# Some Recent Developments in Variational Image Segmentation

Tony Chan<sup>1</sup>, Berta Sandberg<sup>2</sup>, and Mark Moelich<sup>3</sup>

<sup>1</sup> UCLA Mathematics Department chan@math.ucla.edu \*

<sup>2</sup> TechFinity,Inc. bsand@math.ucla.edu <sup>†</sup>

<sup>3</sup> Aerospace Corporation mark.c.moelich@aero.org

**Summary.** This survey paper discusses some recent developments in variational image segmentation and active contours models. Our focus will be on region-based models implemented via level-set techniques, typified by the Chan–Vese (CV) model [11]. The CV algorithm can be interpreted as a level-set implementation of the piecewise constant Mumford–Shah segmentation model and has been quite widely used.

We will first present the basic CV algorithm and an extension to piecewise smooth approximations. We also discuss a recent development in convexifying the CV model to guarantee convergence to a global minimizer. Next, we discuss extensions to handle multi-channel images, including a vector-valued CV model [9], texture segmentation [10], object tracking in video [41], image registration [40], and a logic segmentation framework [49]. Then we discuss multiphase extensions to handle segmentation into an arbitrary number of regions, including the method of Vese and Chan [61] and recent developments of memory efficiency algorithms such as the piecewise constant level set method (PCLSM) of Tai et al. [36] and the multi-layer method of Chung and Vese [13].

Finally, we discuss numerically efficient methods that attempt to compute the optimal segmentation much faster than the original gradient-descent PDE-based method. These methods include the direct pointwise optimization method of Song and Chan [55], an operator-splitting method by Gibou and Fedkiw [26], and a threshold dynamics method by Esedoglu and Tsai [19].

**Key words:** image segmentation, active contours without edges, Mumford-Shaw, level sets, multi-phase, multi-channel, tracking, registration

<sup>\*</sup>This work was supported by Office of Naval Research contract N00014-06-1-0345, National Science Foundation contract DMS-0610079, and National Institute of Health as part of the Center for Computational Biology at UCLA contract U54 RR021813

 $<sup>^\</sup>dagger {\rm This}$  work was supported by Missile Defense Agency Small Business Technology Transfer Program contract HQ0006-05-C-7263

# 1 Introduction

Image segmentation is the process of partitioning an image into regions. Each region has a consistent trait throughout that is different from other regions in the image. Some common traits that have been captured are intensity, color, and texture. Once a decision is made on the desired traits, various segmentation methods are available to reach this goal.

This paper will focus on variational image segmentation and active-contour models and algorithms, which share the common feature that they define optimal segmentation as a minimizer of an objective function that generally depends on the given image and the traits that are used to identify the different segmented regions. The Euler–Lagrange equation of these models can often be described using a partial differential equation, which is iterated until it reaches steady state.

A contour is introduced into the image and is evolved until steady state thereby dividing the image into regions, see Figure 1. A very powerful and popular method for representing the contour is the level-set method originally developed by Osher and Sethian [45], which represents the contour implicitly as a particular (usually the zero) level of a (level-set) function. The main advantage of this representation is that topological changes, such as merging and pinching off of contours can be captured naturally through smooth changes to the level-set function.

In this paper, we will focus primarily on region-based (rather than edgebased) segmentation models. A prototypical example, and the primary one we will discuss in this paper, is the Chan–Vese "Active Contour Without Edges" model [11], which seeks the desired segmentation as the best piecewise constant approximation to a given image. The Chan–Vese model can be interpreted as a level-set implementation of the piecewise-constant special case of the more general Mumford–Shah segmentation model [43].

Due to its simplicity and robustness, the Chan–Vese model has become quite popular and has been adopted in many applications. As a result, a number of generalizations have been developed to improve both its applicability and efficiency. A natural generalization is multi-channel images. Initially, a vector valued method was used with an application in texture segmentation [10]. This was followed by an important conceptual generalization to a



Fig. 1. Evolution of a contour around objects.

logic framework allowing the user to use any logical combination of information in each channel to obtain the desired segmentation. Further extensions include object tracking in video sequences in the presence of clutter, registration of images to identify key objects, and color segmentation that can identify an object in an image with an arbitrary combination of colors.

Another direction of generalization to the basic Chan–Vese model is to multiphase models, which allow the segmentation of the image into arbitrary (> 2) regions. A natural, but inefficient, generalization is to use one level-set function for each phase, taking care to avoid overlap and uncovered regions. Various attempts have been made to improve on this basic approach. The multiphase method of Vese and Chan [61] only needs  $\log_2 n$  level-set functions to represent n regions, without any need to avoid overlap and uncovered regions, drastically improving the efficiency. More recently, Tai et al. [36] and Chung and Vese [13] have developed novel level-set methods that use only one level-set function to represent an arbitrary number of regions. We will review these methods in this paper.

A final direction of generalization is to improve the computational efficiency of these variation segmentation models. The typical approach of gradient flow (i.e., marching the Euler–Lagrange PDE to steady state) usually takes a long time to converge. A variety of methods have been developed to speed this up. One approach is to treat the models as a discrete optimization problem whose solution is the association of each pixel to a particular region. Song and Chan [55] proposed a direct optimization algorithm, which has the surprising property that for noiseless two-phase images the optimal solution can be provably obtained with only one sweep over the pixels. Gibou and Fedkiw [26] use an operator-splitting approach of treating the data term and the regularization (or curvature) term of the Euler–Lagrange equation in two separate steps, each of which can be computed very efficiently. Finally, Esedoglu and Tsai [19] use a threshold dynamics approach to obtain an efficient implementation. These methods will be discussed further in the paper.

The outline of the paper is as follows. Active-contour methods, and in particular the Chan–Vese model, are introduced in Section 2. In Section 3, we discuss multi-channel generalizations and in Section 4 we discuss multiphase generalizations. In Section 5, we discuss efficient implementations. Conclusions and possible directions for future research are given in Section 6.

# 2 Active Contours Methods

There are various schemes to deform the contour to the edges of an object. A quick summary and references for active contours using edge-detection algorithms are give below. For the rest of the paper we address active contours without edges, as written by Chan and Vese [11].

#### 2.1 Classic Active Contours

A classic approach to active contour models is to use the gradient of the image  $u_0$  to locate the edges of the object. Typically, an edge function is used that is positive inside homogeneous regions and strictly zero on the edges. Using this edge detection function, a functional is minimized with respect to contour C,

$$\inf E(C) = \int_{C} |C'(s)|^2 ds + \lambda \int_{C} g(|\nabla u_0(C(s))|)^2 ds,$$
(1)

where g is an edge-detection function. This model is by Caselles, Kimmel, and Sapiro and similar work by Kass, Witkin, and Terzopolous [8, 30]. The model cannot handle automatic topology changes of the contour, and depends on the parameterization of the curve.

In problems of curve evolution, including snakes and active contours, the level-set method of Osher and Sethian [45] has been used extensively, because it allows for automatic topology changes, cusps and corners; moreover, the computations are made on a fixed rectangular grid. Using this approach, geometric active-contour models, using a stopping edge-function, have been proposed in [7], and also in [38]. These models are based on the theory of curve evolution and geometric flows. The evolving curve moves by mean curvature, but with an extra factor in the speed, the stopping edge-function. Therefore, the curve stops on the edges, where the edge-function vanishes. An example of edge-functions used is given by:

$$g(|\nabla u_0|) = \frac{1}{1 + |\nabla (G_\sigma * u_0)|^2}$$

where g is a positive and decreasing function, such that  $\lim_{t\to\infty} g(t) = 0$ . The image  $u_0$  is first convolved with the Gaussian  $G_{\sigma}(x, y) = \sigma^{-1/2} \exp^{-|x^2+y|^2|/4\sigma}$ , especially for the cases where  $u_0$  is noisy. In practice, g is never zero on the edges, and therefore the evolving curve may not stop on the desired boundary. To overcome this problem, a new model has been proposed in [8], as a problem of geodesic computation in a Riemann space, according to the metric g. This gives an added term that increases the attraction of the evolving curve towards the boundary of the object, and is of special help when the boundary has high variations on its gradient values. For another related approach, see also [31].

These models use the gradient of a smoother version of the image  $u_0$ , to detect edges. If the image is noisy, the smoothing in the edge-function has to be strong, thus blurring edge features, or a pre-processing has to be implemented, to remove the noise.

#### 2.2 Active Contours without Edges

The Chan–Vese active-contour model without edges proposed in [11] does not use the stopping edge-function g to find the boundary. The stopping term is based on Mumford–Shah [43] segmentation techniques. The equation for the Mumford–Shah in (u, C) is obtained by minimizing the functional:

$$E(u,C) = \int_{\Omega} (u - u_0)^2 dx + \mu \operatorname{length}(C).$$

While the functional itself is elegant, in practice it is difficult to find a solution as the functional is non-convex, and has an unknown C. Various solutions have been proposed. One solution uses region growing, minimizing the Mumford–Shah functional using greedy algorithms [39, 32]. Elliptical approximations embed the contour C in a 2D phase-field function [1]. The Mumford–Shah functional has also been calculated using a statistical framework [67].

Let  $\Omega$  be a bounded open subset of  $\mathbb{R}^2$ , with  $\partial \Omega$  the boundary. Let  $u_0$  be a given image such that  $u_0 : \overline{\Omega} \to \mathbb{R}$ . Let  $C(s) : [0,1] \to \mathbb{R}^2$  be a piecewise parameterized  $C^1$  curve.

We choose a method with the following form:

$$\inf_{c^+, c^-, C} F(c^+, c^-, C),$$

where

$$F(c^{+}, c^{-}, C) = \mu |C| + \lambda^{+} \int_{in(C)} |u_{0} - c^{+}|^{2} dx + \lambda^{-} \int_{out(C)} |u_{0} - c^{-}|^{2} dx,$$
(2)

where |C| denotes the length of C,  $c^+$  and  $c^-$  are constant unknowns representing the "average" value of  $u_0$  inside and outside the curve, respectively. The parameters  $\mu > 0$ , and  $\lambda^+, \lambda^- > 0$ , are weights for the regularizing term and the fitting term, respectively.

Minimizing the fitting error in (2), the model approximates the image  $u_0$  with a piecewise-constant function, taking only two values, namely  $c^+$  and  $c^-$ , and with one edge C, the boundary between these two constant regions. The object to be detected will be given by one of the regions, and the curve C will be the boundary of the object. The additional length term is a regularizing term, and has a scaling role. If  $\mu$  is large, only larger objects are detected, while for small  $\mu$ , objects of smaller size are also detected. Because the model does not make use of a stopping edge-function based on the gradient, it can detect edges both with and without a gradient as can be seen in Figure 2. It is well known that (2) can be viewed as a special case of the Mumford–Shah segmentation [43].

We rewrite the original model (2) in the level-set formulation. Let the evolving curve C be embedded as the zero level set of a Lipschitz continuous function  $\phi$ , i.e.,  $C(\phi) = \{(x, y) \in \Omega : \phi(x, y) = 0\}$ , with  $\phi$  having opposite signs on each side of C. Following [66] and [11], the energy can be written as:



Fig. 2. The Chan–Vese algorithm is able to segment the image without edges.

$$\begin{split} F(c^+, c^-, \phi) &= \mu |C(\phi)| + \lambda^+ \int_{\phi \ge 0} |u_0(x, y) - c^+|^2 dx dy \\ &+ \lambda^- \int_{\phi < 0} |u_0(x, y) - c^-|^2 dx dy. \end{split}$$

Minimizing  $F(c^+, c^-, \phi)$  with respect to the constants  $c^+$  and  $c^-$ , for a fixed  $\phi$ , yield the following expressions for  $c^+$  and  $c^-$ , function of  $\phi$ :

$$\begin{cases} c^+ = \operatorname{average}(u_0) \text{ on } \phi \ge 0, \\ c^- = \operatorname{average}(u_0) \text{ on } \phi < 0. \end{cases}$$

Minimizing the energy  $F(c^+, c^-, \phi)$  with respect to  $\phi$ , for fixed  $c^+$  and  $c^-$ , using a gradient descent method, yields the associated Euler-Lagrange equation for  $\phi$ , governed by the mean curvature and the error terms (see [11] for more details).

$$\frac{\partial \phi}{\partial t} = \delta_{\epsilon} \left[ \mu \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda^{+} (u_{0} - c^{+})^{2} + \lambda^{-} (u_{0} - c^{-})^{2} \right]$$
(3)

in  $\Omega$ , and with the Neumann boundary conditions.

Using a level-set formulation with this model allows the initial contour to find any number of objects from an initial contour anywhere in the image. For general information, one may consult [44] and [51].

# 2.3 Piecewise Smooth Segmentation

Thus far, we have described objects that are assumed to have constant intensity. The piecewise smooth extension allows for two possible situations. One motivation is an algorithm that can simultaneously denoise and segment an image. A second situation occurs when an object's intensity changes gradually. The general Mumford–Shah piecewise smooth functional [43] is defined as:



Fig. 3. A nebula is segmented using a single contour, giving intensity values, which are the same as in the original image.

$$\inf_{u,\Gamma} E_{ms}(u,\Gamma[u_0]) = \int_{\Omega} |u - u_0|^2 dx + \mu \int_{\Omega/\Gamma} |\nabla u|^2 dx + \nu |\Gamma|,$$

where  $\mu$  and  $\nu$  are positive constants. This allows for a varying intensity of the object, while keeping the boundaries constant. A two-phase function is defined with level set  $\phi$  as follows:

$$u(x) = u^{+}(x)H(\phi(x)) + u^{-}(x)(1 - H(\phi(x)))$$

Here  $u^+$  and  $u^-$  are  $C^1$  functions up to the boundary at  $\phi = 0$ . Corresponding Euler–Lagrange equations are as follows:

$$\mu^{2}(u^{+}-u_{0}) = \Delta u^{+} \text{ on } \phi > 0, \frac{\partial u^{+}}{\partial n}| = 0 \text{ on } \phi = 0,$$
$$\mu^{2}(u^{-}-u_{0}) = \Delta u^{-} \text{ on } \phi > 0, \frac{\partial u^{-}}{\partial n}| = 0 \text{ on } \phi = 0.$$

Denoising is done in the homogeneous region, while leaving the boundary  $\phi = 0$  unchanged. In Figure 3 the correct features are captured within a single object segmentation.

# 2.4 Global Minima Via Convexification

The variational formulation in the Chan–Vese model is non-convex and a typical gradient-descent implementation is not guaranteed to converge to the global minimum and can get stuck in local minima. A typical case is where the

 $\overline{7}$ 

contour is stuck at the outer boundary of an object with an interior hole. Various tricks can be devised to improve the global convergence. One technique, which is used in the original paper [11], is to modify the delta function in the Euler-Lagrange equation so that it is nonzero everywhere. This corresponds to allowing contours to be initiated everywhere in the image, enhancing the chance of capturing the global minimum. Another idea is to initialize the optimization with a large number of small close contours uniformly distributed in the image, which has a similar effect.

A more novel, and fundamentally different, approach has been proposed more recently in [21, 22]. The basic idea is to convexify the objective function by taking advantage of the implicit geometric properties of the variational models. Using an auxiliary variable u, the Chan–Vese model can be recast in the following *convex* minimization problem:

$$\min_{c^-, c^+ \in \mathbb{R}} \min_{0 \le u(x) \le 1} \int_D |\nabla u| dx + \lambda \int_D [(c^+ - u_0)^2 - (c^- - u_0)^2] u(x) dx.$$

It is proved in [21, 22] that if  $(c^+, c^-, u(x))$  is a solution of above minimization problem, then for almost every  $\mu \in (0, 1)$ , the triplet  $(c^+, c^-, \chi_{x:u(x) \ge \mu}(x))$  is a global minimizer of the Chan–Vese model. Since the above minimization problem is convex, it admits many efficient implementations and thus this approach allows an efficient computation of the global minimization of the Chan–Vese model.

# 3 Multi-Channel Extensions in Chan–Vese Model

The Chan–Vese model described above is very flexible. This flexibility lends itself to expanding it in a variety of ways. Initially it was expanded to vectorvalued systems. This allowed for combining multiple images simultaneous to segment the images and identify the key object. We introduce the generalized models below.

#### 3.1 Vector-Valued Models

In this chapter, the Chan–Vese method [11] is extended to vector-valued images. An example of the vector-valued object detector can be seen in Figure 4. Each channel has a different piece missing, but when the two channels are combined, the complete object is detected. Another example where this algorithm is of particular interest is an occlusion occurring in one channel, while a second channel, complete yet noisier, is available. Another example is RGB images, where intensity detectors and channel-by-channel boundary detectors fail.

Let  $u_{0,i}$  be the *i*th channel of an image on  $\Omega$ , with i = 1, ..., N channels, and C the evolving curve. Each channel would contain the same image with some differences, for instance different wavelengths at which the image was



Fig. 4. Each channel has a different part of the same triangle missing. The vectorvalued algorithm can detect the full triangle.

taken, color images, etc. Let  $\overline{c^+} = (c_1^+, ..., c_N^+)$  and  $\overline{c^-} = (c_1^-, ..., c_N^-)$  be two unknown constant vectors.

The extension of the Chan–Vese model to the vector case is:

$$\begin{split} F(\overline{c^+}, \overline{c^-}, \phi) &= \mu \cdot \text{Length}(C) + \int_{inside(C)} \frac{1}{N} \sum_{i=1}^N \lambda_i^+ |u_{0,i}(x, y) - c_i^+|^2 dx dy \\ &+ \int_{outside(C)} \frac{1}{N} \sum_{i=1}^N \lambda_i^- |u_{0,i}(x, y) - c_i^-|^2 dx dy, \end{split}$$

where  $\lambda_i^+ > 0$  and  $\lambda_i^- > 0$  are parameters for each channel.

As in the scalar case, the model looks for the best vector-valued approximation taking only two values, the constant vectors  $\overline{c^+}$  and  $\overline{c^-}$ . The active contour C is the boundary between these two regions. The energy balances the length of the contours in the image, with the fitting of  $u_0$ , averaged over all channels. In this form, when the contour C surrounds the objects, our model can detect edges present in at least one of the channels, and not necessarily in all channels. We can associate this property with the syntax "OR". Likewise we can imagine a system using the intersection of two objects. We will return to this boolean logic framework later in the paper.

An example can be found in multi-spectral images. In Figure 5, we have an airplane imaged from mid-wave and long-wave infrared channels. One channel is very noisy, making it very difficult to detect the edges of the entire airplane, while the other, less noisy, has a partial occlusion of the airplane. Each channel is insufficient for determination of the complete contour. However, in combination, most of the features are detected.

The vector-valued Chan–Vese model can also be used on color images. By dividing the image into red, green, and blue (RGB) channels, one can detect objects normally undetectable when the color image is transformed to a scalar intensity image. An example of this can be seen in Figure 6. We can see the "stop-light" in the RGB image, while the scalar intensity image has the bottom object missing. Channel-by-channel detection would also be



Fig. 5. While the first channel has little noise, but has an occlusion in it, the second channel is very noisy. From these two pictures, we try to detect as much of the airplane as possible. The parameters are as follows:  $\mu = 0.001 \cdot 255^2$ ,  $\lambda_1^+ = \lambda_1^- = 1$ ,  $\lambda_2^+ = \lambda_2^- = 0.55$ . In this example, we first performed a renormalization of the channels to [0, 255].



Fig. 6. We give here an example of a color image that has three objects of different colors, while the corresponding gray scale image only shows two of them. The boundary of all the circles is found, while in the gray-scale image the boundary of one of the circles would never be detected. Note that, since this image does not have gradient edges, a gradient-based algorithm would not be able to find the three objects. The parameters are as follows:  $\mu = 0.06 \cdot 255^2$ ,  $\lambda_i^+ = \lambda_i^- = 1$ , for i = 1, 2, 3.

insufficient in this case, since features of the image are not complete in any single channel. Our model, however, detects all three features easily. Also note, in this particular example, the algorithm detects edges without gradient.

#### 3.2 Texture Segmentation using Vector-Valued Models

There are several problems specific to texture segmentation. When the textures have the same intensities, it is very difficult for the standard segmentation models to tell them apart. Another problem inherent in textured segmentation is that it is often difficult to pick out the boundary between two textures because there is no sharp difference between them. Finally, any texture segmentation algorithm should be robust to noise, since texture has small patterns that are "noise"-like.

We do not assume any apriori knowledge or statistical information on the type of textures, or on the type of intensity, or on the location of boundaries. The proposed model described in detail in [10] is general, and can be applied in many situations.

For the texture discrimination, we propose to use Gabor [24] functions, having properties similar to those of early visual channels, being localized in space and frequency domains [23, 15]. The Gabor functions are convolved with the original textured image to obtain different channels. Some of these channels will be the input of the multi-channel active-contour algorithm.

For other possible transforms instead of the Gabor transform, for texture discrimination, such as wavelets, for example [28]. This paper is related to many other works on active contours and texture segmentation, such as [46], (already mentioned above), [57, 56, 60, 52, 32]. Additional related papers are [37, 34, 6, 48]. Other related works on segmentation, edge-preserving smoothing, and vector-valued images (e.g., multi-channels, color, etc), are [16, 32, 50, 53, 54, 67].

Using all of the channels for segmentation is impractical. Some of the images are redundant while others add noise and obscure detection. At this point we divide our model into two parts: "supervised" texture segmentation, when the user chooses the "best" Gabor transforms, to be used as input channels; and "unsupervised" texture segmentation, where the Gabor transforms to be used are chosen by a dynamic automatic criterion.

The case of supervised texture segmentation allows one to use fewest number of transforms in order to segment the image, and as a result it does a very good job, with optimal computational efficiency. The case of unsupervised texture segmentation is similar to the work of [29, 64]. The criterion that we used for the automatic choice of the Gabor transforms is based on the following: we want the images to have the highest intensity differences relative to the mean of the image. Thus for each transformed channel i we calculate the following:

$$s_i = |c_i^+ - c_i^-|.$$

The  $s_i$  is calculated for each channel. Only n (n < 45) channels, corresponding to the first n largest values of  $s_i$ , are used in our active contour model as inputs, at the initial time. We recalculated the  $s_i$  at later iterations choosing the n largest values again. This allows for better choices of the channels as



Fig. 7. Supervised model with three different Gabor transforms as input channels. Parameters:  $\lambda_i = 1, \mu = 4000, \gamma_i = .3$ . The boundary of the full square is found, and the binary segmentation is represented by "gray" and "black" ("black" if  $\phi \ge 0$ , and "gray" if  $\phi < 0$ ).

the contour is refined closer to the desired object. This criterion does a good job of picking out automatically the "best" channels.

In Figure 7, there is a square in the middle of the image, but it is very hard to distinguish it. The Gabor transforms contrast the square, with the outside texture, and the active-contour model has no problem detecting the edges of the square. In Figures 8, we have used the unsupervised criteria for choosing the Gabor transforms. The segmentation is done well, with the criteria set for unsupervised segmentation.

#### 3.3 Logic Operations on Region-Based Active Contours

The Chan–Vese method of active contours without edges is a region-based method. This is a significant benefit, as it is especially important when finding logical combinations of objects.

Rather than comparing contrast of the object, it compares the fitting errors of each channel. The model does not care that each channel has different intensity values, instead it wants a contour that will minimize the fitting errors based on the average value for each channel.

To set up the logical framework we define two separate logic variables,  $z_i^{in}$  and  $z_i^{out}$ , to denote whether a point (x, y) is in C or not:

 $z_i^{in}(u_0^i,x,y,C) = \begin{cases} 0 & \text{if } (x,y) \in C \text{ and } (x,y) \text{ inside the object in channel } i, \\ 1 & \text{otherwise;} \end{cases}$ 

 $z_i^{out}(u_0^i, x, y, C) = \begin{cases} 1 & \text{if } (x, y) \notin C \text{ and } (x, y) \text{ is inside the object in channel } i, \\ 0 & \text{otherwise.} \end{cases}$ 

A natural way to define  $z_i^{in}$  and  $z_i^{out}$  for the Chan–Vese model is as follows:

$$z_{i}^{in}(u_{0}^{i}, x, y, C) = \frac{|u_{0}^{i}(x, y) - c_{+}^{i}|^{2}}{\max_{(x,y) \in u_{0}^{i}} u_{0}^{i}},$$
  
$$z_{i}^{out}(u_{0}^{i}, x, y, C) = \frac{|u_{0}^{i}(x, y) - c_{-}^{i}|^{2}}{\max_{(x,y) \in u_{0}^{i}} u_{0}^{i}}.$$
 (4)



Fig. 8. Unsupervised texture segmentation with only four active transforms. It is successful in segmenting the zebras and disregarding the stripes.

Note that we use "0" as the "true" value, and "1" as the "false" value, which is the reverse of the usual convention. This is more convenient because our framework is based on a minimizing of an objective function and thus we want the 0 value to correspond to "true".

For the complement of the object in channel i we define:

$$z_i^{in'} = 1 - z_i^{in}$$
  
$$z_i^{out'} = 1 - z_i^{out}$$
(5)

Following the structure of logic operators, we now want to define a truth table for the logic model the the variables described above. We treat the points inside C separately from those outside C.

Continuing with the two-channel example  $A_1 \cup A_2$ , we define it in truth table form. The truth table needs to reflect the union of  $z_i^{in}$  and the intersection of  $z_i^{out}$ . For the point  $(x, y) \in C$  the variable  $z_i^{in}$  is defined. If the



Fig. 9. Logic operations inside and outside the object. The upper triple of images shows that the union of the inside (black) region gives the union of the 2 objects in  $A_1$  and  $A_2$ . The bottom triple shows that the intersection of the outside (black) region gives the complement to the union of two objects.

point  $(x, y) \in C$  is in the object in either channel, the logic model returns 0, otherwise it returns 1—this reflects the union of the inside of the object. If  $(x, y) \in \Omega \setminus C$ , the variable  $z_i^{out}$  is defined. The logic model returns 0 if (x, y) is *not* in the object in either channel, otherwise it will return 1, - this represents the intersection of the outside of the object. The column marked  $A_1 \cup A_2$  relates this information. The logic operations  $A_1 \cap A_2$  and  $A_1 \cap \neg A_2$  are calculated in a similar fashion. For intersection of objects, we take the intersection of the inside of objects and the union of the outside of objects. For negation we substitute  $z'_i$  for  $z_i$  as shown in (5).

For the union and intersection function of logic variables we choose:

$$f_{z_1 \cup z_2} = (z_1 \cdot z_2)^{\frac{1}{2}} f_{z_1 \cap z_2} = 1 - ((1 - z_1)(1 - z_2))^{\frac{1}{2}}.$$

The square roots of the products are taken to keep them of the same order as the original scalar model. Combining the interpolation functions for union of inside the objects, and intersection outside the objects we get the union of objects:

$$f_{A_1 \cup A_2}(x, y) = \sqrt{z_1^{in}(x, y) z_2^{in}(x, y)} + 1 - \sqrt{(1 - z_1^{out}(x, y))(1 - z_2^{out}(x, y))}.$$

Likewise, to get the intersection of objects, we combine the intersection of the inside with the union of the outside, resulting in the following objective function for the intersection of objects: Some Recent Developments in Variational Image Segmentation

$$f_{A_1 \cap A_2}(x,y) = 1 - \sqrt{(1 - z_1^{in}(x,y))(1 - z_2^{in}(x,y))} + \sqrt{z_1^{out}(x,y)z_2^{out}(x,y)}.$$

In the above, we have used the interpolation functions to directly derive the objective functions corresponding to a given logical expression. Even though we have by-passed the corresponding truth table, it can be easily verified that the resulting objection functions do interpolate the function values given in the truth table. The functional may be written using the level-set formulation as described in Section 2. Now we can rewrite the functional F for a general  $f(z_1^{in}, z_1^{out}, \ldots)$  using the level-set function  $\phi$ .

The objective function for the variational model is:

$$F(\phi, c^+, c^-) = \mu |C(\phi)| + \lambda [\int_{\Omega} f_{in}(z_1^{in}, ...z_n^{in}) H(\phi) + f_{out}(z_1^{out}, ..., z_n^{out})(1 - H(\phi)) dx].$$

Derivation of the Euler–Lagrange equation is similar to that of the scalar model and yields the following differential equation (with Neumann boundary conditions):

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \Big[ \mu \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda (f_{in}(z_1^{in}, \dots, z_n^{in}) - f_{out}(z_1^{out}, \dots, z_n^{out})) \Big],$$

which at steady state gives the solution.

For example, for the two logic models presented earlier, the corresponding Euler–Lagrange equations are:

$$\begin{split} \frac{\partial \phi_{A_1 \cup \ldots \cup A_n}}{\partial t} &= \\ \delta_{\epsilon}(\phi) \Big[ \mu \nabla \cdot \Big( \frac{\nabla \phi}{|\nabla \phi|} \Big) - \lambda ((\prod_{i=1}^n (z_i^{in}))^{\frac{1}{n}} + 1 - (\prod_{i=1}^n (1 - (z_i^{out})))^{\frac{1}{n}}) \Big], \\ \frac{\partial \phi_{A_1 \cap \ldots \cap A_n}}{\partial t} &= \\ \delta_{\epsilon}(\phi) \Big[ \mu \nabla \cdot \Big( \frac{\nabla \phi}{|\nabla \phi|} \Big) - \lambda (1 - (\prod_{i=1}^n (1 - z_i^{in}))^{\frac{1}{n}} + (\prod_{i=1}^n z_i^{out})^{\frac{1}{n}}) \Big]. \end{split}$$

Even though the form is complicated, the implementation is very similar to that of the scalar model that is in (3). The details for this scheme can be found in [11, 49]. In this section, we show some examples of the performance of the logical active-contour models described in Section 3.

We show a real life example in Figure 10 with two brain images. They are two MRIs of the brain taken in a time sequence, each with a synthetic tumor placed in a different spot. Using logic operation  $A_1 \cap \neg A_2$ , the tumor in the first image may be extracted, i.e., the logic operations find the object in the first image that is different from the second. The reverse is also true. Using the logic model that describes  $\neg A_1 \cap A_2$ , the model finds the object in the

15



**Fig. 10.** Region-based logic model on a MRI scan of the brain. The first channel  $A_1$ , has a synthetic brain tumor in one place; in the second image the synthetic brain tumor is in a different place. The images are registered. By design we want to find the tumor that is in  $A_1$  and not  $A_2$ ,  $A_1 \cap \neg A_2$ . Likewise we want to find the tumor in  $A_2$  that is not in  $A_1$  and  $\neg A_1 \cap A_2$ .

second image that is not in the first. This happens to be a very complicated example as there are a lot of features and textures. Not only does the model find the tumor, but using logic operations gives the user the capability to define more precisely how information from the different channels are to be combined in order to obtain a desired segmentation, as well as the freedom to use all possible logical combinations using a systematic framework.

In practical terms, the logic framework allows for a single solution global minimum as the union or intersection of the object depending on the model chosen. The vector-valued function depends on the initial contour for the final output, giving either union or intersection of the objects.

# 3.4 Target Tracking in Video

In this section, we show how the Chan–Vese segmentation model can be extended to track deforming objects in video sequences. This methodology was developed by Moelich [40, 41]. Since the Chan–Vese algorithm finds an optimal piecewise-constant approximation to an image, this algorithm works best in tracking objects that have nearly uniform intensity. The main idea is to sequentially segment the frames of a video sequence by using the final partition from one frame as the initial partition of the next.



Fig. 11. Results of tracking an object using a modified version of the Chan–Vese algorithm.

An estimate of each initial contour, which is based on a number of previous frames, can also be used. This, however, is not necessary unless the frameto-frame motion is large compared to the size of the object. Figure 11 shows sample frames from the complete sequence. Note that the algorithm is able to capture much information about the person being tracked, including gait and posture.

Some important modifications are made to the basic Chan–Vese model to adapt it to tracking objects. The first is to use a local background model, where the background is isolated to the region outside, but close to the contour. Second, reintializing the distance function maintains a local minima. Furthermore, once the desired object is identified, the segmentation should occur in the region of interest surrounding the object to maintain a "global" minima.

This method can fail when the estimated position of the object in a frame is far from its true position. This can happen when the frame-to-frame motion of the object is large relative to the size of the object. In this case, the segmentation contour is not in contact with the object and can either begin to isolate a similar nearby object, or vanish. Little can be done if it begins to segment another similar object. If the contour vanishes, however, it can be successively enlarged until it finds the desired object. The image in Figure 12 is the completed segmentation of a frame. This contour is used as the initial contour of the next frame. Due to camera motion, the contour misses the object in the second frame. Since the estimated intensity for the object is not within the contour, the contour shrinks until it vanishes. When this happens, the algorithm successively enlarges the contour until it contacts the object, in which case the algorithm is able to isolate it.

The use of the level-set framework makes "enlarging" the segmentation contour trivial. Recall that the segmentation contour is implicitly defined as the zero level set of a higher-dimensional function  $\varphi$ , where  $\varphi < 0$  inside the contour and  $\varphi > 0$  outside of the contour. Enlarging the segmentation contour is the same as "lowering" the level set  $\varphi$ . By continually reinitializing the distance function, the value of  $|\nabla \varphi|$  is approximately equal 1 near the contour (zero level set). To increase the size of the contour by a fractional amount f, we can simply modify  $\varphi$  as follows:



Fig. 12. Illustration of how algorithm handles position errors. The child moved far from frame to frame, by enlarging the contour the child is found in the following image.



Fig. 13. Tracking in presence of background clutter and poor contrast.

$$\varphi^{\text{new}} = \varphi^{\text{old}} - f \, d/2,\tag{6}$$

where d is an estimate of the diameter of the contour, which is made before the segmentation is applied. We used the larger of the height and width of the contour in the image as the estimate of the diameter.

Figure 13 gives an example sequence that was produced by this algorithm, tracking the car successfully in a highly cluttered environment.

### 3.5 Color Segmentation

The Chan–Vese segmentation model was originally developed to segment piecewise constant images. This algorithm was extended to isolate regions of constant color with the vector-valued models. We described a method for isolating objects that are composed of more than one color. This methodology was developed by Moelich [42].

An object of interest is often composed of a small number of different colors. For example, the cat in Figure 14 is composed of the colors black and white. A piecewise constant segmentation algorithm that is based on intensity alone, would not be able to isolate the entire cat as a single object.



Fig. 14. A black and white cat and output of color logic model.

This segmentation model assumes some apriori knowledge about the colors of the object to be isolated. This can be introduced to the algorithm, for example, by "clicking" on the desired colors in a graphical display. Given a color image  $I: \Omega \to \mathbb{R}^3$  and a set of colors  $c = (c_1, \ldots, c_n)$ , the prototype color logic model uses OR and AND framework described in the previous section to find a contour C that minimizes the energy

$$E(C; c) = \lambda_{in} \int_{\Omega_{in}} F_{in}(I(x); c) + \lambda_{out} \int_{\Omega_{out}} F_{out}(I(x); c) + \mu \operatorname{length}(C),$$
(7)

where

$$F_{in}(I(x); c) = \left(\prod_{i=1}^{n} k_i \| I(x) - c_i \| \right)^{1/n},$$
(8)

$$F_{out}(I(x); c) = 1 - (\prod_{i=1}^{n} k_i \| I(x) - c_i \|)^{1/n},$$
(9)

and where  $\lambda_{in}$ ,  $\lambda_{out}$ , and  $\mu$  are design parameters, and  $\Omega_{in}$  and  $\Omega_{out}$  are the regions interior and exterior to the contour C, respectively. The values of  $k_i$  are chosen to normalize the quantities  $||I(x) - c_i||$  and to ensure that they lie in the unit interval.

Figure 15 shows three additional segmentations that were produced by this model. In each case, two to six different colors were chosen before the segmentation.

The models for the two regions, given by (8) and (9) above, are effective for many images. When the colors in the background are similar to the colors in the object, then an independent background model can be used. In this case, the model for the background in (9) is replaced by

$$F_{out}(I(x); c^{out}) = \left(\prod_{j=1}^{m} k_j \| I(x) - c_{out,j} \| \right)^{1/m},$$
(10)



Fig. 15. Additional example of color logic model.



Fig. 16. Illustration of improved background model. Choosing three colors (left) or two colors (right) with first background model, and choosing three object colors and three background colors for improved background model.

where  $c_{out}$  is the set of m colors used to describe the exterior region. The two images on the left of Figure 16 were generated without this new model. In each of these cases, the segmentation was stopped before completion. In the image on the left, the colors red, white, and blue were selected from the flag. Since the color of the clouds behind the flag are nearly white, the algorithm considers them part of the object, and the segmentation contour grows to include them. In the middle image, only the colors red and blue were chosen. In this case the clouds, along with the white strips are excluded. Because of the regularity term, the thin red strips are also excluded.

The improved background model (10) was applied to the image on the right of Figure 16. In this case, the colors red, white, and blue were selected from the flag to define the object model, and additional three colors were selected from the background to define the background region. The use of independent models for the object and background regions provides the desired segmentation.

#### 3.6 Image Registration

An algorithm for the joint segmentation and registration of images. is decribed. Similar to other algorithms that we have discussed, the main idea is to use information from more than one image to develop a segmentation. We do not assume that the images are registered, or "lined up." This algorithm simultaneously finds both the segmentation and the registration between the



Fig. 17. Individual contours are mappings of a reference contours.

images. This approach was created by Yezzi, Zöllei, and Kapur [65], and further developed by Moelich [41]. Our description focuses on the case of two images; however, the same method can be applied to a larger number of images.

Consider two, possibly unregistered images,  $I^1: \Omega \to \mathbb{R}$  and  $I^2: \Omega \to \mathbb{R}$ . Let  $\{\Omega^1_{obj}, \Omega^1_{gb}\}$  denote the segmentation of the image  $I^1$  and let  $\{\Omega^2_{obj}, \Omega^2_{gb}\}$  denote the segmentation of image  $I^2$ . These two segmentations are viewed as the mapping of a reference segmentation  $\{\Omega_{obj}, \Omega_{gb}\}$  under a parameterized mapping g. Figure 17 illustrates this relationship, where  $p_1$  and  $p_2$  represent two sets of parameters for the mapping g.

The segmentation and registration  $p = (p_1, p_2)$  are found by minimizing a segmentation energy that is defined as

$$E(\Omega_{obj}, \Omega_{bg}; p) = E_{obj}(\Omega_{obj}; p) + E_{bg}(\Omega_{bg}; p) + \mu |\partial \Omega_{obj}|, \qquad (11)$$

where

$$E_{obj}(\Omega_{obj}; p) = \int_{\Omega_{obj}} \lambda_{obj} F_{obj}(x; p) \, dx, \qquad (12)$$

$$E_{bg}(\Omega_{bg}; p) = \int_{\Omega_{bg}} \lambda_{bg} F_{bg}(x; p) \, dx.$$
(13)

The expressions for the region descriptors  $F_{obj}(x; p)$  and  $F_{bg}(x; p)$  depend upon which segmentation models are used.

There are many valid choices for the mapping g, but for many applications a simple Euclidean transformation g(x; p) = MRx + T is adequate, where

$$M = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}, \quad R = \begin{bmatrix} \cos \Delta\theta - \sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{bmatrix}, \quad \text{and} \quad T = \begin{bmatrix} \Delta x \\ da \Delta y \end{bmatrix}$$

The parameters of the transformation are given by  $p = (\Delta x, \Delta y, \Delta \theta, m)$ , where  $\Delta x$  and  $\Delta y$  are translation,  $\Delta \theta$  is rotation about the center, and m is

magnification. When p = (0, 0, 0, 1), the transformation  $g(\cdot; p)$  is the identity map.

The segmentation energy (11) depends on both the partition of the image and on the registration parameters. This energy can be minimized by "interleaving" the processes of segmentation and registration, as suggested in [65]. During each iteration of the algorithm, the segmentation, i.e., the level-set function  $\varphi$ , is first held constant while the estimates of the registration parameters are improved, then these parameters are held fixed while the level-set function  $\varphi$  is evolved.

The registration parameters are improved by using a gradient-descent line search. The direction of the search is found by taking numerical derivatives of the energy  $E^{obj}(\Omega_{obj}; p)$  with respect to the components of  $p = (p_1, p_2)$ . Since  $p_1$  and  $p_2$  are independent, it is useful to update each set of parameters separately. Also, since translation, rotation, and magnification have different scales, it is useful to have different time steps for these components.

The segmentation is improved by making a small evolution of the level-set function by numerically integrating

$$\frac{\partial \varphi}{\partial t} = \delta_{\varepsilon}(\varphi) \left( \lambda_{obj} F_{obj}(x; p) - \lambda_{bg} F_{bg}(x; p) + \mu \operatorname{div}(\frac{\nabla \varphi}{|\nabla \varphi|}) \right) \quad \text{in } \Omega, \quad (14)$$

$$\frac{\partial \varphi}{\partial n} = 0 \quad \text{on } \partial \Omega, \tag{15}$$

$$\varphi(x,0) = \varphi_k \quad \text{in } \Omega \tag{16}$$

for a few steps. The energy decreases with each iteration. The process of alternatively improving the registration and segmentation continues until the algorithm converges.

When the initial estimate of the registration parameters are poor, an initial registration phase can be used to put the segmentation contours in contact with the object of interest in each image. This initial registration phase assumes that an apriori estimate of the average intensities of the object of interest is known. The initial phase can be applied to either, or both images. For sake of discussion, we assume that the initial guess for  $p_1$  is known to be reasonable, but that the error in the initial guess for  $p_2$  can be large. It is further assumed, in this case, that the average intensity of  $\Omega_{obj}^1$  is an estimate for  $\hat{c}_{obj}^2$ , of the intensity of  $\Omega_{obj}^2$ . The estimate  $\hat{c}_{obj}^2$  is used to construct an initial registration phase energy

$$E_{\psi}(C_2) = \frac{1}{m} \int_{C_2} \psi^m(x) \, ds, \qquad (17)$$

where  $\psi(x)$  is the distance from x to the intensity of interest in  $I_2$  and where  $C_2 = \partial \Omega_{obj}^2$  is the segmentation contour. The value of m can be taken as either 1 or 2. A value of m = 1 usually gives a faster convergence, although using m = 2 gives better behavior near the minima.



Fig. 18. Typical behavior of the algorithm. Initial contour (top), end of initial registration phase (middle), and final segmentation (bottom).

A gradient descent is used to minimize the energy  $E_{\psi}$  of the initial registration phase. The values of  $\Delta x$  and  $\Delta y$ , which are registration parameters for the image, are updated using the following equations:

$$\frac{\partial \Delta x}{dt} = -\frac{\partial E_{\psi}}{\partial x} = -\int_{C_2} \psi^{m-1} \frac{\partial \psi(\mathbf{x})}{\partial x} ds,$$

$$\frac{\partial \Delta y}{dt} = -\frac{\partial E_{\psi}}{\partial y} = -\int_{C_2} \psi^{m-1} \frac{\partial \psi(\mathbf{x})}{\partial y} ds.$$
(18)

Figure 18 illustrates the typical behavior of the complete algorithm. In this example, the piecewise constant Chan–Vese segmentation model is used. The images in the left and right columns, respectively, were taken from slightly different perspectives, at sightly different times. The estimates of the registration parameters are reasonable for the image on the left, but not for the image on the right. The initial registration phase is used to drive the contour in the image on the right toward the object of interest. Once the initial phase energy is minimized, the phase changes and joint segmentation and registration is used to both isolate the object and determine the registration parameters.



**Fig. 19.** Logical AND model restricts the segmentation. Initial contour (top) and logical AND (bottom).



Fig. 20. Logical OR model combines information. Initial contour (top) and final segmentation (bottom).

Figure 19 illustrates how the logical AND model can be used. The image of the person on the left is used as a template to restrict the segmentation of the image of the person on the right. The initial contours are shown in the top row, and the final contours are shown on the bottom. Note the the person in the background is ignored.

In Figure 20, the logical OR model is used to reconstruct an image from two incomplete images.



Fig. 21. A physical representation of the difference in region segmentation between one and two level sets. The number of regions possible is  $2^m$  where m is the number of level sets, here m = 1, 2.

### 4 Multi-Phase Extensions

Several multi-phase algorithm extensions are described below. The original one developed by Chan and Vese [61] adds a new contour to add new regions. For n contours one would be able to develop algorithms for  $2^m$  regions. This can be bulky to program. Further work has been done on multiphase methods to increase their efficiency and ease of programming. A recursive method was introduced by [25] that segments the image in a hierarchical way. First into two regions, then segmenting each region into two new regions, and so on. Piecewise constant method by [36] motivated by island dynamics for modeling epitaxial growth is used for memory efficiency. A multi-layer method by [13] uses different ranges of a function for different regions, however, nested regions and triple junctions require more than one function. Binary methods were introduced by [35] and [55], which require no Delta or Heaviside functions, obtain direct optimization, for faster implementation.

#### 4.1 Multi-Phase Active Contours without edges

In the previous sections we have discussed segmentation for a single object. We now show the multi-phase extensions that have been suggested by [62]. The initial multi-phase algorithm follows the natural extension of the piecewise constant Mumford–Shah functional, as shown below:

$$\inf_{u,\Gamma} E_{ms}[u,\Gamma,u_0] = \sum_{i=1}^N \int_{\Omega_i} |u_0 - c_i|^2 + \mu |\Gamma|,$$

25



Fig. 22. The image of the brain with initial contours are in section (upper row) and the final output split into 4 different regions (lower row).

where  $\Gamma'_i s$  are the connected components of  $\Omega \setminus \Gamma$  and  $u = c_i$  on  $\Gamma_i$ . Rewriting this in level-set form, we see that for m level sets there are  $n = 2^m$  phases that partition the image into n regions

$$u = c_{11}H(\phi_1)H(\phi_2) + c_{12}H(\phi_1)(1 - H(\phi_2)) + c_{21}(1 - H(\phi_1))H(\phi_2) + c_{21}(1 - H(\phi_1))(1 - H(\phi_2)).$$

The Mumford–Shah segmentation becomes:

$$\begin{split} E_4[c, \varPhi|u_0] &= \int_{\varOmega} |u_0(x) - c_{11}^2 H(\phi_1) H(\phi_2) dx \\ &+ \int_{\varOmega} |u_0(x) - c_{12}|^2 H(\phi_1) (1 - H(\phi_2)) dx \\ &+ \int_{\varOmega} |u_0(x) - c_{21}|^2 (1 - H(\phi_1)) H(\phi_2) dx \\ &+ \int_{\varOmega} |u_0(x) - c_{22}|^2 (1 - H(\phi_1)) (1 - H(\phi_2)) dx \\ &+ \mu \int_{\varOmega} |\nabla H(\phi_x)| + |\nabla H(\phi_2)| dx. \end{split}$$

Minimizing the Mumford–Shah equation leads to the Euler–Lagrange equation, fixing  $\varPhi$  and minimizing c, then the reverse:



Fig. 23. A synthetic image with a noisy t-junction is segmented using two level sets [61].

$$\begin{aligned} c_{ij}(t) &= \text{ average of } u_0 \text{ on } (2i-1)\phi_1 > 0, \quad (2j-1)\phi_2 > 0, \quad i, j = 1, 2 \\ \frac{\partial \phi_1}{\partial t} &= \delta(\phi_1) \left[ \mu \nabla (\frac{\nabla \phi_1}{|\nabla \phi_1|} - ((|u_0 - c_11|^2 - (u_0 - c12)^2)H(\phi_2) \\ &- ((u_0 - c_{21})^2 - (u_0 - c_{22})^2)(1 - H(\phi_2))) \right], \\ \frac{\partial \phi_2}{\partial t} &= \delta(\phi_2) \left[ \mu \nabla (\frac{\nabla \phi_2}{|\nabla \phi_2|} - ((|u_0 - c_11|^2 - (u_0 - c_{12})^2)H(\phi_1) \\ &- ((u_0 - c_{21})^2 - (u_0 - c_{22})^2)(1 - H(\phi_1))) \right]. \end{aligned}$$

The equations are effected by mean curvatures and jumps of data energy terms across the boundary. We show two examples in Figure 22, and t-junction example shows the robustness of the methods in Figure 23, the equations for which can be found in [62].

# 4.2 Piecewise Constant Level-Set Method (PCLSM)

The motivation of this model is the same as the one shown above, but to accomplish this in a single level sets. The multi-region segmentation model is defined using a single function  $\phi$  is to assume that  $\phi$  is a piecewise constant function taking the values:

$$\phi = i \text{ in } \Omega_i, i = 1, 2, ..., n.$$

The discontinuities of  $\phi$  give curves that separate the regions [36]. Using this definition of regions the minimization problem for image  $u_0$  is:

$$\min_{\mathbf{c},\phi,K(\phi)=0} F(\mathbf{c},\phi) = \int_{\Omega} |u - u_0|^2 dx + \beta \sum_{i=1}^n \int_{\Omega} |\nabla \psi_i| dx,$$

where the function  $\psi_i$  and the constraint are:

$$\psi_i = \frac{1}{\alpha_i} \prod_{k=1, k \neq i}^n (\phi - k) \text{ and } \alpha_i = \prod_{k=1, k \neq i}^n (i - k),$$
  
$$K(\phi) = \prod_{i=1}^n (\phi - i)$$

and u is defined by

$$u = \sum_{i=1}^{n} c_i \psi_i.$$

For details on calculating the minimum see [36, 12]. Updating the constant values is very ill-posed, a small perturbation in  $\phi$  can yield a large jump in **c**, putting some constraints. The benefit of this algorithm is that it can segment very noisy images, as can be seen in Figure 24. Even though the star is very noisy, PCLSM is able to segment the image.

Further work has been done that minimizes only the level-set function, not the constant values, and both gradient-descent and Newton's method are used to solve the Euler-Lagrange differential equations [58]. An example is shown for a two-phase image segmentation. A landscape that has some complicated shapes is segmented using both Newton's method and gradientdescent method in Figure 25.

#### 4.3 Multi-Layer Active Contours without Edges

The multi-layer method uses a single  $\phi$  with layers. The idea was inspired by multilayer techniques for modeling epitaxial growth [5]. The minimization described is non-convex, non-unique, and works locally, but the implementation is simple and the results are good.

Below we show the energy equation for a single function  $\phi$  with m levels  $l_1 < l_2 < ... < l_m$ . This will split the image into m+1 regions with the following boundaries:

$$R_m = x \in \Omega; l_{m-1} < \phi(x) < l_m$$

The energy functional for this layering is as follows:



**Fig. 24.** (a) Observed image  $u_0$  (SNR about 10.6). (b) Initial level set  $\phi$ , (c) Different phases using PCLSM where  $\phi = 1 \lor 2 \lor 3 \lor 4$  are depicted as bright regions. (d) View of  $\phi$  at convergence. for further details see [36].



Fig. 25. From left to right: observed image, segmentation using Newton's method, and segmentation using gradient descent.



Fig. 26. Segmentation of a noisy real blood cells image using one level-set function and two levels, for further details see [13].

$$\begin{split} \inf_{c_1,...c_m+1,\phi} F(c_1,c_2,...c_{m+1},\phi) &= \int_{\Omega} |f(x) - c_1|^2 H(l_1 - \phi(x)) dx \\ &+ \sum_{i=2}^m \int_{\Omega} |f(x) - c_i|^2 H(\phi(x) - l_i) dx \\ &+ \int_{\Omega} |f(x) - c_{m+1}|^2 H(\phi(x) - l_m) dx \\ &+ \mu \sum_{i=1}^m \int_{\Omega} |\nabla H(\phi - l_i)| dx. \end{split}$$

The Euler–Lagrange equations are as follows:

$$c_{1}(t) = \frac{\int_{\Omega} f(x)H(l_{1} - \phi(x, t))dx}{\int_{\Omega} H(l_{1} - \phi(x, t))dx},$$
  

$$c_{i}(t) = \frac{\int_{\Omega} f(x)H(\phi(x, t) - l_{i-1})H(l_{i} - \phi(x, t))dx}{\int_{\Omega} H(\phi(x, t) - l_{i-1})H(l_{i} - \phi(x, t))dx},$$
  

$$c_{m+1}(t) = \frac{\int_{\Omega} |f(x) - c_{1}|^{2}H(l_{1} - \phi(x, t))dx}{\int_{\Omega} H(\phi(x, t) - l_{m})dx}.$$

For further algorithmic development see [13]. In Figure 26 a noisy image of a red blood cell is segmented.

# 5 Fast Algorithms

The image processing techniques described above are very promising, but they could be somewhat slow even on simple images, because the model iterates

until it comes to a stable solution. Ways to speed up the algorithms have therefore been discussed in a number of papers.

### 5.1 Direct Optimization

One solution by [44] is to solve the partial differential equation in a narrow band, close to where the level set is zero. Another possibility proposed by [62] is to simply use implicit methods and take large steps. Multigrid methods have been developed [59]. New ideas that have been developed over the last several years include operator splitting by [26], direct optimization [19, 55], and threshold dynamics.

One approach that has been developed is to use the level-set function, without solving any differential equations. For problems that are formulated using level sets  $\phi$  and can be written in the form:

# $\min F(H(\phi)),$

the values of the objective function F are calculated directly. F does not need to be differentiable, which allows an extra degree of freedom in picking a model. The values of the level set is not needed, just the sign. Instead of evolving the differential equation, one can calculate the original objective function, then note the changes to the objective function if the sign of the levelset function is changed for the particular pixel. The algorithm follows three straightforward steps. It is initialized and objective function F is calculated for the initial partition of  $\phi > 0$  and  $\phi < 0$ . For each point, x in the image, if the energy F decreases, then change  $\phi(x)$  to  $-\phi(x)$ . Continuing to recalculate F through the image until the energy F remains unchanged.

The requirements of this algorithm are satisfied by the Chan–Vese model. The algorithm for the Chan–Vese model follows the three-step process described above. When a local change to  $\phi(x)$  occurs, the global values of the energy can be changed with a local calculation. For two-phase images it is proven in [55] that this algorithm converges in one sweep independently of the sweep order. It was further proven by [20] that this holds for images with small noise. In Figure 5.1, the convergence occurs in four steps.

#### 5.2 Operator Splitting

Another fast method that was developed by Gabou and Fedkiw [26] also uses only the sign of the level-set function rather than the value. It splits the curvature from the data-fidelity term. First, it calculates the Euler–Lagrange equation without the length term. This allows the method to take large time steps. The length term is handled by a separate step.

• Discarding the length term in the Euler–Lagrange equation, let

$$V(x) = \frac{\partial \phi}{\partial t} = -\lambda_1 (u - c_1)^2 + \lambda_2 (u - c_2)^2$$



Fig. 27. A synthetic noisy image is segmented in four iterations, which are shown. [55].

- If  $V(x)\phi(x) < 0$  then  $\phi(x) = -\phi(x)$ .
- There is an anisotropic diffusion step which then handles noise.

This method takes large time steps and so it converges quickly. Finally, there is a decrease in energy at each time step.

#### 5.3 Threshold Dynamics

More recently, work has been done by Esedoglu and Tsai [19], which uses threshold dynamics. This is motivated by a phase-field version of the twophase piecewise constant Mumford–Shah model. This yields the following gradient-descent equation for u:

$$u_t = 2\epsilon \Delta u - \frac{1}{\epsilon} |W'(u) - 2\lambda [u(c_1 - f)^2 + (u - 1)(c_2 - f)^2)],$$

where  $W(\psi) = \psi^2 (1 - \psi)^2$ .

Using the method developed by Merriman, Bence, and Osher (MBO) [2], the method alternates between a linear parabolic partial differential equation and thresholding:

• Let  $v(x) = S(\delta t)u_n(x)$ , where  $S(\delta t)$  is the propagator of the linear equation

$$w_t = \Delta w - 2\lambda [w(c_1 - f)^2 + (w - 1)(c_2 - f)^2].$$

Some Recent Developments in Variational Image Segmentation 33

• Set

$$u_{n+1}(x) = \left\{ 0 \text{ if } v(x) \in (-\infty, \frac{1}{2}), 1 \text{ if } v(x) \in (\frac{1}{2}, \infty) \right\}$$

This method is fast because the first step is calculated quickly using an fast Fourier transform, and the second step is a threshold. A higher-order scheme has been developed in [18].

# 6 Acknowledgment

We would like to thank Selim Esedoglu, Richard Tsai, Luminita Vese, Xue-Cheng Tai, and Jason Chung for their support.

# References

- 1. L. Ambrosio and V. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via  $\gamma$  convergence. Comp. Applied Math., 63:707–711, 1990.
- B. Merriman, J. K. Bence, and S. J. Osher. Diffusion generated motion by mean curvature. *Proceedings of the Comutational Crystal Growers Workshop*, AMS:73–83, 1992.
- B. Merriman, J. K. Bence, and S. J. Osher. Motion of multiple junctions: A level set approach. J. Comput. Phys, 112, 1994.
- P. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. IEEE Trans. Comm., 31:532–540, 1983.
- R. E. Caflisch, M. F. Gyure, B. Merriman, S. Osher, C. Rasch, D. D. Vedonsky, and J. J. Zinck. Island dynamics and the level set method for epitaxial growth. *Appl. Math. Letters*, 12(4):13, 1999.
- S. Casadei, S. Mitter, and P. Perona. Boundary detection in piecewise homogeneous textured images. Lect. Notes Comput. Sci., 588:174–183, 1992.
- V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Num. Math.*, 66:1–31, 1993.
- V. Caselles, R. Kimmel, and G. Sapiro. Geodisic active contours. Int. J. Comp. Vis., 22(1):61–79, 1997.
- T. Chan, B. Sandberg, and L. Vese. Active contours without edges for vectorvalued images. J. Vis. Comm. Image Represent., 11:130–141, 1999.
- T. Chan, B. Sandberg, and L. Vese. Active contours without edges for textured images. CAM report 02-39, 2002.
- T. Chan and L. Vese. Active contours without edges. *IEEE Trans.Image Process.*, 16(2):266–277, 2000.
- 12. O. Christiansen and X.C. Tai. Fast implementation of the piecewise constant level set method. In *this book*.
- J. T. Chung and L. A. Vese. Image segmentation using a multilayer level-set approach. UCLA Cam Report 03-53, http://www.math.ucla.edu/applied/cam/index.html, 2001.
- M. Clark, A. C. Bovik, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):55–73, 1990.

- J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filter. J. Opt. Soc. Am. A, 2(7):1160–1169, 1985.
- 16. F. Dibos and G. Koepfler. Color segmentation using a variational formulation. Actes du 16me Colloque GRETSI, pages 367–370, 1997.
- D. Dunn, W. E. Higgins, and J. Wakeley. Texture segmentation using 2-d Gabor elementary functions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2), 1994.
- S. Esedoglu, S. Ruuth, and Y. H. Tsai. Threshold dynamics for high order geometric motions. UCLA CAM Report 06-23, http://www.math.ucla.edu/applied/cam/index.html, 2006.
- S. Esedoglu and Y. H. Tsai. Threshold dynamics for the piecewise constant Mumford–Shah functional. J. Comput. Phys. 211(1):367–384, 2006.
- T. F.Chan and S. Esedoglu. A multiscale algorithm for Mumford–Shah image segmentation. UCLA CAM Report 03-77, 2003.
- T. F.Chan and S. Esedoglu. Aspects of total variation regularized L<sup>1</sup> function approximation. SIAM J. Appl. Math. 65(5):1817–1837, 2005.
- T. F.Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, 66(5):1632–1648, 2006.
- I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biol. Cybern.*, 61:103–113, 1989.
- D. Gabor. Theory of communication. Journal of the Institute of Electrical Engineers, 93:429–457, 1946.
- S. Gao and T.D. Bui. A new image segmentation and smoothing model. *ISBI*, pages 137–140, 2004.
- F. Gibou and R. Fedkiw. A fast hybrid k-means level set algorithm for segmentation. In "4th Annual Hawaii International Conference on Statistics and Mathematics", pp. 281-291, 2005. Also: *Stanford Technical Report 02-08*, 2002.
- F. Guichard. A morpholocial affine and Galilean invarient scale space for movies. IEEE Trans. Image Process., 7(3):444–456, 1998.
- Portilla J. and Simoncelli E.P. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–71, 2000.
- A. K. Jain and F. Farrakhonia. Unsupervised texture segmentation using Gabor filters. *Pattern Recogn.*, 23(12):1167–1186, 1991.
- M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contours model. Int. J. Comput. Vis., 1:1167–1186, 1991.
- S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. *Proc. of ICCV, Cambridge*, pages 810–815, 1995.
- G. Koepfler, C. Lopez, and J.M. Morel. A multiscale algorithm for image segmentation by variational method. SIAM J. Numer. Anal., 31(1):282–299, 1994.
- G. Koepfler, C. Lopez, and L. Rudin. Data fusion by segmentation. application to texture discrimination. Actes du 14me Colloque GRETSI, Juan-les-Pins, September, pages 707–710, 1993.
- T.S. Lee, D. Mumford, and A. Yuille. Texture segmentation by minimizing vector-valued energy functionals - the coupled-membrane model. *Lect. Notes Comput. Sci.*, 588:165–173, 1992.
- 35. J. Lie, M. Lysaker, and X.C. Tai. A binary level set model and some applications to Mumford–Shah image segmentation. *IEEE Trans. Image Process.*,15(5):1171–1181, 2006.

- J. Lie, M. Lysaker, and X.C.Tai. A variant of the level set method and applications to image segmentation. *Math. Comp.*, 75:1155–1174, 2006.
- J. Malik and P. Perona. A computational model of texture segmentation. Proceedings of CVPR '89, pages 326–332, 1989.
- R. Malladi, J. A. Sethian, and B. C. Vemuri. A topology independent shape modeling scheme. Proc. SPIE Conf. on Geometric Methods in Computer Vision II, San Diego, 2031:246–258, 1993.
- 39. G. Dal Maso, J.-M. Morel, and S. Solimini. A variational method in image segmentation. existence and approximation results. *Acta Math.*, 63.
- M. Moelich and T. Chan. Joint segmentation and registration using logic models. J. Vis. Commun. Image R., 15:333–358,2005.
- M. Moelich and T. Chan. Tracking objects with the Chan–Vese algorithm. CAM Reports 03-14, www.math.ucla.edu/applied/cam/index.shtml(03-14), March 2003.
- 42. M. Moelich. Logic Models for Segmentation and Tracking. *Thesis*, UCLA Mathematics Department, 2004.
- D. Mumford and J. Shah. Optimal approximation by piecewise-smooth functions and associated variational problems. *Commun. Pure Appl. Math.*, 42:577– 685, 1989.
- 44. S. Osher and R. Fedkiw. Level set methods and dynamic implicit surfaces. Springer-Verlag, 2003.
- S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulation. J. Comput. Phys., 79:12–49, 1988.
- 46. N. Paragios and R. Deriche. Geodesic active contours for supervised texture segmentation. Proceedings of the International Conference on Computer Vision and Pattern Recognition, June Fort Collins, Colorado, 1999.
- N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. Proceedings of the 7th International Conference on Computer Vision, pages 100–115, 1999.
- C. Sagiv, N.A. Sochen, and Y.Y. Zeevi. Geodesic active contours applied to texture feature space. M. Kerckhove (Ed.): Scale-Space 2001, LNCS 2106, pages 344–352, 2001.
- B. Sandberg and T. Chan. Logic operators for active contours on multi-channel images. J. Vis. Commun. Image. R.,16:333-358, 2005.
- 50. G. Sapiro. Color snakes. Comput. Vis. Image Understand., pages 247-253, 1997.
- 51. G. Sapiro. Geometric partial differential equations and image analysis. Cambridge University Press, 2001.
- G. Sapiro. Vector (self) snakes: a geometric framework for color, texture, and multiscale image segmentation. *Proc. IEEE ICIP, Lausanne*, I:817–820, September 1996.
- 53. G. Sapiro and D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans. Image Process.*, 5:1582–1586, 1996.
- Zhu S.C., Lee T.S., and A.L. Yuille. Region competition: Unifying snakes, region growing, energy/bayes/mdl for multi-band image segmentation. *Proceedings of* the IEEE 5th ICCV, Cambridge, pages 416–423, 1995.
- 55. B. Song and T. Chan. A fast algorithm for level set based optimization. UCLA CAM Report 02-68, http://www.math.ucla.edu/applied/cam/index.html, 2002.
- P. H. Suen and G. Healey. Modeling and classifying color textures using random fields in a random environment. *Pattern Recogn.*, 32(6):1009–1017, 1999.

- 57. P. H. Suen and G. Healey. The analysis and recognition of real-world textures in three dimensions. *IEEE PAMI*, 22(5):491–503, 2000.
- X.-C. Tai and C. H. Yao. Image segmentation by piecewise constant Mumford– Shah model without estimating the constants. J. Comput. Math., 24(3):435–443, 2006.
- 59. A. Tsai, A. Willsky, and A. Yezzi. Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans. Image Process.*, 10(8):1169–1186, August 2001.
- D.-M. Tsai, S.-K. Wu, and M.-C. Chen. Optimal gabor filter design for texture segmentation using stochastic optimization. *Image Vis. Comput.*, 19:299–316, 2001.
- 61. L. Vese and T. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. Comput. Vis.*, 2002.
- 62. L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. Comput. Vis.*, 2001.
- T. P. Weldon and W. E. Higgins. Design of multiple gabor filters for texture segmentation. *Proc. ISCAS*, 1996.
- T. P. Weldon and W. E. Higgins. An algorithm for designing multiple Gabor filters for segmenting multi-textured images. *IEEE Conference on Image Processing, Chicago, IL, Oct.* 4-7, 1998.
- A. Yezzi, L. Zollei, and T. Kapur. A variational approach to joint segmentation and registration. *IEEE Conf. on Comp. Vision and Pattern Recognition*, pages 810–815, 2001.
- H. K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. J. Comput. Phys., 127:179–195, 1996.
- S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18:884–900, 1996.