. The bottom row shows some intermediate instances of propagation and the final segmented image. The percentage of classification error has been found to be about 2.1%.

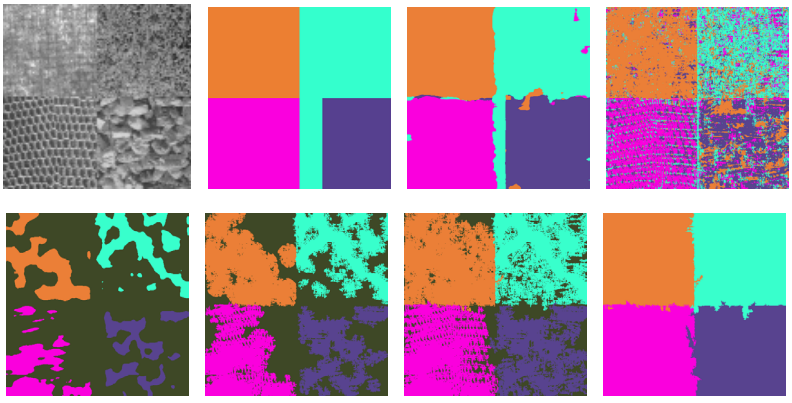

FIG. 7. Texture segmentation of the *four-regions* image

Figure 8 illustrates the results obtained on the natural scene *GrassPlants* (MIT Media Vistex data set), showing the original image, the block-set image, the initially labeled image, and the final segmented image.

FIG. 8. Texture segmentation of the *GrassPlants* image

4.5. Change detection

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

4.5.1. Problem modeling

In our approach, the simple inter-frame grey level difference $x(s)$ is considered:

$$x(s) = I(s, t+1) - I(s, t). \tag{21}$$

Therefore, a pixel is an *unchanged pixel* if the observed difference $x(s)$ 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

$$p(x(s)|\Theta(s) = l) = \frac{\lambda_l}{2} \ e^{-\lambda_l |x(s)|}. \tag{22}$$

If P_{static} and P_{mobile} are the two *a priori* probabilities, the probability density function of the difference is given by

$$p_X(x) = P_{static} \ p_0 \ (x|static) + P_{mobile} \ p_1(x|mobile). \tag{23}$$

In this mixture distribution, P_l and λ_l ($l \in \{static, mobile\}$) are unknown parameters. As stated in section 3.1, the principle of Maximum Likelihood is used to obtain an estimate of these parameters.

4.5.2. Initial labeling

An initial map of labeled sites is obtained using statistical tests. The first test detects changed sites with high confidence, using a threshold T_1. Then, tests are performed for finding unchanged sites with high confidence, that is, with small probability of non-detection. For these tests, a series of six windows of dimension $(2w+1)^2, (w = 1, \ldots, 6)$, are considered and the corresponding thresholds are predefined as a function of λ_1. The level set initialization is given in Figure 10 for one couple of frame of the *White Trevor* image sequence, where "black" color means "unchanged" site, "white" color means "changed" site, and "grey" color means "unlabeled" site.

4.5.3. Label propagation

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 velocity 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 scale, the decision process is highly noisy, and could be made robust by taking into account the known labels in the neighborhood of the considered pixel. Finally, the exact propagation velocity for the "unchanged" label is

$$F_0(s) = \frac{1}{1 + e^{\beta_0(|x(s)| - n\zeta - \theta_0)}} \tag{24}$$

and for the "changed" label

$$F_1(s) = \frac{1}{1 + e^{\beta_1(\theta_1 - |x(s)| - (n+\alpha)\zeta)}} \tag{25}$$

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, otherwise 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). More details about the setting of these parameters are given in [22].

Figure 9 presents the two speeds as functions of the absolute inter-frame difference for typical parameter values. Figure 10 shows the initial labeled map, some

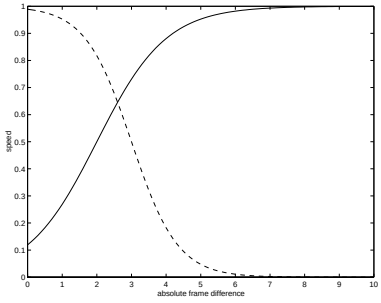

FIG. 9. The propagation speeds of the two labels (dashed line for label *static* and solid line for label *mobile*)

instances of propagation and the final labeled image.

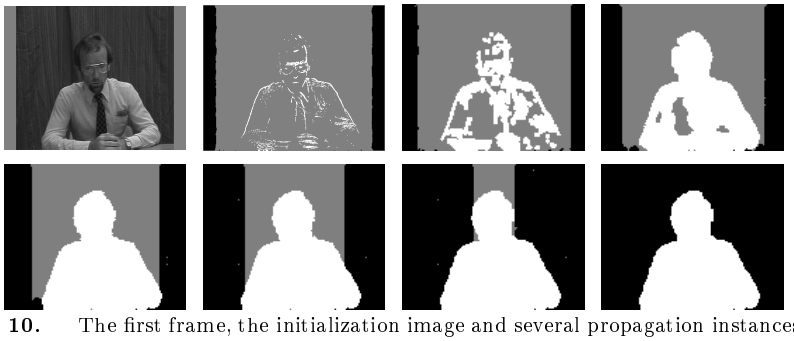

FIG. 10. The first frame, the initialization image and several propagation instances for two frames of the *White Trevor* sequence

4.6. Motion field segmentation

The segmentation of motion fields in video sequences is another interesting way of extracting scene content. In the case of a static camera or after camera motion compensation, segmenting motion fields leads to mobile objects extraction. In the case of a moving camera, it may provide an approximation of the 3D structure of a static scene, by extracting depth layers. In our approach, some restrictions are made on the camera motion, for segmenting mobile objects (considered as planar), or constant depth layers from motion fields. The motion field components u and v are constant, for a constant depth or for a moving surface, if the camera motion is assumed to be translational only, in a direction parallel to the image plane [11]. In that particular case, finding depth layers or moving objects is equivalent to segment the motion field into regions of constant values. The task is made difficult by the important noise contained in the computed optical flow, which is a crude approximation of the real motion field. As stated in 3.1, we assume that components u and v of the optical flow can be decomposed into mixtures of n Gaussian distributions.

The parameters of the Gaussian distribution are determined by analysis of the 2-D optical flow histogram. The Multi-Label Fast Marching level set algorithm, is then

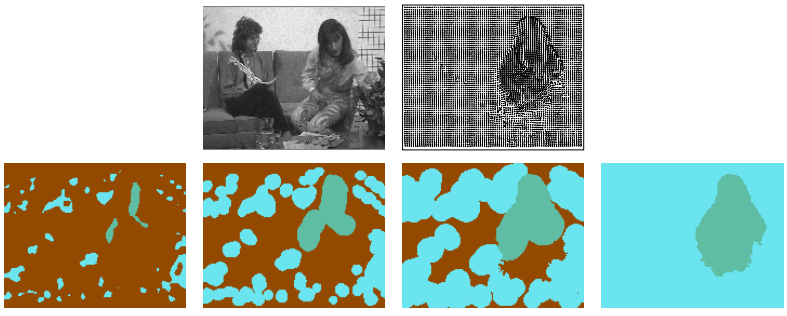

FIG. 11. Mobile object segmentation from the *interview* optical flow.

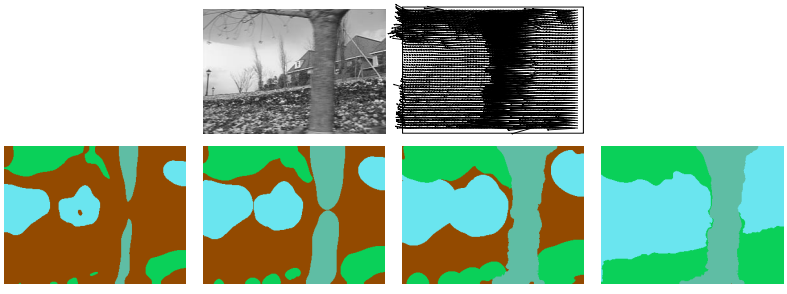

FIG. 12. Depth-layer segmentation from the *flower-garden* optical flow.

applied for all sets of points initially labeled. The propagation velocity for a given label is the one given in equation (8), assuming the *a priori* probability to be equal to the distance from the site s to class l, which is defined by

$$d_l(s) = \frac{(u(s) - u(l))^2}{2\sigma_{1l}^2} + \frac{1}{2}\ln\sigma_{1l}^2 + \frac{(v(s) - v(l))^2}{2\sigma_{2l}^2} + \frac{1}{2}\ln\sigma_{2l}^2 \tag{26}$$

where σ_{il} is the standard deviation of i^{th} feature in class l. Figure 11 illustrates the segmentation results of the multi-label fast marching algorithm on an optical flow image extracted from the *interview* sequence. In this part of the sequence, the interviewed person is the only moving entity. Even if parts of the body exhibit different motions, our goal is to segment the moving section of the image as a whole. The first row of Figure 11 presents an image of the sequence, the computed optical flow, whereas the second row shows the initial label map, some intermediate instances of propagation and the final segmented image, containing the segmented moving object. An example of depth layering is also considered in the difficult case of the *flower-garden* sequence. Before applying the algorithm to this image sequence, the existence of a "panning" motion was confirmed through the use of the motion estimation algorithm presented in [11]. It should be noted that the lack of texture in the sky part leads to the erroneous estimation of optical flow components. The first row of Figure 12 presents an image of the sequence, the computed optical flow, whereas the second row shows the initial label map, some

intermediate instances of propagation and the final segmented image, containing three depth layers.

5. CONCLUSION

A new level set based framework for image segmentation was presented in this paper. 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. Several specific segmentation applications are addressed in order to illustrate the usability of this new algorithm as a fast, precise and generic technique for unsupervised segmentation or labeling.

The proposed approach shares the objective of maximization of the *a posteriori* probability with other techniques, like deterministic relaxation. Such approaches utilize the notion of an objective function whose global minimum yields the optimal segmentation solution and guarantee spatial coherence by the consideration of pixel neighborhoods or cliques, over which a certain degree of uniformity is enforced. The absence of an objective function can be considered as a theoretical weakness of the proposed algorithm, yet the dependence of the propagation velocities on the *a posteriori* probabilities of the competing regions clearly biases the segmentation result towards the same goal, while allowing for an extremely efficient non-iterative implementation. Furthermore, deterministic relaxation algorithms are often trapped in local minima while the proposed methodology has been shown to be highly robust against locally optimal solutions. A comparative survey of the Multi-Label Fast Marching algorithm and the deterministic relaxation ICM algorithm for labeling can be found in [12].

In comparison with existing level set implementations of active contours the Multi-Label Fast Marching algorithm clearly lacks the capability of incorporating a smoothness constraint into the propagation process, thus often resulting to 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, the new level set algorithm can handle multiple segmentation classes with an algorithmic complexity that outperforms most existing multi-class 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 discriminative. Additionally, the initial high confidence classification decisions impose strict constraints on the morphology and shape of the final regions. Subsequently, narrow or small and isolated parts of a given region exhibit an inherent difficulty of detection. Finally, future extensions of our work may spawn from the combined use of several pattern features, such as texture and chromaticity information, for solving a given segmentation problem. This can easily be performed using the same general framework by incorporating these distinct features into the velocity field definition.

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. of* 2^{nd} *Int. Conf. on Scale-Space Theories in Computer Vision*, pages 141–151, 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, March 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. *International Journal of Computer Vision*, 1:321–332, Jan. 1988.
11. N. Komodakis and G. Tziritas. Robust 3D Motion Estimation and Depth Layering. In *Proc. of Int. Conf. on Digital Signal Processing*, pages 425–428, 1997.
12. S. Liapis, E. Sifakis, and G. Tziritas. Color and/or Texture Segmentation Using Deterministic Relaxation and Fast Marching Algorithms, *to appear* in *Proc. of Int. Conference on Pattern Recognition*, Barcelona, 2000.
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 for Texture Segmentation. Technical Report RR-3440, INRIA, France, 1998.
17. T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer Science Press, Rockvill MD, 1982.
18. C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia. A Level Set Model for Image Classification. In *Proc.* 2^{nd} *Int. Conf. on Scale-Space Theories in Computer Vision*, pages 306–317, 1999.
19. J. Sethian. A Marching Level Set Method for Monotonically Advancing Fronts. *Proc. Nat. Acad. Sci.*, 93:1591–1595, 1996.
20. J. Sethian. Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces. *Acta Numerica*, pages 309–395, 1996.
21. J. Sethian. Fast Marching Methods. *SIAM Review*, 41:199–235, 1999.
22. E. Sifakis and G. Tziritas. Fast Marching to Moving Object Location. In *Proc. of* 2^{nd} *Int. Conf. on Scale-Space Theories in Computer Vision*, 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. M. Unser. Texture Classification and Segmentation using Wavelet Frames. *IEEE Trans. on Image Processing*, IP-4:1549–1560, Nov. 1995.
25. R. Kimmel V. Caselles and G. Sapiro. Geodesic Active Contours. *Int. Jour. of Computer Vision*, pages 61–79, 1997.
26. A. Yezzi, A. Tsai, and A. Willsky. A Statistical Approach to Snakes for Bimodal and Trimodal Imagery. In *Proc. of* 7^{th} *International Conference on Computer Vision*, pages 898–903, 1999.
27. T. Young and K. S. Fu. *Handbook of Pattern Recognition and Image Processing*. Acedemic Press, 1986.
28. 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.