

# A Variational Framework for Multi-Region Pairwise Similarity-based Image Segmentation

Luca Bertelli, *Student Member, IEEE*, Baris Sumengen, *Member, IEEE*,  
B. S. Manjunath, *Fellow, IEEE*, Frédéric Gibou

**Abstract**— Variational cost functions that are based on pairwise similarity between pixels can be minimized within level set framework resulting in a binary image segmentation. In this paper we extend such cost functions and address multi-region image segmentation problem by employing a multi-phase level set framework. For multi-modal images cost functions become more complicated and relatively difficult to minimize. We extend our previous work [1], proposed for background/foreground separation, to the segmentation of images into more than two regions. We also demonstrate an efficient implementation of the curve evolution, which reduces the computational time significantly. Finally, we validate the proposed method on the Berkeley Segmentation Data Set by comparing its performance with other segmentation techniques.

**Index Terms**— Region-based image segmentation, grouping, level sets, multi-phase motion, pairwise similarity measure.

## I. INTRODUCTION

IN this paper we present a variational approach to multi-region segmentation that is based on pairwise pixel similarity. Pairwise similarity-based cost functions have been extensively used in the literature, primarily for background/foreground segmentation [2], [3], [1]. Extensions to multiple regions is then obtained by recursively bi-partitioning the regions, which may not lead to a good overall segmentation. In contrast, the method that we propose explicitly starts with the goal of segmenting the image into more than two regions, and we derive the appropriate evolution equations that result in the desired partitioning. It combines the advantages of pairwise pixel similarity based cost functions—their ability to embed heterogeneous information derived from different image cues—with the flexibility of the variational methods to deal with multiple regions, into a single, well defined framework for image segmentation.

A pairwise similarity based variational framework was first introduced in [1] for the case of two-region segmentation. However, as we will see in the following, extension of this to multi-region segmentation is not straightforward and requires a reformulation of the cost functions. Using the notation  $w(p_1, p_2)$  to represent the pairwise dissimilarity between point  $p_1$  and point  $p_2$  (where  $p_i$  is a 2D point in the image domain)

we can write this variational cost function as:

$$E(C) = - \iint_{p_1 \in R_o(C)} \iint_{p_2 \in R_i(C)} w(p_1, p_2) dp_1 dp_2 \quad (1)$$

where  $R_i(C)$  and  $R_o(C)$  are respectively the regions inside and outside of the curve  $C$ . The minimization of  $E(C)$  with respect to the curve  $C$  leads to a partitioning of the image, which maximizes the dissimilarity between regions  $R_i(C)$  and  $R_o(C)$ . Steepest descent with respect to the curve  $C$  yields the following curve evolution equation:

$$\frac{\partial C}{\partial t} = \left( \iint_{p \in R_o(C(t))} w(c, p) dp - \iint_{p \in R_i(C(t))} w(c, p) dp \right) \vec{N} \quad (2)$$

where  $c$  is a point on the curve  $C$ . Thus, every pixel on the curve  $C$  is compared with the interior and the exterior of the curve in terms of their similarity. The curve is then expanded or shrunk accordingly in the normal direction. At steady state the region inside of  $C$  and the region outside of it are the segmented background and foreground of the image.

The extension of this above framework for multiple region segmentation is not straightforward. In fact, the direct minimization with respect to the explicit representation of the curve  $C$ , as in [1], does not allow a mathematically sound extension to the multi-region case. This leads to one key contribution of the work, the reformulation in terms of pairwise dissimilarity within the regions (instead of across-region cuts as in [1]). This allows us to use the level set formulation of curves, as proposed by Osher and Sethian [4], and its extension to multiple regions [5]. In [4], a curve  $C$ , the boundary of an open set  $\omega \in \Omega$  (*i.e.*  $C = \partial\omega$ ), is implicitly represented as the zero level set of a continuous Lipschitz function  $\phi : \Omega \mapsto \mathbb{R}$ . The function  $\phi$  is positive for the points within the set  $\omega$  and negative elsewhere (*i.e.* for the points within  $\Omega \setminus \omega$ ). Therefore the Heaviside function  $H(\phi)$ , along with its complementary  $(1 - H(\phi))$ , can serve as indicator function for the points in  $\omega$  and  $\Omega \setminus \omega$  respectively [6].

$$H(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{elsewhere} \end{cases} \quad (1 - H(\phi)) = \begin{cases} 1 & \text{if } \phi < 0 \\ 0 & \text{elsewhere} \end{cases} \quad (3)$$

In [5] the authors showed how  $k$  level set functions can be used to construct up to  $n = 2^k$  different indicator functions and therefore to represent up to  $n$  different regions. We will make use of this binary/multiphase level set representation to reformulate/extend the similarity based segmentation framework introduced in [1].

The main contributions of this paper are:

Luca Bertelli and B.S. Manjunath are with the Department of Electrical and Computer Engineering at the University of California, Santa Barbara (emails {lbartelli,manj}@ece.ucsb.edu). Baris Sumengen is with Like.com (email sumengen@ece.ucsb.edu). Frédéric Gibou is with the Department of Mechanical Engineering and the Department of Computer Science at the University of California, Santa Barbara (email fgibou@engineering.ucsb.edu).

- Design of variational segmentation cost functions suitable for multi-region segmentation of images. This consists of two steps:
  - 1) Reformulation using within-region dissimilarities rather than across-region cuts.
  - 2) Representation of several regions and complex topologies via a binary/multiphase level set framework.
- Derivation of a steepest descent minimization of these cost functions that can be implemented using a region-based active contour model and demonstration of an efficient implementation of the curve evolution.
- Introduction of an efficient regularization technique based on over-segmentation, which significantly reduces the computational time. The regularization is performed locally and adaptively, leading to improved segmentation results.

The rest of the paper is organized as follows. In Section II we briefly review related work in level sets based image segmentation using region-based approaches. In Section III, IV and V we present pairwise similarity based cost functions and derive the corresponding minimizations for the two-region and multi-region cases. In Section VI and VII we present regularization techniques and a fast implementation of our algorithms. Experimental results are shown in Section VIII. In Section IX, we discuss the estimation of number of phases required for segmenting an image and then conclude.

## II. PREVIOUS WORK ON REGION-BASED LEVEL SET METHODS

Region-based active contour models [7], [8], [6], [9], [10], [11] have become increasingly popular in the past ten years. The main idea is to evolve a curve using information from the statistics of the interior and the exterior of the curve such that at the end of the evolution the active contour coincides with the boundaries of the objects. Compared to edge driven approaches [12], [13], region based approaches are more appealing because of their less dependency on edge detection, which can be sensitive to noise and clutter. Moreover, region-based active contours are less constrained by the initial configuration of the contour, since they incorporate both local and global information of the image statistics.

A region based model for variational image segmentation by minimizing a cost function was proposed by Mumford and Shah in [14]. Chan and Vese minimized this functional, using level set methods for both piecewise constant [6] and piecewise smooth [5] approximations of the image. The cost function for the piecewise constant case is:

$$E(C) = \lambda_1 \iint_{R_i(C)} |I(p) - c_1|^2 dp + \lambda_2 \iint_{R_o(C)} |I(p) - c_2|^2 dp + \mu \text{Length}(C) + \nu \text{Area}(R_i(C)) \quad (4)$$

where  $I$  is the image,  $R_i(C)$  and  $R_o(C)$  are respectively the regions inside and outside the curve  $C$  ( $C(p) : [0, 1] \rightarrow \mathbb{R}^2$ ),

$c_1$  and  $c_2$  are constants and  $\lambda_1, \lambda_2, \mu, \nu$  are fixed parameters. This functional was then expressed by the authors within the level set formulation developed by Osher and Sethian in [4]. Minimizing this cost function with respect to  $c_1, c_2$  and the curve  $C$  gives the separation of the image in two regions and the corresponding evolution equation for  $C$  becomes:

$$\frac{\partial C}{\partial t} = \left[ -\lambda_1 |I - c_1|^2 + \lambda_2 |I - c_2|^2 + \mu\kappa + \nu \right] \vec{N} \quad (5)$$

The constant  $c_1$  and  $c_2$  come out to be the mean intensity values of the regions  $R_i(C)$  and  $R_o(C)$ . This model was only suitable for segmentation of a gray scale image into two regions, one representing the object to be detected and the second one the background. The model was then extended to vector-valued images in [15] and to the segmentation of more than two regions in [5].

Yezzi et al. in [9] used a different optimization criterion based on maximizing the separation of the mean values of the two regions. The corresponding cost function can be expressed as:

$$E(C) = -\frac{1}{2}(m_1 - m_2)^2 \quad (6)$$

where  $m_1$  and  $m_2$  are the mean intensity values of the regions  $R_i(C)$  and  $R_o(C)$ . Minimizing this functional using a gradient descent approach leads to the equation for the curve evolution:

$$\frac{\partial C}{\partial t} = (m_1 - m_2) \left( \frac{I - m_1}{A_1} + \frac{I - m_2}{A_2} \right) \vec{N} \quad (7)$$

where  $A_1$  and  $A_2$  are the areas of the two regions. A stabilizing term proportional to the curvature ( $\alpha\kappa\vec{N}$ ) is then added to prevent the contour from wrapping around isolated noisy pixels.

Paragios and Deriche introduced Geodesic Active Regions, which revisited the region-based segmentation in a probabilistic framework [16], [17]. The corresponding cost function is composed of two different terms, a region-based and a boundary related term. Let  $p(I(q)|B)$  be the conditional boundary density function, which measures the probability of the point  $q$  belonging to the real object boundaries. The boundary term of the cost function can be expressed as:

$$E_B(C) = \int_0^1 g(p(I(C(p))|B)) |\dot{C}(p)| dp \quad (8)$$

where, as before,  $C(p) : [0, 1] \rightarrow \mathbb{R}^2$  is a 2D parameterization of the region boundaries and  $g(t)$  is a positive monotonic decreasing function, such that  $g(0) = 1$  and  $g(t) \rightarrow 0$  as  $t \rightarrow \infty$ . The region-based term of the cost function is:

$$E_R(C) = -\sum_{i=1}^2 \iint_{R_i} \log(p(I(x, y)|R_i)) dx dy \quad (9)$$

Minimization of  $E_{TOT} = (1 - \alpha)E_B - \alpha E_R$ , using a gradient descent method leads to a curve evolution equation:

$$\frac{\partial C}{\partial t} = \left[ \alpha \log \left( \frac{p(I(C)|R_2)}{p(I(C)|R_1)} \right) + \right. \quad (10)$$

$$\left. - (1 - \alpha)(g(C)\kappa + \nabla g(C)\vec{N}) \right] \vec{N} \quad (11)$$

More recent works include [18], where Aubert et al. revisited the problem of region-based image segmentation functional

minimization using the concept of shape derivatives. In [19], [20], [21] Cremers et al. proposed a new variational framework to integrate shape priors information into the region-based segmentation approach. Other relevant work in prior-based segmentation using level sets includes [22], [23], [24].

In this paper we work with variational cost functions based on pairwise pixel similarities (or dissimilarities). The main asset of this approach is the ability to embed heterogeneous information coming from different image cues into the similarity measure. The similarity measure can be defined as a function of color distances, or in the case of texture, can be computed in the feature space generated by Gabor filter outputs [25]. Several approaches have also been proposed to integrate edge information into similarity measure such as considering edge strength along the lines connecting two pixels (intervening contour) [26], or using geodesic distances between pixels [27].

### III. BACKGROUND/FOREGROUND SEGMENTATION

In this section we revisit the steepest descent minimization of the cost function given in (1). We will first formulate this cost function in a slightly different way. This will help us with the extension to the multi-region case, which is given in Section IV.

Define:

$$diss(A, A) = \iint_{p_1 \in A} \iint_{p_2 \in A} w(p_1, p_2) dp_1 dp_2 \quad (12)$$

and

$$cut(A, B) = \iint_{p_1 \in A} \iint_{p_2 \in B} w(p_1, p_2) dp_1 dp_2 \quad (13)$$

where  $w(p_1, p_2)$  is a dissimilarity metric,  $p_1 = (x_1, y_1)$  and  $p_2 = (x_2, y_2)$  are 2D points inside the image and  $A, B$  form a partitioning such that  $A \cup B = \Omega$ , where  $\Omega$  represents the whole image domain. We can now write (1) as:

$$E = -cut(A, B) \quad (14)$$

*Lemma:*

Let

$$E' = \frac{1}{2}(diss(A, A) + diss(B, B)) \quad (15)$$

Minimizing  $E'$  is equivalent to minimizing  $E$ .

*Proof:*

$$cut(A, B) = diss(A, \Omega) - diss(A, A)$$

equivalently

$$cut(A, B) = diss(B, \Omega) - diss(B, B)$$

we can write:

$$2E = -2cut(A, B) = diss(A, A) - diss(A, \Omega) + diss(B, B) - diss(B, \Omega)$$

Note that

$$diss(A, \Omega) + diss(B, \Omega) = diss(\Omega, \Omega)$$

which is independent of the partitioning and therefore can be eliminated from the cost function.

$$2E = diss(A, A) + diss(B, B) - \cancel{diss(\Omega, \Omega)}$$

This concludes the proof.  $\blacksquare$

We name the cost function (15) as *Total Dissimilarity*.

Define a 3D surface  $\phi$  such that its zero level set is the curve  $C$ , which is the boundary between foreground and background. We can write:

$$\begin{aligned} C &= \{(x, y) \in \Omega | \phi(x, y) = 0\} \\ A &= \{(x, y) \in \Omega | \phi(x, y) > 0\} \\ B &= \{(x, y) \in \Omega | \phi(x, y) < 0\} \end{aligned}$$

Now using the Heaviside function  $H(z)$ , equal to 1 if  $z > 0$  and 0 if  $z < 0$ , we can rewrite (15) as follows (dropping the constant):

$$\begin{aligned} E &= \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) H(\phi(p_2)) dp_1 dp_2 + \\ &\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) (1 - H(\phi(p_2))) dp_1 dp_2 \end{aligned} \quad (16)$$

The gradient projection method minimizing  $\int f(\phi(x)) dx$ , using  $t$  as the descent variable leads to (See Appendix A for proof) [28]:

$$\frac{\partial \phi}{\partial t} = - \frac{\partial f}{\partial \phi} \quad (17)$$

In our case

$$\begin{aligned} f(p_2) &= \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) H(\phi(p_2)) dp_1 \\ &+ \iint_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) (1 - H(\phi(p_2))) dp_1 \end{aligned} \quad (18)$$

Applying (17) to (18) yields:

$$\begin{aligned} \frac{\partial \phi(p_2)}{\partial t} &= - \frac{\partial f(p_2)}{\partial \phi} \\ &= \iint_{\Omega} w(p_1, p_2) \left[ -\delta(\phi(p_1)) H(\phi(p_2)) + \right. \\ &\quad \left. - H(\phi(p_1)) \delta(\phi(p_2)) \right] dp_1 + \\ &\iint_{\Omega} w(p_1, p_2) \left[ \delta(\phi(p_1)) \left( 1 - H(\phi(p_2)) \right) + \right. \\ &\quad \left. \left( 1 - H(\phi(p_1)) \right) \delta(\phi(p_2)) \right] dp_1 \end{aligned} \quad (19)$$

Rearranging the terms we get:

$$\begin{aligned} \frac{\partial \phi(p_2)}{\partial t} &= - \iint_{\Omega} w(p_1, p_2) \left[ \delta(\phi(p_1)) H(\phi(p_2)) \right] dp_1 \\ &+ \iint_{\Omega} w(p_1, p_2) \left[ \delta(\phi(p_1)) \left( 1 - H(\phi(p_2)) \right) \right] dp_1 \\ &- \iint_{\Omega} w(p_1, p_2) \left[ H(\phi(p_1)) \delta(\phi(p_2)) \right] dp_1 \\ &+ \iint_{\Omega} w(p_1, p_2) \left[ \left( 1 - H(\phi(p_1)) \right) \delta(\phi(p_2)) \right] dp_1 \end{aligned} \quad (20)$$

We can furthermore simplify (20) as:

$$\begin{aligned} \frac{\partial \phi(p_2)}{\partial t} = & - \int \int_C w(p_1, p_2) H(\phi(p_2)) dp_1 \\ & + \int \int_C w(p_1, p_2) (1 - H(\phi(p_2))) dp_1 \\ & - \int \int_{\Omega} w(p_1, p_2) [H(\phi(p_1)) \delta(\phi(p_2))] dp_1 \\ & + \int \int_{\Omega} w(p_1, p_2) [(1 - H(\phi(p_1))) \delta(\phi(p_2))] dp_1 \end{aligned} \quad (21)$$

where we can discard the first two terms since they are integrals calculated over a set of measure zero, and therefore are negligible with respect to the last two terms.

The curve evolution, corresponding to the steepest descent, becomes:

$$\begin{aligned} \frac{\partial \phi(p_2)}{\partial t} = & \delta(\phi(p_2)) \left[ \int \int_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) dp_1 \right. \\ & \left. - \int \int_{\Omega} w(p_1, p_2) H(\phi(p_1)) dp_1 \right] \end{aligned} \quad (22)$$

This is equivalent to the curve evolution equation (2) given in Section I.

#### IV. MULTI-REGION SEGMENTATION

By extending (15) to the multi-region case, the cost function becomes the following:

$$E = \sum_i^n \int \int_{p_1 \in A_i} \int \int_{p_2 \in A_i} w(p_1, p_2) dp_1 dp_2 \quad (23)$$

where  $n$  is the number of regions. The goal therefore is to minimize the dissimilarity within the regions. In this case we use  $k = \log_2 n$  level set functions to represent  $n$  phases (or regions) with potentially complex topologies, such as triple junctions [5]. Formation of vacuums and overlaps are avoided since the partition will be a disjoint and exhaustive decomposition of  $\Omega$ . We introduce  $k = \log_2 n$  level set functions  $\phi_i$ , such that the union of their zero level sets is the contours of the segmented regions. Define the level set vector  $\Phi^k = (\phi_1, \phi_2, \dots, \phi_k)^T$ . At each point  $H(\Phi^k(x, y))$  is a binary vector, whose elements are either 0 or 1. Two pixels  $(x_1, y_1)$  and  $(x_2, y_2)$  belong to the same region if and only if  $H(\Phi^k(x_1, y_1)) = H(\Phi^k(x_2, y_2))$ . This vector can only take  $n = 2^k$  different values and can be used for representing up to  $n$  different regions.

Let us call  $\mathbf{b}^k$  the set of all the possible  $n$  configurations of this  $k$ -component binary vector

$$\mathbf{b}^k = \{\mathbf{b}_1^k, \mathbf{b}_2^k, \dots, \mathbf{b}_n^k\}$$

where

$$\mathbf{b}_i^k = (b_{i1}^k, b_{i2}^k, \dots, b_{ik}^k)^T$$

and  $b_{ii}^k = \{1, 0\}$ . Now we can define  $n = 2^k$  characteristic functions, one for each of the  $n$  regions (i.e. a function, which

takes a value of 1 if the pixels belongs to that region and 0 otherwise) as follows:

$$\chi_i = \prod_{l=1}^k H(\phi_l)^{b_{li}^k} (1 - H(\phi_l))^{(1-b_{li}^k)} \quad (24)$$

for  $i = 1, \dots, n$  (Note that  $0^0 = 1$ ). We can now rewrite the cost function (23) as follows:

$$E = \sum_i^n \int \int_{\Omega} \int \int_{\Omega} w(p_1, p_2) \chi_i(p_1) \chi_i(p_2) dp_1 dp_2 \quad (25)$$

Before proceeding with the minimization of the cost in (25) we introduce another notation. Define a level set vector  $\Phi^{k-1, i}$  by removing  $\phi_i$  from the vector  $\Phi^k$ .

$$\Phi^{k-1, i} = (\phi_1, \dots, \phi_{i-1}, \phi_{i+1}, \dots, \phi_k)^T$$

$H(\Phi^{k-1, i})$  is a binary vector which can only take  $\frac{n}{2} = 2^{k-1}$  different values. Let  $\mathbf{b}^{k-1}$  be the set of all these  $\frac{n}{2}$  possible configurations.

$$\mathbf{b}^{k-1} = \{\mathbf{b}_1^{k-1}, \mathbf{b}_2^{k-1}, \dots, \mathbf{b}_{\frac{n}{2}}^{k-1}\}$$

where

$$\mathbf{b}_i^{k-1} = (b_{i1}^{k-1}, b_{i2}^{k-1}, \dots, b_{i, k-1}^{k-1})^T$$

and  $b_{ii}^{k-1} = \{1, 0\}$ . We can now define  $\chi_j^i$  as

$$\begin{aligned} \chi_j^i = & \prod_{l=1}^{i-1} H(\phi_l)^{b_{jl}^{k-1}} (1 - H(\phi_l))^{b_{jl}^{k-1}} \\ & \cdot \prod_{l=i+1}^{k-1} H(\phi_l)^{b_{jl}^{k-1}} (1 - H(\phi_l))^{b_{jl}^{k-1}} \end{aligned}$$

for  $j = 1, \dots, n/2$ . Now reasoning along the same lines of the previous section (two region case), we can obtain the curve evolution equation for each  $\phi_i$ . The steepest descent minimization of (25) is then:

$$\begin{aligned} \frac{\partial \phi_i(p_2)}{\partial t} = & \delta(\phi_i(p_2)) \left[ \sum_{j=1}^{n/2} \chi_j^i(p_2) \right. \\ & \left( - \int \int_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) dp_1 + \right. \\ & \left. \left. \int \int_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) dp_1 \right) \right] \end{aligned} \quad (26)$$

See Appendix D for the 4-region case ( $k = 2$ ).

#### V. NORMALIZED FRAMEWORKS

The cost function defined in (15) is a particular case of the following cost function:

$$E = \alpha \text{diss}(A, A) + \beta \text{diss}(B, B)$$

where  $\alpha, \beta$  are two constants such that  $\alpha + \beta = 1$ . Using  $\alpha = \beta = \frac{1}{2}$ , as in (15), the minimization of the cost function becomes biased towards equal size partitions. These two parameters can be tuned in order to give more importance to one region with respect to the other one, limiting the effect of this bias. Since the final goal is the design of a self-tuning

segmentation system, the introduction of tunable parameters in the algorithm is not desirable.

In this section, we therefore present two types of normalizations of (15) and (23) in order to limit the effect of this bias without introducing any new parameters (the first type of normalization, *i.e.* normalization with respect to the area of the partitions, was introduced in [21] in the context of shape priors). We derive the equations for the two region case but these can easily be extended to the multi-region case.

#### A. Average Dissimilarity

We define *Average Dissimilarity* within regions as:

$$\begin{aligned} E_A &= \frac{1}{Area_1} \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_1(p_1) \chi_1(p_2) dp_1 dp_2 \\ &+ \frac{1}{Area_2} \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_2(p_1) \chi_2(p_2) dp_1 dp_2 \\ &= \frac{diss(A, A)}{Area_1} + \frac{diss(B, B)}{Area_2} \end{aligned} \quad (27)$$

where dissimilarity within each region is normalized by its area. Equivalently we can write:

$$\begin{aligned} E_A &= \frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) H(\phi(p_2)) dp_1 dp_2}{\iint_{\Omega} H(\phi(p)) dp} + \quad (28) \\ &\frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) (1 - H(\phi(p_2))) dp_1 dp_2}{\iint_{\Omega} (1 - H(\phi(p))) dp} \end{aligned}$$

Lemma:

The curve evolution corresponding to the steepest descent minimization of  $E_A$  (28) is:

$$\begin{aligned} \frac{\partial \phi(p_2)}{\partial t} &= - \frac{\delta(\phi(p_2)) \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) dp_1}{Area_1} \\ &+ \frac{\delta(\phi(p_2)) \iint_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) dp_1}{Area_2} \quad (29) \\ &+ \delta(\phi(p_2)) \left( \frac{diss(A, A)}{(Area_1)^2} - \frac{diss(B, B)}{(Area_2)^2} \right) \end{aligned}$$

*Proof:*

In order to minimize the cost function in (28), we need to find the first variation of elements such as:

$$\varepsilon = \frac{\int f(\phi(x)) dx}{\int g(\phi(y)) dy} \quad (30)$$

The steepest descent minimization of (30) using  $t$  as descent variable yields (See Appendix B):

$$-\phi_t(x) = \frac{f_{\phi}(\phi(x)) \int g(\phi(x)) dx - g_{\phi}(\phi(x)) \int f(\phi(x)) dx}{\left( \int g(\phi(x)) dx \right)^2} \quad (31)$$

$E_A$  can be formulated as:

$$E_A = \frac{\int f_1(\phi(x)) dx}{\int g_1(\phi(y)) dy} + \frac{\int f_2(\phi(x)) dx}{\int g_2(\phi(y)) dy}$$

Therefore (see Section III for details of the derivation)

$$\frac{\partial f_1}{\partial \phi} = \delta(p_2) \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) dp_1$$

$$\begin{aligned} \frac{\partial f_2}{\partial \phi} &= -\delta(p_2) \iint_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) dp_1 \\ \frac{\partial g_1}{\partial \phi} &= \delta(p_2) \\ \frac{\partial g_2}{\partial \phi} &= -\delta(p_2) \end{aligned}$$

Finally, putting everything together, we obtain the curve evolution equation:

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \frac{\frac{\partial f_1}{\partial \phi} Area_1 - \frac{\partial g_1}{\partial \phi} diss(A, A)}{(Area_1)^2} \\ &+ \frac{\frac{\partial f_2}{\partial \phi} Area_2 - \frac{\partial g_2}{\partial \phi} diss(B, B)}{(Area_2)^2} \end{aligned}$$

which after some algebra becomes (29) concluding the proof. ■

In the multi-region case the *Average Dissimilarity* becomes:

$$E_A = \sum_{i=1}^n \frac{1}{Area_i} \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_i(p_1) \chi_i(p_2) dp_1 dp_2 \quad (32)$$

and the corresponding evolution equation is:

$$\begin{aligned} \frac{\partial \phi_i(p_2)}{\partial t} &= \delta(\phi_i(p_2)) \left[ \sum_{j=1}^{n/2} \chi_j^i(p_2) \right. \\ &\left( - \frac{\iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) dp_1}{\iint_{\Omega} \chi_j^i(p) H(\phi_i(p)) dp} \right. \quad (33) \\ &+ \frac{\iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) dp_1}{\iint_{\Omega} \chi_j^i(p) (1 - H(\phi_i(p))) dp} \\ &\left. + (\Gamma_1 - \Gamma_2) \iint_{\Omega} w(p_1, p_2) dp_1 \right] \end{aligned}$$

where

$$\Gamma_1 = \frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) \chi_j^i(p_2) H(\phi_i(p_2)) dp_1 dp_2}{\left( \iint_{\Omega} \chi_j^i(p) H(\phi_i(p)) dp \right)^2}$$

and

$$\Gamma_2 = \frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) \chi_j^i(p_2) (1 - H(\phi_i(p_2))) dp_1 dp_2}{\left( \iint_{\Omega} \chi_j^i(p) (1 - H(\phi_i(p))) dp \right)^2}$$

#### B. Normalized Dissimilarity

Along the lines of the previous subsection, we can now define a new unbiased measure normalizing by the total dissimilarity of each region with respect to the whole domain. We borrow this concept from graph theory where it is called normalized cuts.

Therefore, the cost function for the *Normalized Dissimilarity* framework becomes:

$$\begin{aligned} E_N &= \frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_1(p_1) \chi_1(p_2) dp_1 dp_2}{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_1(p_2) dp_1 dp_2} \\ &+ \frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_2(p_1) \chi_2(p_2) dp_1 dp_2}{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_2(p_2) dp_1 dp_2} \\ &= \frac{diss(A, A)}{diss(A, \Omega)} + \frac{diss(B, B)}{diss(B, \Omega)} \end{aligned} \quad (34)$$

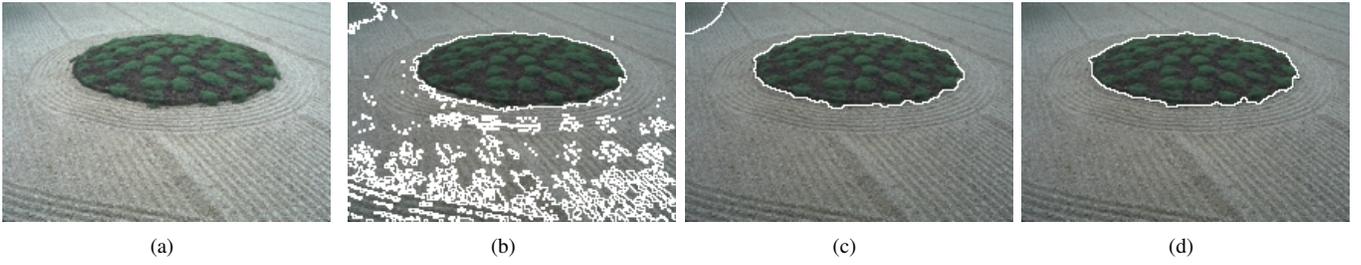


Fig. 1. a) Original Image ( $216 \times 144$ ). b) Segmentation using the *Average Dissimilarity* Algorithm without any regularization. c) Segmentation using the *Average Dissimilarity* Algorithm with regularization using curvature ( $\mu = 4000$ ); CPU time for the curve evolution is 8.49 s (783 iterations). d) Segmentation using the *Average Dissimilarity* Algorithm with regularization using oversegmentation; CPU time for the oversegmentation 0.28 s, CPU time for the curve evolution 0.12 s (10 iterations).

Lemma:

The curve evolution corresponding to the steepest descent minimization of  $E_N$  (34) is:

$$\begin{aligned} \frac{\partial \phi(p_2)}{\partial t} = & -\frac{\delta(\phi(p_2)) \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) dp_1}{diss(A, \Omega)} \\ & + \frac{\delta(\phi(p_2)) \iint_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) dp_1}{diss(B, \Omega)} \\ & + \left( \frac{diss(A, A)}{(diss(A, \Omega))^2} - \frac{diss(B, B)}{(diss(B, \Omega))^2} \right) \\ & \cdot \delta(\phi(p_2)) \iint_{\Omega} w(p_1, p_2) dp_1 \end{aligned} \quad (35)$$

*Proof:*

As done for  $E_A$  in order to minimize the cost function in (34) we need to find the first variation of elements such as:

$$\varepsilon = \frac{\int f(\phi(x)) dx}{\int g(\phi(y)) dy} \quad (36)$$

We previously showed that this can be obtained using (31).  $E_N$  can be written as:

$$E_N = \frac{\int f_1(\phi(x)) dx}{\int g_1(\phi(y)) dy} + \frac{\int f_2(\phi(x)) dx}{\int g_2(\phi(y)) dy}$$

In this case we have:

$$\begin{aligned} \frac{\partial f_1}{\partial \phi} &= \delta(p_2) \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) dp_1 \\ \frac{\partial f_2}{\partial \phi} &= -\delta(p_2) \iint_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) dp_1 \\ \frac{\partial g_1}{\partial \phi} &= \delta(p_2) \iint_{\Omega} w(p_1, p_2) dp_1 \\ \frac{\partial g_2}{\partial \phi} &= -\delta(p_2) \iint_{\Omega} w(p_1, p_2) dp_1 \end{aligned}$$

Combining (31) with the four terms above, we obtain the following curve evolution equation:

$$\begin{aligned} -\frac{\partial \phi}{\partial t} = & \frac{\frac{\partial f_1}{\partial \phi} diss(A, \Omega) - \frac{\partial g_1}{\partial \phi} diss(A, A)}{(diss(A, \Omega))^2} \\ & + \frac{\frac{\partial f_2}{\partial \phi} diss(B, \Omega) - \frac{\partial g_2}{\partial \phi} diss(B, B)}{(diss(B, \Omega))^2} \end{aligned} \quad (37)$$

Rearranging the terms and after some algebra we get (35), which concludes the proof. ■

Generalizing the model to the case of  $n$  regions yields the following cost function:

$$\begin{aligned} E_N &= \sum_{i=1}^n \frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_i(p_1) \chi_i(p_2) dp_1 dp_2}{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_i(p_2) dp_1 dp_2} \\ &= \sum_{i=1}^n \frac{diss(A_i, A_i)}{diss(A_i, \Omega)} \end{aligned} \quad (38)$$

Using the same notation as in Section IV we can write the expression of the curve evolution for the surface  $\phi_i$ , which minimizes (38), as:

$$\begin{aligned} \frac{\partial \phi_i(p_2)}{\partial t} = & \delta(\phi_i(p_2)) \left[ \sum_{j=1}^{n/2} \chi_j^i(p_2) \right. \\ & \left( -\frac{\iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) dp_1}{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) dp_1 dp_2} \right. \\ & + \frac{\iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) dp_1}{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) dp_1 dp_2} \\ & \left. \left. + (\Gamma_1 - \Gamma_2) \iint_{\Omega} w(p_1, p_2) dp_1 \right) \right] \end{aligned} \quad (39)$$

where

$$\Gamma_1 = \frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) \chi_j^i(p_2) H(\phi_i(p_2)) dp_1 dp_2}{\left( \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) dp_1 dp_2 \right)^2}$$

and

$$\Gamma_2 = \frac{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) \chi_j^i(p_2) (1 - H(\phi_i(p_2))) dp_1 dp_2}{\left( \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) dp_1 dp_2 \right)^2}$$

## VI. REGULARIZATION

In the case of noisy images, our proposed method (similarly many other curve evolution techniques) suffers from the fact that the curves tend to split in many small portions wrapping around several few-pixel-sized regions (see Fig. 1). This demonstrates a need for regularizing techniques. In this section, we first talk about the regularization through motion by mean curvature, which, despite being extensively utilized in the literature, has some drawbacks. To get around these problems we then introduce a novel regularization method based on oversegmentation, which shows higher robustness to noise and allows a fast implementation of the curve evolution.

### A. Regularization using Motion by Mean Curvature

The cost function of the curve evolution framework can include a regularizing term proportional to the length of the curve. The length of the zero level set contour (i.e. the curve) is:

$$L = \int_{\Omega} |\nabla H(\phi(x, y))| dx dy = \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy \quad (40)$$

A term proportional to  $L$  can be added to the cost function and weighted by a constant term  $\mu$ . By minimizing this additional term using the steepest descent and using  $t$  as a descent variable we can obtain:

$$\frac{\partial \phi}{\partial t} = \mu \delta(\phi) \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) = \mu \delta(\phi) \kappa \quad (41)$$

where  $\kappa$  is the curvature of the level set function  $\phi$ . This constraint introduces a new term controlling the elasticity of the curve, preventing it from splitting in many small parts.

This approach, extensively utilized in the literature (see for example [6], [9], [10]), has some drawbacks. First drawback is the increased computational burden. This is due to the fact that we need to solve a nonlinear parabolic partial differential equation (PDE) on large domains with stringent time steps restrictions. The motion by mean curvature is responsible for the parabolic nature of the PDE. Secondly, the nature of this new term is uncorrelated with the image characteristics. Thirdly, tuning of the parameter  $\mu$  is critical in order to obtain good segmentation results and its choice is image dependent.

### B. Regularization using Oversegmentation

As an alternative way to regularizing the curve evolution we propose a novel approach based on oversegmentation. The robustness to noise is achieved by using an adaptive partitioning of the image into superpixels (in our experiments the number of superpixels vary between 800 and 1500), which reduces the sensitivity to isolated noisy pixels. Dissimilarities are calculated at superpixel level using average feature values of the superpixels. The motivation for this is that pixels are not natural entities but purely a consequence of a discretization process. On the other hand superpixels provide a locally adaptive, coherent and concise representation, which maintains most of the information necessary for segmentation [29].

Oversegmentation is performed in three steps using a watershed-like region growing algorithm. First Canny edge detector is applied to get the edges. Then  $N$  points are randomly sampled as the seed points. We make sure no edge points are sampled (if an edge point is sampled, we resample). Then using Fast Marching [30], every pixel of the image is assigned to the closest seed point in the geodesic distance sense. The cost of traveling towards the seed is defined by a nonlinear scaling of the edge strengths. Let  $F(x, y)$  be the edge strength function. Calculating distances to the seeds boils down to solving the following equation for  $T$  (the distance function):

$$|\nabla T(x, y)| = F(x, y) \quad (42)$$

Each point is assigned to the closest seed (Fig. 2 shows an example of the oversegmentation algorithm output).



Fig. 2. (a) An image of a bear. (b) Oversegmentation in 800 superpixels, using the proposed watershed-like region growing algorithm.

The first beneficial consequence of the oversegmentation approach is in terms of memory/time requirement to store/access the dissimilarity matrix. In fact, storing the full matrix requires storing  $m^2 n^2$  values, where  $m$  and  $n$  are width and height of the image. A solution proposed in [1] relies on a dimensionality reduction of the similarity matrix along one dimension.

Our proposed oversegmentation approach allows adaptive partitioning of the image, which reduces the dimensionality of the similarity matrix in both dimensions. Consider for example the oversegmentation of the image into  $k$  superpixels. Now we can afford computing and storing only  $k \times k$  similarities, reducing both the memory requirements and the computation load of calculating the dissimilarity-based evolution forces (integrals in equations (26), (33) or (39)). In addition, the absence of the curvature term yields a fast implementation of the curve evolution, which is described in the next section.

## VII. FAST IMPLEMENTATION

The bottleneck in terms of computational requirements of the proposed algorithms is in the calculation of the integrals of equations (26), (33) or (39), which has to be repeated every iteration. These integrals can be significantly simplified if we assume that only few pixels change phase from one iteration to the next one. If this holds true, in fact, the normalizing quantities  $Area_i$  in (32) and  $diss(A_i, \Omega)$  in (38) can be considered approximately constant. Therefore the minimization of (32) and (38) can be performed similarly to the minimization of (23), leading to the following two curve evolution equations (simplified versions of (33) and (39) respectively):

$$\frac{\partial \phi_i(p_2)}{\partial t} = \delta(\phi_i(p_2)) \left[ \sum_{j=1}^{n/2} \chi_j^i(p_2) \left( - \frac{\iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) dp_1}{\iint_{\Omega} \chi_j^i(p) H(\phi_i(p)) dp} + \frac{\iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) dp_1}{\iint_{\Omega} \chi_j^i(p) (1 - H(\phi_i(p))) dp} \right) \right] \quad (43)$$

and

$$\frac{\partial \phi_i(p_2)}{\partial t} = \delta(\phi_i(p_2)) \left[ \sum_{j=1}^{n/2} \chi_j^i(p_2) \left( - \frac{\iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) dp_1}{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) H(\phi_i(p_1)) dp_1 dp_2} + \frac{\iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) dp_1}{\iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \chi_j^i(p_1) (1 - H(\phi_i(p_1))) dp_1 dp_2} \right) \right] \quad (44)$$

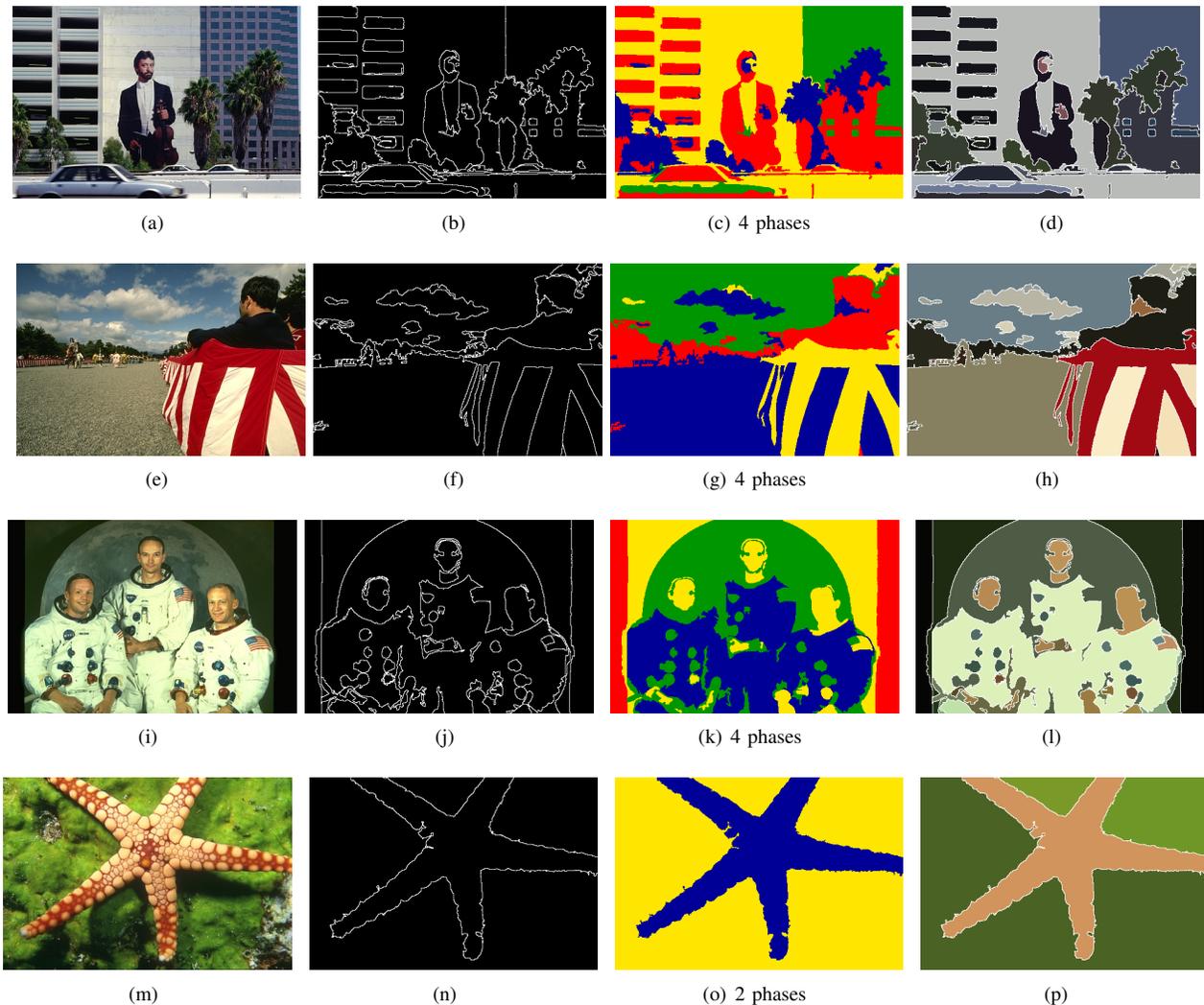


Fig. 3. Segmentation results using the fast implementation of the *Average Dissimilarity* algorithm. First column: original images ( $144 \times 216$ ). Second column: edge maps. Third column: 4 (or 2 for the last row) phases, each represented by different colors. Fourth column: segments represented by their average colors. (a-d) Cpu time for the curve evolution 0.02 secs, converged in 3 iterations. (e-h) Cpu time for the curve evolution 0.03 secs, converged in 8 iterations. (i-l) Cpu time for the curve evolution 0.04 secs, converged in 4 iterations. (m-p) Cpu time for the curve evolution 0.58 secs, converged in 12 iterations. See Table I for the comparison with the standard implementation of the regularization using curvature (Images are best viewed in color).

Later in this section we demonstrate a sequential implementation of the evolution that allows the use of these simplified expressions.

There are essentially two ways to alleviate the computational burden: the first one is to speed up the integral calculation and the second one is to reduce the number of iterations. We have addressed the first point using oversegmentation to approximate these integrals. In this section we demonstrate a method to achieve a fast converging evolution, reducing therefore the number of iterations.

Since we do not need to include a curvature term, we can convert the PDE into an ODE relaxing the stringent condition on the time step. Moreover, following the approach presented in [31], we extend the support of the evolving points from the support of the delta function to the whole image domain. This is done by replacing the delta function by 1 in equations (22), (29), (35). We obtain the following new evolution equation

(for the 2-region *Total Dissimilarity* case):

$$\begin{aligned} \frac{\partial \phi(p_2)}{\partial t} = & - \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) dp_1 \\ & + \iint_{\Omega} w(p_1, p_2) (1 - H(\phi(p_1))) dp_1 \end{aligned} \quad (45)$$

We have therefore extended the evolution to all the points in the image domain, increasing the robustness of the evolution process to avoid local minima. We also utilize a fast technique to carry on the evolution. Consider now the ODE in (45), which can be rewritten as:

$$\frac{\partial \phi(p)}{\partial t} = V(p), \quad (46)$$

This means, if  $V$  is positive then  $\phi$  increases and vice versa if  $V$  is negative ( $p$  is, as usual, a 2D point on the image domain). Since we are interested only in the discontinuities of  $\phi$ , in particular at the points where  $\phi$  changes sign, we can

also ignore the ODE and set

$$\phi(p) = \begin{cases} 1 & \text{if } V(p) \geq 0 \\ -1 & \text{if } V(p) < 0 \end{cases}$$

The velocity  $V$  is given through the computation of the integrals in (45). We can interpret this procedure as assigning the points to a region (-1 or +1) that they are more similar to. We can therefore establish a connection between this procedure and the nearest neighbor rule of the sequential k-Means algorithm: each point is assigned according to a particular metric (*Total Dissimilarity*, *Average Dissimilarity*, *Normalized Dissimilarity*) to the closest cluster. To complete the analogy we can note that the centroid rule corresponds to recalculating the velocities  $V$  according to the updated partitions. In addition, by utilizing the sequential k-Means algorithm instead of the batch version and updating the partitions each time we update a pixel value, we make sure that at most one pixel changes phase at each iteration. This makes it possible to use the simplified equations in (43) and (44) since  $Area_i$  and  $diss(A_i, \Omega)$  can be approximated as constants. It can be easily shown that using this fast implementation procedure we still perform a steepest descent minimization on the original cost functions.

The experimental results show that this fast implementation technique (regularized using oversegmentation) reduces the computational time by more than an order of magnitude with respect to the standard implementation of the regularization by mean curvature. In fact, for the vast majority of the images, the convergence is reached in less than ten iterations. In Table I we present the CPU running times for the segmentation of the images shown in Figure 3. All the experiments are performed using an Intel Pentium 4, 3 GHz with 1 GB of RAM. Despite the small overhead due to performing the oversegmentation, the total time is drastically reduced.

TABLE I  
COMPUTATIONAL REQUIREMENTS FOR THE IMAGES IN FIG. 3.

Image	Standard Implementation	Fast Implementation
Street	Evolution 56.55 s (1895 it.) Total time 56.55 s	Oversegmentation 0.235 s
		Evolution 0.02 s (3 it.) Total time 0.255 s
Parade	Evolution 57.20 s (1910 it.) Total time 57.20 s	Oversegmentation 0.251 s
		Evolution 0.03 s (8 it.) Total time 0.281 s
Astronauts	Evolution 56.56 s (1895 it.) Total time 56.56 s	Oversegmentation 0.203 s
		Evolution 0.04 s (4 it.) Total time 0.243 s
Seastar	Evolution 3.21 s (251 it.) Total time 3.21 s	Oversegmentation 0.218 s
		Evolution 0.58 s (12 it.) Total time 0.798 s

## VIII. EXPERIMENTAL RESULTS

In this section we compare the performance of the segmentation frameworks presented in the previous sections with the the state of the art techniques in region-based variational segmentation and in graph partitioning segmentation. In particular we chose to compare our algorithm with the Chan-Vese model [6] and with the Normalized Cuts (Ncut) algorithm [2]. These algorithms are largely adopted by the research community and

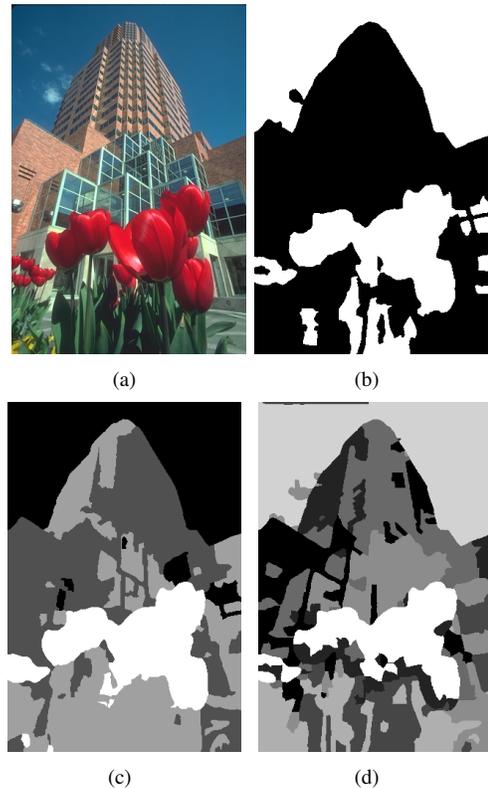


Fig. 4. (a) Original Image (b), (c), (d) Segmentation results using the *Total Dissimilarity* Algorithm, with 2, 4 and 8 phases respectively. Each phase (or region), not necessarily connected, is visualized with a different gray level. The dissimilarities are computed as distances in the CIE-Lab color space. The regularization is performed using oversegmentation.

considered state of the art in region-based active contours and graph partitioning segmentation respectively.

In Fig. 4 we provide a visual comparison of the segmentation output of the *Total Dissimilarity* Algorithm, regularized using oversegmentation, using a different number of phases (2, 4 and 8) on the same image. The dissimilarities are computed as  $L_2$  distances in the CIE-Lab color space as follows:

$$w(p_1, p_2) = \left( \sum_{i=1}^3 (c_i(p_1) - c_i(p_2))^2 \right)^{\frac{1}{2}} \quad (47)$$

where  $c_i$  are the three Lab channels. The regions are visualized with different gray levels. As we increase the number of phases, more details are captured in the segmentation but uniform regions (e.g. the sky) are still preserved and not oversegmented.

In order to have a quantitative evaluation of the segmentation results, we compared the three proposed algorithms (*Total*, *Average* and *Normalized Dissimilarity*) with the Chan-Vese model using the Berkeley Segmentation Data Set (BSDS) benchmark. The test set is composed of 100 images and for every image several human segmentations are provided. These human segmentations are considered ground truth and are used to compute *precision* ( $p$ ) and *recall* ( $r$ ) as measures of the accuracy of the segmentation. Precision is the probability that a pixel indicated as a boundary pixel by the segmentation algorithm is truly a boundary pixel. Recall is the probability that

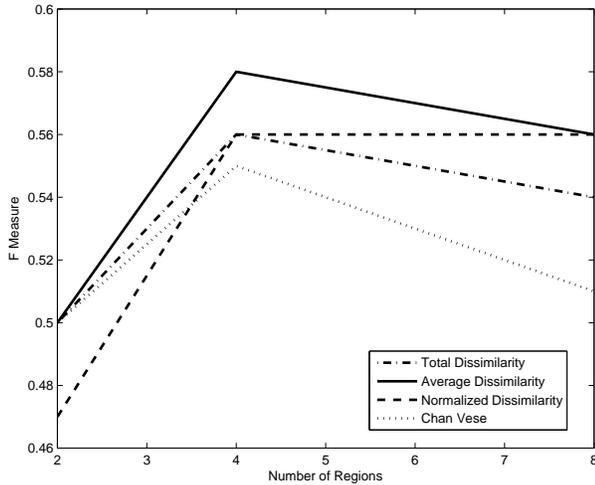


Fig. 5.  $F$  measure plots for the *Total*, *Average* and *Normalized Dissimilarity* algorithm, compared with the Chan-Vese model. All the experiments are performed using the CIE-Lab color space. In the Chan-Vese model the curve is initialized with circular sub-parts on a square grid of side 25 pixels, to avoid getting stuck in local minima. The dissimilarity-based methods are initialized sequentially assigning superpixels to the different regions.

a true boundary pixel in the ground truth is correctly detected by the algorithm. Then the  $F$  measure, the harmonic mean of precision and recall ( $F = \frac{2pr}{p+r}$ ), is presented as a measure of performance combining both precision and recall. In Fig. 5 we present the  $F$  measures scored by the four algorithms, using 2, 4 and 8 phases. All the experiments are conducted in the CIE-Lab color space. The Chan-Vese algorithm has been implemented using the semi-implicit finite difference scheme described in [5], and a parameter space search has been performed in order to best tune the  $\mu$  parameter, coefficient of the curvature related term (we refer to Appendix C for more details about this experiment). Notice that the algorithms proposed in this work do not require any parameter tuning since the regularization is performed using oversegmentation and the fast implementation described in Section VII is used to carry out the evolution.

With only two phases, the scores of different algorithms are very similar and not very insightful since the background-foreground model is not suitable for the segmentation of such complex natural images. On the other hand, we notice that with increasing number of phases (4 and 8), our dissimilarity based models outperform Chan-Vese model. We also see that normalized frameworks achieve better performance (*Average* and *Normalized Dissimilarity*). This confirms our intuition in terms of the need of normalized frameworks (See Section V).

In Fig. 6(a,b,c), we present a comparison between the performances obtained regularizing the proposed framework using over-segmentation and using motion by mean curvature (again, a search in the parameter space was performed to best tune the curvature coefficient  $\mu$ ). Notice that the regularization by over-segmentation leads to higher scores (except for the two-region case, which is not very informative). This can be motivated by the fact that motion by mean curvature smoothes curves in an isotropic fashion, while over-segmentation is locally adaptive and therefore preserves sharp edges. We also

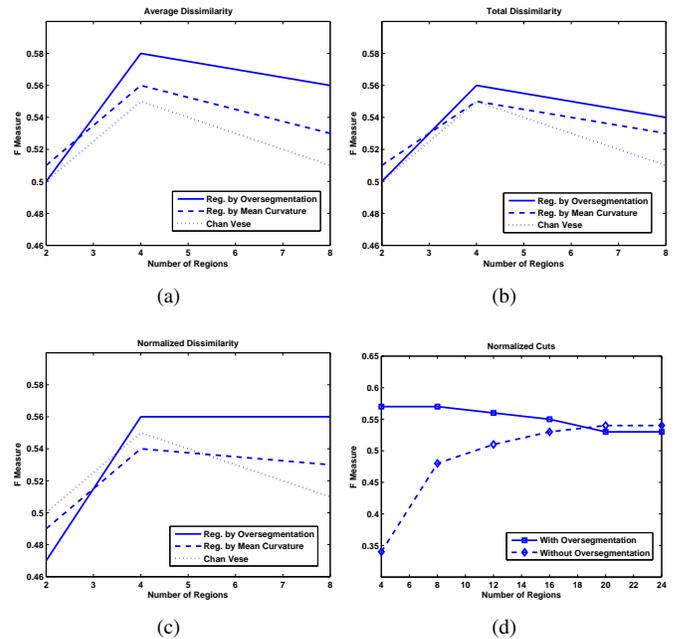


Fig. 6. (a,b,c) Comparison between regularization techniques for the three dissimilarity-based frameworks. For the regularization using motion by mean curvature, the curve is initialized with circular sub-parts on a square grid of 25 pixels. The simplified equations (43) and (44) are used to carry on the evolution (the assumption of small number of pixels changing phase at each iteration holds in most of the cases). (d) Comparison between Normalized Cuts at pixel and super-pixel level. For the pixel level case, similarities are calculated in a  $30 \times 30$  window around each pixel (memory limit of the eigensolver). For the super-pixel level case, images are over-segmented in 1500 super-pixels.

notice that the scores of the proposed framework are still generally higher with respect to the Chan-Vese model, except for the *Normalized Dissimilarity* algorithm in the case of low number of regions (2 and 4). Fig. 6(d) is meant to demonstrate the applicability of the over-segmentation to other segmentation methods based on pairwise dissimilarity, in this case the Normalized Cuts algorithm [2]. Working at the pixel level requires to enforce the sparsity of the similarity matrix and high scores are obtained only for the segmentation in a high number of regions (20-24). On the other hand, the over-segmentation in superpixels allows the reduction of the sparsity (or even the use of a full similarity matrix), leading to high  $F$ -measure even for the segmentation in a small number of regions.

We can also extend the definition of dissimilarity measure to include edge information, which is an important cue in identifying the segmentation boundaries. This demonstrates the flexibility of our algorithm. Dissimilarities are still computed as distances in the color space but we add a *geodesic* term, which represents the distance in the geodesic sense between pixels (or superpixels). This geodesic term is high if the pixels are separated by edges or low (close to zero) if the pixels belong to the same uniform region (see [27]). Fig. 7(b) and 8(b) are the output of the *Average Dissimilarity* algorithm after adding the geodesic term to the dissimilarity measure. In Fig. 7(b) each of the planet is assigned to one region, while using the Chan-Vese model (Fig. 7(c)) this

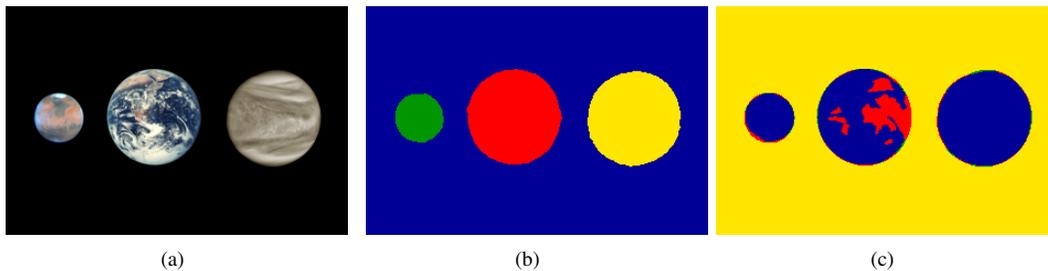


Fig. 7. (a) An image of Mars, Earth and Venus. (b) Segmentation using the *Average Dissimilarity* algorithm adding a geodesic distance term to the distance in the color space as measure of pairwise dissimilarity. Every phase corresponds to a planet (one is the background). (c) Segmentation using the Chan-Vese model. In this case the different regions are somewhat clusters in the color space and individual planets are not captured (Images are best viewed in color).

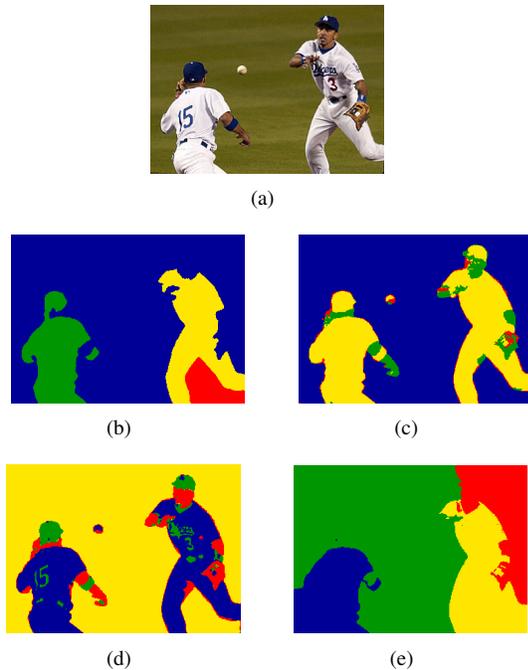


Fig. 8. (a) An image of a baseball game. (b) Segmentation results using the *Average Dissimilarity* with the geodesic distance term. (c) Segmentation results using the *Average Dissimilarity* without the geodesic distance term. (d) Segmentation results using the Chan-Vese model. (e) Segmentation output of the Normalized Cuts algorithm (Images are best viewed in color).

can not be achieved. Similarly in Fig. 8(b) each baseball player's body is assigned to one particular region, while with Chan-Vese model this can not be obtained (Fig. 8(d)). Fig. 8(e) shows the output of the Normalized Cuts algorithm on the baseball image. In this case, since the similarities have to be calculated locally for computational reasons (memory limits of the eigensolver), the different regions are likely to be composed of connected components but some details are mis-segmented, as for example the region between the legs or the region on the bottom left of the image. Fig. 8(c) shows that, without including the geodesic term in the dissimilarity measure, we can get results similar to the Chan-Vese case (Fig. 8(d)), demonstrating the increased flexibility of our framework.

Fig. 9 provides a visual comparison between *Normalized Dissimilarity* (4 phases) and Normalized Cuts (Ncut) algorithm (asking for 10 segments). We compare these two methods

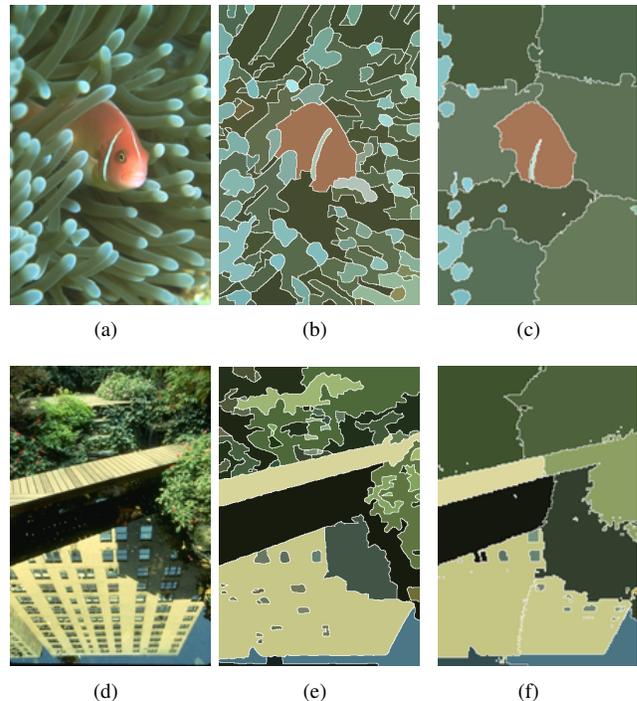


Fig. 9. (a), (d) Original images. (b), (e) Output of the *Normalized Dissimilarity* algorithm. (c), (f) Output of the Normalized Cuts algorithm (10 segments). All the experiments have been performed in the CIE-Lab color space. For the normalized cuts the similarities are calculated in a  $30 \times 30$  neighborhood around each pixel, sampled with probability 0.5 (maximum allowed by the memory limits of the eigensolver) (Images are best viewed in color).

since they are both minimizing similar cost functions (see [2]). All the experiments are conducted in the CIE-Lab color space. Since the similarity calculations in Normalized Cuts are restricted to local neighborhoods, Ncut algorithm tends to oversegment uniform regions. On the other hand the output of the *Normalized Dissimilarity* algorithm maintains coherence within big regions and preserves small scale details at the same time.

## IX. DISCUSSION AND CONCLUSION

### A. Towards a Self Tuning Segmentation, Estimation of the Number of Image Modes

One point that needs to be addressed is the estimation of the number of phases given an image. This problem can be addressed by analyzing the image feature space with the goal

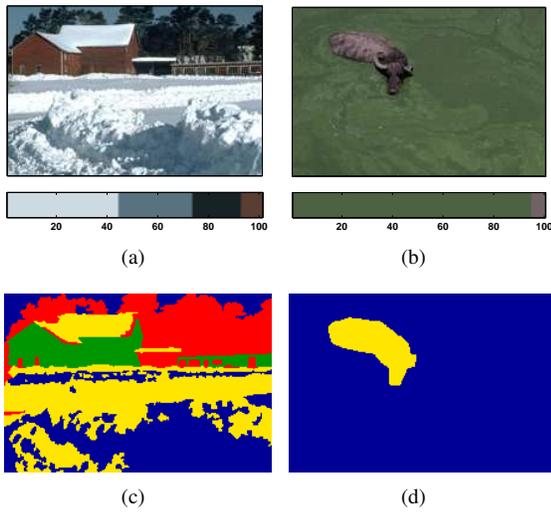


Fig. 10. (a) and (b) Image signatures obtained using the Mean Shift clustering algorithm. We pictorially represent the signature as the cluster representative colors with width proportional to the percentage of pixels belonging to it. (c) and (d) Segmentations using the *Average Dissimilarity* algorithm with number of phases according to the signature (4 and 2 respectively).

of obtaining an estimate of the number of clusters present in this space.

Many clustering methods are not suitable for this kind of analysis since they require the knowledge of the number of clusters (See [32]). On the other hand, nonparametric clustering methods do not require any preliminary assumption and can be used to analyze a complex data distribution, as in our case, feature space of natural images.

We compute image signatures using nonparametric clustering based on density estimation. By thinking of the feature points as samples drawn from a probability density function of the feature parameters we can search for the modes of the distribution, i.e. the local maxima and identify these modes with the cluster center candidates. The Parzen Windows technique (see for example [33], section 2.5.6), also known as kernel density estimation, is suitable for this purpose. Given  $N$  data points  $\mathbf{x}_i$ ,  $i = 1, \dots, N$  in an  $l$ -dimensional space, the estimate of the density function is given by:

$$\hat{p}(\mathbf{x}) = \frac{1}{Nh^l} \sum_{i=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (48)$$

where  $K(\mathbf{x})$  is the kernel function and  $h$  is the bandwidth parameter (in our case the pixels of the image are in the CIELab space, where  $l = 3$ ). Searching along the gradient direction of this density function to identify the peaks can be efficiently performed using the Mean Shift procedure introduced by Fukunaga and Hostetler in [34]. It can be shown that if the kernel function is chosen to be of the form (known as the Epanechnikov kernel)

$$K(\mathbf{x}) = \begin{cases} c(1 - \|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| \leq 1 \\ 0 & \text{if } \|\mathbf{x}\| > 1 \end{cases}$$

where  $c$  is a normalization factor necessary for the kernel function to integrate to one. The gradient of the estimated density is proportional to the sample mean shift. Starting

from a random point in the feature space and performing the simple iterative algorithm described in [35] for example we can efficiently identify the modes of the distribution. All the points belonging to the basin of one particular mode (in other words converging to that mode) will be clustered together. The only parameter to tune is the bandwidth  $h$ , which is a characteristic of the feature space we work on. The issue of the bandwidth (scale) selection is therefore task dependent. For CIELab space we choose  $h = 10$ . Fig. 10 shows that the number of phases indicated by the mean shift procedure can be effectively utilized in segmentation. See Fig. 10 for two examples for which 4 and 2 phases came out to be optimal respectively.

## B. Conclusion

In this paper we presented variational cost functions based on pairwise pixel dissimilarities and we derived the minimization of such cost functions within a multiphase level sets framework in order to achieve a multi-region image segmentation. One of the advantages of this approach is the flexibility granted by the possibility of embedding into the dissimilarity measure information coming from heterogeneous feature spaces or different image cues. All the models introduced in this paper are not restricted to background/foreground segmentation and independent of the choice of the dissimilarity measure.

In the experimental section we validated the proposed method by comparing its performance with the state of the art techniques in region based and graph based segmentation and showing improvements in the performance. We also introduced a fast implementation technique, which significantly reduces the computational time of the curve evolution and avoids the need of hand-tuning a curvature parameter, critical in most of the cases to obtain a good segmentation result.

## APPENDIX A

Minimizing  $E = \int f(\phi(x, t))dx$  with the steepest descent method, using  $t$  as the descent variable leads to:

$$\phi_t(x) = -f_\phi(\phi(x)) \quad (49)$$

*Proof:*

We need to impose that the first variation of  $E$  with respect to  $t$  is negative

$$\frac{\partial E}{\partial t} = \frac{\partial}{\partial t} \int f(\phi(x, t))dx \leq 0 \quad (50)$$

We can expand  $E$  as follows:

$$E = \int f(\phi(x, t))dx = \int f(\phi(x, t_0) + (t - t_0)\phi_t(x, t_0))dx$$

and again

$$E = \int f(\phi(x, t_0)) + f_\phi(\phi(x, t_0))(t - t_0)\phi_t(x, t_0)dx \quad (51)$$

Now taking the derivative w.r.t.  $t$  and interchanging the order of derivative and integral (which is allowed by the fact that the integration domain does not depend on  $t$ ) we get

$$\frac{\partial E}{\partial t} = \int f_\phi(\phi(x, t_0))\phi_t(x, t_0)dx \quad (52)$$

Therefore the steepest descent minimization is obtained by

$$\phi_t(x, t_0) = -f_\phi(\phi(x, t_0)) \quad (53)$$

which concludes the proof. ■

#### APPENDIX B

Minimizing

$$E = \frac{\int f(\phi(x, t))dx}{\int g(\phi(y, t))dy} = \frac{\Lambda}{\Theta} \quad (54)$$

with the steepest descent method, using  $t$  as the descent variable leads to:

$$\phi_t(x) = -\frac{f_\phi(\phi(x))\Theta - g_\phi(\phi(x))\Lambda}{\Theta^2} \quad (55)$$

*Proof:*

Taking the derivative of  $E$  w.r.t. to  $t$  leads to

$$\frac{\partial E}{\partial t} = \frac{(\frac{\partial}{\partial t} \int f(\phi(x, t))dx)\Theta - (\frac{\partial}{\partial t} \int g(\phi(y, t))dy)\Lambda}{\Theta^2} \quad (56)$$

Now expanding  $f(\phi(x, t))$  and  $g(\phi(y, t))$  along the lines of Appendix A and interchanging the order of derivation and integration yields to:

$$\frac{\partial E}{\partial t} = \frac{(\int f_\phi(\phi(x))\phi_t(x)dx)\Theta - (\int g_\phi(\phi(y))\phi_t(y)dy)\Lambda}{\Theta^2} \quad (57)$$

Now after a change of variable  $y = x$  we can write:

$$\frac{\partial E}{\partial t} = \frac{\int \phi_t(x)(f_\phi(\phi(x))\Theta - g_\phi(\phi(x))\Lambda)dx}{\Theta^2} \quad (58)$$

Therefore we have the steepest descent minimization for:

$$\phi_t(x) = -\frac{f_\phi(\phi(x))\Theta - g_\phi(\phi(x))\Lambda}{\Theta^2} \quad (59)$$

which concludes the proof. ■

#### APPENDIX C

In Fig. 11 we show the F measure plots for the Chan-Vese algorithm, changing the curvature coefficient  $\mu$ .

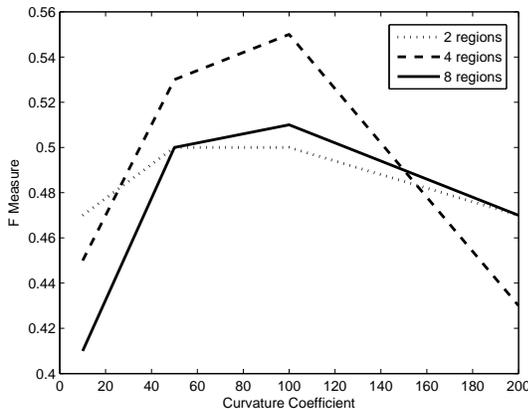


Fig. 11. F measure plots for the Chan-Vese algorithm, changing the curvature coefficient  $\mu$ . The algorithm has been implemented using the semi-implicit finite difference scheme described in [5]. The curve is initialized with circular sub-parts on a square grid of side 25 pixels, to avoid getting stuck in local minima.

#### APPENDIX D

In the 4-region case,  $k = 2$ , (23) becomes:

$$\begin{aligned} E = & \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) [\chi_1(p_1)\chi_1(p_2)] dp_1 dp_2 \quad (60) \\ & + \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) [\chi_2(p_1)\chi_2(p_2)] dp_1 dp_2 \\ & + \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) [\chi_3(p_1)\chi_3(p_2)] dp_1 dp_2 \\ & + \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) [\chi_4(p_1)\chi_4(p_2)] dp_1 dp_2 \end{aligned}$$

In this case the characteristic functions are:

$$\chi_1(p) = H(\phi_1(p))H(\phi_2(p)) \quad (61)$$

$$\chi_2(p) = H(\phi_1(p))(1 - H(\phi_2(p))) \quad (62)$$

$$\chi_3(p) = (1 - H(\phi_1(p)))H(\phi_2(p)) \quad (63)$$

$$\chi_4(p) = (1 - H(\phi_1(p)))(1 - H(\phi_2(p))) \quad (64)$$

and for example

$$\chi_1^1(p) = H(\phi_2(p)) \quad (65)$$

$$\chi_2^1(p) = (1 - H(\phi_2(p))) \quad (66)$$

Now using (17), we can find the evolution equations for the surfaces  $\phi_1$  and  $\phi_2$ .

Reorganizing the terms we can interpret these evolution forces (we present here the evolution equation for  $\phi_1$ , while the one for  $\phi_2$  is derived in a similar fashion):

$$\begin{aligned} \frac{\partial \phi_1(p_2)}{\partial t} = & \quad (67) \\ & - [H(\phi_2(p_2))\delta(\phi_1(p_2))] \iint_{\Omega} w(p_1, p_2)\chi_1(p_1)dp_1 \\ & - [(1 - H(\phi_2(p_2)))\delta(\phi_1(p_2))] \iint_{\Omega} w(p_1, p_2)\chi_2(p_1)dp_1 \\ & + [H(\phi_2(p_2))\delta(\phi_1(p_2))] \iint_{\Omega} w(p_1, p_2)\chi_3(p_1)dp_1 \\ & + [(1 - H(\phi_2(p_2)))\delta(\phi_1(p_2))] \iint_{\Omega} w(p_1, p_2)\chi_4(p_1)dp_1 \end{aligned}$$

The first and third term are acting on the part of  $C_1$  that is inside of  $C_2$ , expanding it or restricting it according to the dissimilarity of the points on the curve to the two neighboring regions. The second and the fourth term are acting on the part of  $C_1$  outside of  $C_2$ , according to the same logic. In a similar fashion we can obtain the evolution equation for  $\phi_2$ .

#### ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their constructive comments. This research was supported by NSF ITR grant #0331697. The research of F. Gibou was supported in part by the Alfred P. Sloan Foundation through a research fellowship in Mathematics.

## REFERENCES

- [1] B. Sumengen and B.S. Manjunath, "Graph partitioning active contours (GPAC) for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 509–521, April 2006.
- [2] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 888–905, August 2000.
- [3] P. Perona and W. T. Freeman, "A factorization approach to grouping," in *European Conference on Computer Vision (ECCV)*, 1998, pp. 655–670.
- [4] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
- [5] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, pp. 271–293, December 2002.
- [6] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–77, February 2001.
- [7] S. C. Zhu, T. S. Lee, and A. Yuille, "Region competition: unifying snakes, region growing, and bayes/MDL for multiband image segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 1995, pp. 416–23.
- [8] S. C. Zhu and A. Yuille, "Region competition: unifying snakes, region growing, and bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 884–900, September 1996.
- [9] A. Jr. Yezzi, A. Tsai, and A. Willsky, "A statistical approach to snakes for bimodal and trimodal imagery," in *International Conference on Computer Vision (ICCV)*, 1999, pp. 898–903.
- [10] A. Tsai, A. Jr. Yezzi, and A. S. Willsky, "Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Transactions on Image Processing*, pp. 1169–86, August 2001.
- [11] A. Chakraborty and J. Duncan, "Game-theoretic integration for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 1, pp. 12–30, January 1999.
- [12] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–80, February 1997.
- [13] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, pp. 321–31, 1988.
- [14] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1985, pp. 22–6.
- [15] T. Chan, B. Y. Sandberg, and L. Vese, "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, pp. 130–141, 2000.
- [16] N. Paragios and R. Deriche, "Geodesic active regions for supervised texture segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 1999, vol. 2, pp. 926–932.
- [17] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, pp. 223–47, February 2002.
- [18] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson, "Image segmentation using active contours: Calculus of variations or shape gradients?," *SIAM Applied Mathematics*, 2003.
- [19] D. Cremers, N. Sochen, and C. Schnorr, "Multiphase dynamic labeling for variational recognition-driven image segmentation," in *European Conference on Computer Vision (ECCV)*, Springer, Ed., 2004, pp. 74–86.
- [20] D. Cremers, "Dynamical statistical shape priors for level set based tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1262–1273, August 2006.
- [21] D. Cremers, S. J. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 335–351, 2006.
- [22] T. Riklin Raviv, N. Kiryati, and N. Sochen., "Prior-based segmentation by projective registration and level sets," in *IEEE International Conf. on Computer Vision (ICCV)*, October 2005, pp. pp 204–211.
- [23] T. Riklin Raviv, N. Kiryati, and N. Sochen, "Segmentation by level sets and symmetry," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, New York, June 2006.
- [24] T. Riklin Raviv, N. Kiryati, and N. Sochen, "Prior-based segmentation and shape registration in the presence of projective distortion," *International Journal of Computer Vision (IJCV)*, Accepted for publication, 2006.
- [25] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 837–42, August 1996.
- [26] J. Malik, S. Belongie, J. Shi, and T. Leung, "Textons, contours and regions: cue integration in image segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, September 1999, pp. 918–925.
- [27] B. Sumengen, L. Bertelli, and B. S. Manjunath, "Fast and adaptive pairwise similarities for graph cuts-based image segmentation," in *Proc. IEEE CVPR Workshop on Perceptual Organization in Computer Vision (POCV)*, June 2006.
- [28] H. K. Zhao, T. Chan, B. Merrimar, and S. Osher, "A variational level set approach to multiphase motion," *Journal of Computational Physics*, vol. 127, pp. 179–195, 1996.
- [29] X. Ren and J. Malik, "Learning a classification model for segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, vol. 1, pp. 10–17.
- [30] J. A. Sethian, "A marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci.*, vol. 93, no. 4, 1996.
- [31] 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*.
- [32] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical Pattern Recognition: a Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, January 2000.
- [33] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Elsevier Academic Press, 2003.
- [34] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with application in pattern recognition," *IEEE Transactions of Information Theory*, vol. 21, no. 1, pp. 32–40, January 1975.
- [35] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.



**Luca Bertelli** received the D.Ing. degree summa cum laude in Electronic Engineering from the University of Modena, Italy in 2003 and the M.S. degree in Electrical and Computer Engineering from the University of California, Santa Barbara in 2005, where he is currently pursuing the Ph.D. degree in the field of Computer Vision and Image Processing. His research interests include variational methods for image segmentation, level set methods, shape based segmentation and other aspects of computer vision and machine learning.



**Baris Sumengen** received the BS degree from Bogazici University, Turkey, in both electrical engineering and mathematics in 1998. He received the MS degree in 2000 and the PhD degree in 2004 in electrical and computer engineering from University of California at Santa Barbara. From 2004 to 2005, he has been a postdoc at the UC Santa Barbara Center for bio-image informatics. Since 2006, he works at Like.com (formerly Riya) as a research architect leading the design of scalable content based image similarity engines. His primary research in-

terests include image segmentation, content based image retrieval, and data mining in large multimedia databases. He is a member of the IEEE Computer Society.



**B. S. Manjunath** received the BE degree in electronics (with distinction) from the Bangalore University in 1985, the ME degree (with distinction) in systems science and automation from the Indian Institute of Science in 1987, and the PhD degree in electrical engineering from the University of Southern California in 1991. He is now a professor of electrical computer engineering and director of the Center for Bio-Image Informatics at the University of California, Santa Barbara. Dr. Manjunath was a recipient of the national merit scholarship (1978-1985) and was awarded the university gold medal for the best graduating student in electronics engineering in 1985 from the Bangalore University. His current research interests include image processing, data hiding, multimedia databases, and bio-image informatics. He is a coeditor of the book on Introduction to MPEG-7 (Wiley, 2002). He was an associate editor of the IEEE Transactions Image Processing and is currently an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence, and the IEEE Transactions on Multimedia. He is a fellow of the IEEE.



**Frédéric Gibou** received his D.E.A in applied mathematics from the University of Bordeaux, France, in 1994. He then obtained his Masters in applied mathematics at UCLA, Los Angeles, California in 1998 and a PhD in applied mathematics at UCLA, Los Angeles, California in 2001. He was awarded a NSF postdoctoral fellowship to pursue his research at Stanford University (2001-2004) in the Department of Computer Science and in the Department of Mathematics.

Prof. Gibou is an Assistant Professor in the Department of Computer Science and in the Department of Mechanical Engineering at the University of California Santa Barbara. He was awarded the Alfred P. Sloan Fellowship in Mathematics in 2006. His research interests include level set methods for computing moving fronts involving topological changes and the development of new methods for approximating nonlinear partial differential equations.