A Fast Hybrid k-Means Level Set Algorithm For Segmentation

Frédéric Gibou

Computer Science Department &
Mathematics Department
Stanford University, CA 94305-9020
fgibou@math.stanford.edu

Ronald Fedkiw

Computer Science Department
Stanford University, CA 94305-9020
fedkiw@cs.stanford.edu

Abstract

In this paper, we first draw a connection between a level set algorithm and k-Means plus nonlinear diffusion preprocessing. Then, we exploit this link to develop a new hybrid numerical technique for segmentation that draws on the speed and simplicity of k-Means procedures, and the robustness of level set algorithms. The proposed method retains spatial coherence on initial data characteristic of curve evolution techniques, as well as the balance between a pixel/voxel's proximity to the curve and its intention to cross over the curve from the underlying energy. However, it is orders of magnitude faster than standard curve evolutions. Moreover, it does not suffer from the limitations of k-Means due to inaccurate local minima and allows for segmentation results ranging from k-Means clustering type partitioning to level set partitions.

1. Introduction

Segmentation is the art of automatically separating an image into different regions in a fashion that mimics the human visual system. It is therefore a broad term that is highly dependent on the application at hand, e.g. one might want to segment each object individually, groups of objects, parts of objects, etc.. In order to segment a particular image, one must first identify the intended result before a set of rules can be chosen to target this goal. The human eye uses low-level information such as the presence of boundaries, regions of different intensity or colors, brightness and texture, etc., but also mid-level and high-level cognitive information, for example, to identify objects or to group individual objects together. As a direct consequence, there are a wide variety of approaches to the segmentation problem, and many successful algorithms have been proposed and developed to simulate a number of these different processes.

Canny [3] introduced the most widely used edge detection method, which was based on local gradients. Techniques such as seeded region growing [2] or split-and-merge [28], using Markov random field modeling, were the first region-based methods. A recent interesting region-growing approach using fast marching type algorithms was proposed by Sifakis *et al.* [36], and a good review on region-growing based algorithms can be found in [47]. Among the most widely used segmentation techniques are that of Comaniciu and Meer [9] based on the mean shift, and that of Shi and Malik [35] where the grouping problem is viewed as a graph partitioning and the normalized cut criteria is proposed for segmenting the graph.

Our approach is loosely motivated by the class of methods known as deformable models. These models are based on functional minimization and were introduced to the computer vision [41] and computer graphics [40, 39] communities in the late 1980's by Terzopoulos *et al.*. An example that has gained popularity is known as "snakes" [15]. The main drawbacks of the original snakes are their sensitivity to initial conditions and the difficulties associated with topological transformations. Cohen [8] proposed a balloon force model that uses an internal inflation term to partially alleviate the sensitivity to initial conditions. Fua and Leclerc [13] introduced the first geometric model, and Caselles *et al.* [4] the first level set formulation in a non-variational setting (see also Malladi *et al.* [20]). A major advantage of the level set approach [25] is the ability to handle complex topological changes automatically. Caselles *et al.* [5] and Kichenassamy *et al.* [16] proposed the geodesic active contour, which was the first both geometric and variational model in level set form. Many models extend geodesic active contours by incorporating shape information or probabilistic estimates (e.g. see [27], [17] and the references therein). Prior literature on deformable models is vast and the interested reader is referred to the book by Sapiro [32], the survey by McInerney and Terzopoulos [21], and the references therein.

Based on the Mumford-Shah minimal partition functional [23], Chan and Vese [6, 44] (see also Yezzi [46]) proposed a new level set model for active contours to detect objects whose boundaries are not necessarily defined by a gradient. The main drawback of this algorithm is the computational expense inherent to solving the proposed nonlinear parabolic partial differential equation. Starting from this formulation we show that under suitable assumptions, the Chan and Vese model reduces to the k-Means algorithm with a nonlinear diffusion preprocessing step. We then exploit this link to propose a new hybrid algorithm that benefits from the simplicity and efficiency of k-Means while preserving the robustness of level set algorithms. Moreover, this algorithm offers flexibility allowing the user to control the amount of curve evolution properties as desired for the application at hand efficiently trading speed for accuracy.

2. A connection between level sets and k-Means

In this section, we show the link between the Chan and Vese level set implementation of the Mumford-Shah functional and the standard k-Means clustering type algorithm.

2.1 Background

Consider a possibly noisy or blurry two-dimensional image u_0 with image domain Ω and segmenting curve C. Mumford and Shah [23] proposed to decompose the image into piecewise-smooth approximations by minimization of the Mumford-Shah functional:

$$F(u,C) = \mu L(C) + \lambda \int_{\Omega} (u_0 - u)^2 d\Omega + \int_{\Omega \backslash C} |\nabla u|^2 d\Omega, \tag{1}$$

where $L(C)$ denotes the length of C, $\lambda > 0$ and $\mu \geq 0$ are parameters. The first term is a constraint on the curve's length and controls its smoothness while the other terms separate the image into different regions while allowing for discontinuities along the edges.

Chan and Vese [6] proposed an algorithm for decomposing the image into two regions with piecewise constant approximations (i.e. u is constant in each region). This corresponds to decomposing the image into regions with respect to their mean intensity values, and in this formulation the last term of equation (1) vanishes. The authors then formulated this functional in terms of the level set formalism of Osher and Sethian [25]. Later, [44] generalized this process to treat multiple regions.

The level set method is a general technique for evolving curves or surfaces that may undergo complex topological changes such as merging and pinching. The level set equation

$$\phi_t + V_n |\nabla \phi| = 0, \tag{2}$$

where ϕ is the level set function and V_n is the normal velocity at the interface, is used to keep track of the interface location as the set of points where $\phi = 0$. The interior and exterior are then designated by the points where $\phi \leq 0$ and $\phi > 0$ respectively. With standard finite differences, one can compute the normal $\vec{n} = \nabla\phi / |\nabla\phi|$ and the mean curvature $\kappa = \nabla \cdot \vec{n}$. Moreover, it is also straightforward to define the length of a curve or to integrate a quantity over a given region, etc.. For more details on the level set method, see e.g. [24, 34].

In the case of the piecewise constant approximation and only two separate regions, the functional (1) can be written in terms of the level set function as

$$F(\phi, c_1, c_2) = \mu \int_{\Omega} |\nabla H(\phi)| d\Omega + \lambda_1 \int_{\Omega} (u_0 - c_1)^2 H(\phi) d\Omega + \lambda_2 \int_{\Omega} (u_0 - c_2)^2 (1 - H(\phi)) d\Omega,$$

with the unknown constants c_i representing the mean intensity value of the region labelled i and defined by

$$c_1 = \frac{\int_{\Omega} u_0 H(\phi) d\Omega}{\int_{\Omega} H(\phi) d\Omega}, \quad c_2 = \frac{\int_{\Omega} u_0 (1 - H(\phi)) d\Omega}{\int_{\Omega} (1 - H(\phi)) d\Omega}, \tag{3}$$

where H is the Heaviside distribution. Minimizing this functional in the Euler-Lagrange framework leads to

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2] \tag{4}$$

with appropriate boundary conditions on $\partial\Omega$. The constants c_1 and c_2 are computed using (3) and a regularization H_ϵ of H. Likewise, $\delta_\epsilon = H_\epsilon^{'}$ (see e.g. [24] for numerical approximations for H_ϵ and δ_ϵ). In order to extend the evolution to all level sets of ϕ, an often used approach is to take $\delta_\epsilon(\phi) = |\nabla \phi|$. In this case, the normal velocity of equation (2) is of the form $V_n = a + b\kappa$, where a and b are constants.

This algorithm is computationally expensive because it requires the solution of a nonlinear parabolic partial differential equation (PDE) on a large time domain (i.e., from the initial curve location until the steady state is reached). Motion by mean curvature is responsible for the parabolic nature of the PDE and imposes a stringent time step restriction of the form $\triangle t = O(\triangle x^2)$, where $\triangle x$ is inversely proportional to the number of pixels/voxels in a cross section of the image. This regularizing term comes from the length constraint and provides a way to control the “elasticity” of the evolving front allowing for the processing of noisy images, and providing a length scale for grouping objects.

The nonlinearity of the parabolic PDE makes it difficult to solve with implicit time integration, although we note that the semi-implicit methods in [12, 37] could be used to partially alleviate the computational burden. However, one would still have to evolve the curve from its initial state to its final state with a $\triangle t = O(\triangle x)$ time step restriction due to the hyperbolic terms in equation (4). Moreover, [37] demonstrates that solving implicitly for nonlinear smoothing operators often leads to inaccuracy when topological changes occur.

Another relevant attempt to develop a fast implicit scheme is that of Goldenberg *et al.* [14] who considered a geodesic active contour with an edge detection criterion and regularization by area minimization. The corresponding level set formulation is

$$\frac{d\phi}{dt} = \left(\alpha g(u_0) + \nabla \cdot \left(g(u_0)\frac{\nabla\phi}{|\nabla\phi|}\right)\right)|\nabla\phi|, \qquad (5)$$

where g is an edge detection function. They proposed an efficient numerical approach based on the narrow band level set of Adalsteinsson and Sethian [1] and the additive operator splitting (AOS) of Weickert *et al.*. Their scheme is *exactly* equivalent to the following two steps: First, solve $\phi_t = \alpha g(u_0)$ with one explicit Euler time step; second, the linear inhomogeneous diffusion equation $\phi_t = \nabla \cdot (g(u_0)\nabla\phi)$ is evolved implicitly for one time step with an AOS model. However, the regularization term is reduced to a variable coefficient diffusion equation under the assumption that ϕ is a signed distance function at the end (moreover throughout) the implicit time step. This assumption is not valid. Even if the level set is initially a signed distance function, it will not be a signed distance function at time t^{n+1}, i.e. it is erroneous to assume $|\nabla\phi^{n+1}| = 1$ in order to simplify $\nabla\cdot\big(g(u_0)\nabla\phi^{n+1}/|\nabla\phi^{n+1}|\big)|\nabla\phi^{n+1}|$ to $\nabla\cdot(g(u_0)\nabla\phi^{n+1})$. If one starts with a signed distance function, motion by mean curvature can be emulated by a simple diffusion step followed by a re-initialization of ϕ in an *explicit* time integration algorithm since the discretizations are identical, but this process can not be extrapolated to *implicit* time integration. This is well addressed in [24]. Consequently, the scheme proposed in [14] does not solve for the original curvature-like regularization leaving out the correction terms from [12, 37].

The method in [14] seeks to allow for large time steps and therefore ignores the time step restriction of the forward Euler step. This introduces both numerical errors and instability. These instabilities may be dampened if enough diffusion is introduced during the smoothing operation of step two, however if the degree of smoothing is too small instabilities *will* develop. In spite of these comments, we agree that a diffusion-like operation can be used in place of motion by mean curvature to regularize the level set evolution. In fact in section 2.3 we propose a similar preprocessing step on the image in order to introduce the notion of length scale. On the other hand, it is also important to have the option of using mean curvature regularization, and we propose this option as well in section 3.

2.2 Connection to k-Means

For the sake of clarity, we present the connection between the Chan and Vese algorithm and k-Means in the case of two regions, but we note that it extends trivially to multiple regions. We begin by considering the segmentation of individual objects from a clean (i.e. no noise) but possibly blurred image. In this simple case the notion of a length scale is not important, and we can ignore the computationally costly parabolic term (the first term) in equation (4). The resulting equation is hyperbolic in nature when $\delta_\epsilon(\phi)$ is approximated by $|\nabla\phi|$. This rather awkward approximation of $\delta_\epsilon(\phi)$ results from attempts to extend the evolution to all the level sets in order to segment interiors of objects such as the circle embedded inside the square shown in figure 1.

We propose a more direct approach for extending the evolution to surfaces by considering the following ordinary differential equation (ODE)

$$\frac{d\phi}{dt} = -\lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2 \qquad (6)$$

obtained by setting $\delta_\epsilon(\phi) = 1$ in equation (4) and ignoring the first (regularization) term. If we define the support of $\delta_\epsilon(\phi)$ to be that of the delta distribution, then *only* the zero level set of ϕ will evolve according to the Mumford-Shah functional. Setting $\delta_\epsilon(\phi) = 1$ allows the evolution to be extended to all level sets of ϕ.

Note that c_1 and c_2 are updated via (3) as ϕ evolves. Figure 2 illustrates the surface evolution dictated by this equation. In the first (upper left) image we have an initial surface representing the circle that surrounds the outside of the square in figure 1. Denoting the right hand side of equation (6) by V, we note that the values of ϕ increase (decrease) when V is positive (negative). As the evolution proceeds forward in time the object boundaries are represented by discontinuities in ϕ (not surprisingly, see [30]) as shown in the last (lower right) image in figure 2. Since in this case we do not care about the accuracy of the evolution, but only the steady state result, we can take rather large time steps. Moreover, since we only care about the discontinuities in ϕ, which represent the boundaries of the segmented image, we can ignore the ODE entirely and obtain the same result by setting $\phi = 1$ when $V > 0$ and setting $\phi = -1$ when $V < 0$. Again, we note that the constants c_1 and c_2 depend on ϕ, and thus this process has to be repeated updating c_1 and c_2 for each iteration. In practice, we have found that only one or two iterations are needed to reach convergence. Finally, we note that choosing $\phi = \mp 1$ allows for an efficient computation of the constants: $c_1 = \sum u_0(\phi+1)/\sum(\phi+1)$ and $c_2 = \sum u_0(\phi-1)/\sum(\phi-1)$.

The k-Means procedure [19] for two clusters starts with placing randomly two centers of mass m_1 and m_2, which corresponds to initializing the level set and defining the centers of mass as the mean image intensity values inside and outside ϕ, i.e. $m_1 = c_1$ and $m_2 = c_2$. Then the following two steps are processed until there are no changes. First, clusters are defined using a minimum-distance classifier to separate them. This step corresponds to the fitting terms in (6) that compare the distances of each pixel to the average image value of each cluster (with the λ_i giving the ability to weight the clusters). Second, the centers of mass are replaced by the new mean of the cluster. This corresponds to the computation of the new constants c_i.

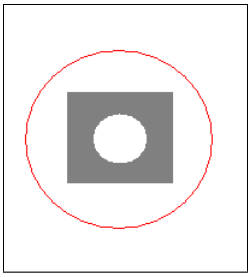

Figure 1: A two-region image. The initial $\phi = 0$ isocontour is shown as a red circle in this case.

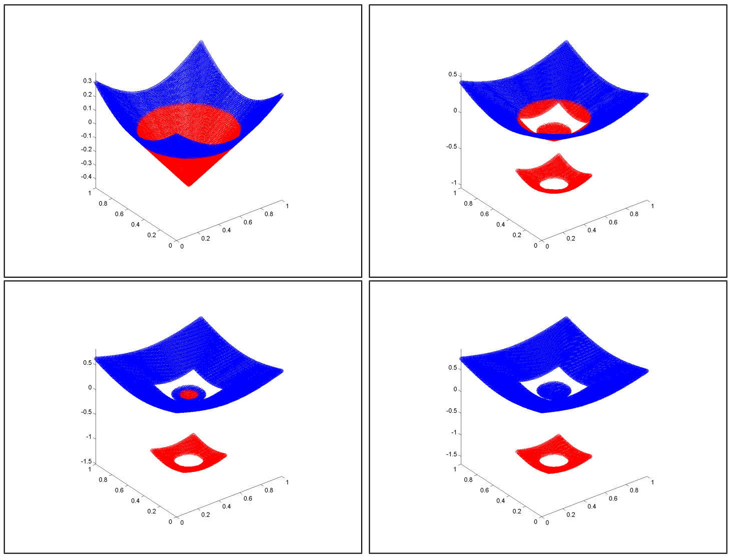

Figure 2: Evolution of the surface during the segmentation of the image shown in figure 1 at four different times. The negative values of ϕ are depicted in red while the positive values are depicted in blue.

2.3 Introducing a Length Scale

In the case of noisy images or for the grouping of objects, the notion of length scale is necessary. In this section, we show that a possible pre-processing step to re-introduce the notion of scale is obtained by using (at most) a few time steps of nonlinear diffusion instead of solving for the computationally expensive motion by mean curvature of equation (4). In order to optimize the computational efficiency of this preprocessing step, we use an implicit backward Euler time integration scheme on a linearized approximation of the nonlinear diffusion equation at each time step. The amount of nonlinear diffusion can be tuned with a single parameter in real time allowing the user to decide on the appropriate denoising and grouping of objects that mimics the desired adaptation and classification properties of the human visual system. For example, we show that results similar to those in [6] can be obtained in this fashion with the obvious gain in CPU efficiency.

Perona and Malik [29] introduced the use of isotropic (note-there is no directional anisotropy) nonlinear diffusion for denoising images while still keeping the image edges intact. The nonlinear equation they solved is

$$\frac{\partial I(\vec{x},t)}{\partial t} = \nabla \cdot (g(|\nabla I|)\nabla I)\,, \qquad (7)$$

where $I(\vec{x},t)$ defines the image intensity map at the voxel location $\vec{x}$ and fictitious time t, and g is an edge-stopping function chosen such that $\lim_{s\to\infty} g(s) = 0$ so that diffusion stops at the location of large gradients. Based on the original function proposed by Perona and Malik [29], we take $g(s) = \nu/(1+s^2/K^2)$, where K is a threshold parameter tuning the edge-stopping sensitivity on the image gradient and ν is a parameter controlling the length scale. In our work, ν replaces the curvature coefficient μ of equation (4) and plays the same role, i.e. large values of ν translate into larger elastic forces that hold the curve together (denoising images and grouping objects). Although one might also tune the parameter K, so far we have always used $K = 7$ in our examples.

Consider a mesh superimposed on the image in such a way that each pixel corresponds to a grid node (x_i, y_j). We solve equation (7) with the following fully conservative semi-implicit numerical discretization,

$$I^{n+1}_{i,j} - \triangle t(\nabla \cdot (g^n \nabla I^{n+1}))_{i,j} = I^n_{i,j},$$

where $g^n = g(|\nabla I^n|)$ is defined at time t^n to linearize the numerical approximation. We use standard central difference approximations for the derivatives and we obtain a system of equations for the unknowns $I^{n+1}_{i,j}$ that is both symmetric and linear, and thus can be solved using a robust and fast iterative solver. We use a preconditioned conjugate gradient method with a modified Cholesky preconditioner, see e.g. [31]. Moreover, this linear system does not need to be solved exactly, but only for a few iterations until the residual is reasonably small. Indeed, it is well known that the conjugate gradient algorithm for solving a linear system can be related to a steepest decent type minimization algorithm for optimization indicating that one can obtain useful results without iterating to convergence. Of course, ADI [38] or AOS [45] methods could be used here as well, although our approach allows one to tune the accuracy by choosing the number of iterations whereas AOS always incurs the operator splitting error.

We note that this replacement of curvature by nonlinear diffusion is similar to [14] in principle but different in practice since they apply the diffusion to the level set and we apply it to the image. Moreover, instead of applying the diffusion to ϕ we evolve motion by mean curvature directly, but occasionally, as we discuss next.

3. A Hybrid k-Means Level Set Algorithm

The speed gained by the k-Means procedure comes with its drawbacks, i.e. the results produced depend on the initial values for the means and suboptimal partitions are frequently found. Moreover, the partitioning may be noisy with an amount of noise too large for the smoothing preprocessing to be effective. Likewise, the length scale introduced by the nonlinear diffusion preprocessing is limited to the case where the features to segment are not significantly separated. The properties of curve evolution, i.e. the ability to preserve the dependence on initial data and to include motion by mean curvature in order to provide elastic properties to the curve, have received considerable attention and can alleviate some of these issues. We propose a simple algorithm that retains the robustness of curve evolution but that is faster by orders of magnitude. It consists of the following three steps:

- Initialize ϕ as a signed distance function,
- Evolve the ODE
$$\frac{d\phi}{dt} = V(\vec{x}), \tag{8}$$
- Periodically use the Fast Marching Method [43, 33] to re-initialize ϕ to a signed distance function and/or apply motion by mean curvature for a few explicit time steps (or semi-implicitly as in [12, 37]).

Here, the velocity field V defines the association of each pixel/voxel to one cluster or the other and may depend on different statistical measures, geometrical curve properties, probabilistic models, texture, etc.. Obvious generalization to vector valued functions and partitioning into more that two regions can be obtained with several level set functions.

Evolving the curve using equation (8) does not require any time step restrictions since an implicit method could be used. In fact, the solution obtained solving it in one step, only looking for a steady state, is k-Means as discussed above. Thus, in this case the ODE is very weakly dependent on the initial data depending only on the sign of ϕ, not on its value. This ODE can be given a much stronger dependence on the initial data by using an accurate time integration method. From a geometrical point of view, a pixel/voxel far away from the interface is not likely to cross over it. However, if the underlying energy (i.e. V) is strong enough this can happen. This balance between a pixel/voxel's proximity to the interface and the strength of the underlying energy is characteristic of curve evolution and can produce a preferable steady state. We note that since the initial data is a signed distance function, the result looks like a level set solution to curve evolution. For example consider a cone with apex above the plane moving downward, corresponding to a circle moving inward. If the right hand side of (8) was spatially constant we would exactly get the curve evolution answer.

Standard level set methods are slow because of time step restrictions and the fact that the spatial terms need to be evaluated at every time step. However, our hybrid procedure only re-initializes as often as the spatial curve evolution dependence is desired and does not suffer from time step restrictions to ensure stability, translating into a method that is orders of magnitude faster and much simpler. In addition, besides re-initializing every so often, one could apply motion by mean curvature for a few time steps, etc. We reiterate that explicit time stepping is needed in order to obtain accurate result for the motion by mean curvature, even though semi-implicit solvers [12, 37] might be enough for regularizing the evolution.

The philosophy behind our method is that one can evolve a curve most of the time with an ODE solver, allowing for large time steps and no evaluation of spatial terms. Then, one can direct the solution to a more preferable local minima using spatial information only occasionally. This gives a handle on the compromise between speed and accuracy and allows for segmentation results ranging from k-Means clustering type partitioning to level set solutions.

4. Examples

The connection we made suggests that one can employ an algorithm that is orders of magnitude faster than that presented in [6] to segment images with a small amount of noise and to group objects that are close enough together. We consider examples taken from [6] for the sake of comparison. The computational efficiency of the method is illustrated by reporting the CPU times on a Pentium IV laptop.

4.1 Findings Edges

One of the many areas of research that produce large volumes of data is PLIF (Planar Laser-Induced Fluorescence) imaging. The Hemisquilla ensiguera californica is a type of stomatopod that relies on olfaction to locate mates, food sources and to avoid predators. Studying this stomatopod leads to tens of thousands of images of plumes of dye in turbulent flows as illustrated for example in figure 3. Segmenting this data and finding characteristics such as points of maximum concentration, width in the normal or user-defined direction, onset slopes at the segment boundaries, etc. gives vital information about the signal available to the searching animals at different distances from the source and from different hydrodynamic conditions. Also, computing geometrical features, such as the curvature and the length of a plume, helps to connect the plume shape to the physics of the flow. The interested reader is referred to [10, 11] for a detailed exposition on this research.

The use of our algorithm enables researchers to process the large amount of data they are confronted with in a timely fashion and enables them to draw a probability density functional associated with their problem. Geometrical properties such as curvature and length are easily computed after transforming the discontinuous implicit surface obtained from our segmentation process into a signed distance function using the fast marching method [43, 33]. The implicit surface framework offers other advantages as well. Moreover, one can convert an implicit surface into an explicit triangulated surface model using a marching cubes algorithm, e.g. see [18, 7], or into a tetrahedralized volume that can be used for finite element simulation, see e.g. [22].

Figure 3 depicts the segmentation of an image along with the shape of the two different regions. Since the image is free of noise, it is not necessary to preprocess the image with anisotropic diffusion. A typical CPU time for an image of this size, 924×528 pixels, is less than .2 seconds.

4.2 Three-Dimensional Images

The ability to treat large amounts of data is also critical in the case of three-dimensional objects. Figure 4 depicts the three-dimensional reconstruction of the left hemisphere of the brain from two-dimensional slices. The image size is $192 \times 128 \times 61$ and the CPU time is .3 seconds. Figure 4 (right) demonstrates that detailed features inside the brain are correctly segmented. Figure 5 depicts the three dimensional reconstruction of a turbulent flow from two-dimensional PLIF imaging data. The image size is $500 \times 200 \times 65$ and CPU time is 1.5 seconds. In this example, geometrical properties such as curvatures are easily computed.

4.3 Blurry Images

One of the main advantages of the Mumford-Shah functional is that it is particularly well-suited for segmentation of images without clearly defined edges. In the case of a blurry image, algorithms searching for a steep image gradient are guaranteed to fail. Figure 6 illustrates that our algorithm retains this desirable capability. The image size is 320×220 and the CPU time is .06 seconds.

4.4 Multiple Regions Images

This example demonstrates the straightforward extension of our algorithm to multiple region segmentation. We consider a slice of the left hemisphere of a brain and seek to outline four different regions. We use two different implicit surfaces, as proposed in [44], and evolve each of them as described above. The results are shown in figure 7. The image size is 256×171 and the CPU time is .16 second.

4.5 Noisy Images

The presence of noise in an image is easily filtered out by the anisotropic diffusion preprocessing stage, especially since such a treatment was specifically designed to denoise images while preserving edges. Figure 8 illustrates the segmentation of a military aircraft in a noisy image. In this example we take $\nu = .4$ (and $K = 7$). The image size is 311×161 and the CPU time is .14 seconds.

4.6 Grouping Objects

The following examples illustrate the capability of our method to reproduce results comparable with those published in [6] for grouping objects. Figure 9 depicts the effect of varying ν (note that $K = 7$ in every test) in the case where the image is composed of two clusters of three disks. By setting $\nu = 0$ one can segment each disk individually (top right), and by increasing its value to $\nu = 1$ one can group each cluster (bottom left). Increasing its value again to $\nu = 4$ provides even more 'elastic' tension forces and groups the clusters without wrapping closely to the boundaries (bottom right). The original image (left) is 124×145 in size and the CPU times are .03, .21 and .31 seconds respectively. These fast CPU times allow one to tune the desired effect in real time.

Figure 11 simulates the segmentation of the location of a simulated mine-field. In this example, we use the anisotropic diffusion preprocessing with $\nu = 1$ (and $K = 7$). The image size is 128×128 and the CPU time is .05 seconds.

In figure 10 we first set $\nu = 0$ to identify detailed structures of the galaxies and individual stars (upper right). By increasing the value of ν (again $K = 7$ in all tests), individual stars are ignored and each galaxy is segmented separately (lower left with $\nu = 1$). Again increasing ν to $\nu = 4$ groups the two galaxies together as a single cluster (lower right). The image size is 141×141 and the CPU times are .01 seconds, .04 seconds and .05 seconds respectively.

4.7 Extension to Other Statistical Measures

The piecewise constant Mumford-Shah model was originally proposed to separate images into regions about their mean intensity values, but one is not bound to that particular statistical measure to discriminate features. The constants c_1 and c_2 might be chosen to represent the statistical component about which we seek to separate the different regions. For example, we seek a descriptor for the locally dominant scale of a discrete signal in order to facilitate the extraction of segments exhibiting distinguishable degrees of variability [42]. The scale descriptor we consider is given by:

$$Scale \approx \sum_{l=1}^{L} l \sigma_{x_{[2^{-l-1}, 2^{-l}]}},$$

where σ_{x_I} is the standard deviation of the input signal component corresponding to the frequency interval I.

Figure 12 depicts the segmentation of a map of Europe from its city lights. We first preprocess the image with the scale defined above and then apply our algorithm with $\nu = 1$ (and $K = 7$). The image size is 597×360 and the CPU time is .58 second.

4.8 Curve Evolutions Properties

We then turn our attention to cases where the nonlinear diffusion plus k-Means fails and where the properties of curve evolution allow for the intended segmentation. We first consider the case of the segmentation of an object with no discontinuous boundary. In this example a nonlinear diffusion preprocessing would not be able to re-introduce the notion of length scale as desired. Figure 13 illustrates that motion by mean curvature can properly introduce the desired length scale. In this example we alternate the evolution of equation (6) with a few time steps of motion by mean curvature, i.e. one step of the evolution of equation (6) is followed by few steps of motion by mean curvature. We re-iterate that the evolution of equation (6) is not hampered by a stringent stability time step restriction of parabolic nature as in [6].

Second, we consider the case where only the exterior of an object is to be segmented. Application of this process is often needed in medical applications. Dependence on the initial data is necessary in this case, mimicking the procedure of the snakes algorithm. However, detection of gradients (as in snakes) often requires the user to correctly guess the value of a threshold parameter making the method delicate to use in a non-supervised setting and can even fail if the boundary is not described by a uniform threshold value. We instead use the Mumford-Shah functional criterion. We apply our hybrid algorithm, i.e. start with a signed distance function outside the object, and occasionally re-initialize the level set while solving equation (6) in order to prevent ϕ from segmenting the inside of the object. Figure 14 depicts the segmentation of the outside of the brain hemisphere using such a method.

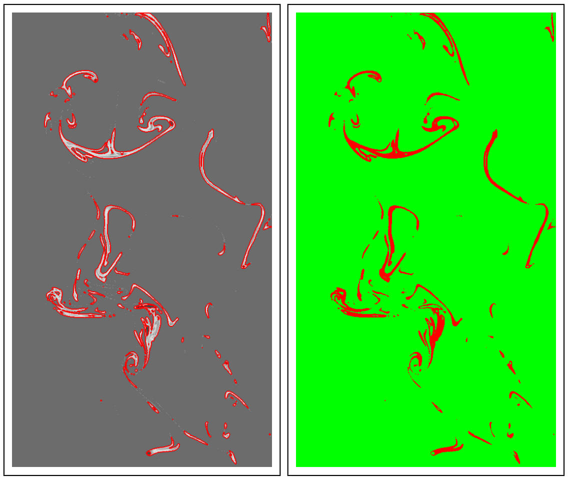

Figure 3: Segmentation of a plume of dye in a turbulent flow. Original image with the segmentation outlined in red (left) and regions defined by the segmentation (right). The image size is 528×924 and the CPU time is .16 seconds.

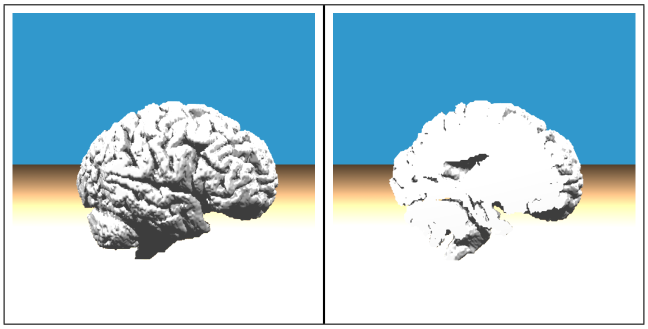

Figure 4: Three-dimensional reconstruction of the left hemisphere of a brain (top). The image size is $192 \times 128 \times 61$ and the CPU time is .3 second. A cross section (bottom).

5. Conclusions

We have made a connection between a level set algorithm and k-Means plus nonlinear diffusion. Numerical examples demonstrate that these two process combined together pro-

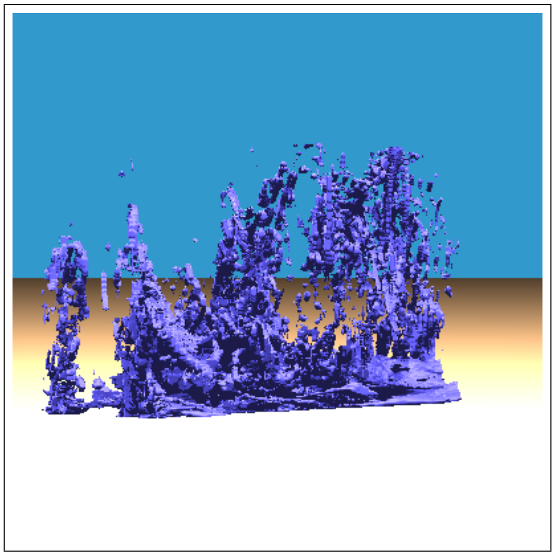

Figure 5: Three-dimensional reconstruction of a turbulent flow. The image size is $500 \times 200 \times 65$ and CPU time is 1.5 seconds.

duce results comparable to [6] with significantly less computational effort. Then we have proposed a novel hybrid numerical algorithm that draws on the speed and simplicity of k-Means procedures, and the robustness of level set algorithms. In particular, our method is orders of magnitude faster than curve evolution techniques while retaining spatial coherence on initial data as well as the balance between a pixel/voxel's proximity to the curve and its intention to cross over the curve from the underlying energy. Regularization processes such as motion by mean curvature are easily included. Moreover, it does not suffer from the limitations of k-Means due to inaccurate local minima and allows for segmentation results ranging from k-Means clustering type partitioning to level set solutions.

Acknowledgments

This work was supported in part by an NSF Postdoctoral Fellowship (DMS 0102029), an ONR YIP award and a PECASE award (ONR N00014-01-1-0620), a Packard Foundation Fellowship, ONR N00014-97-1-0027 and NSF ITR-0121288. The authors would like to thank Chris Bregler, Ron Kimmel, Stan Osher, Guillermo Sapiro, Harry Shum, Demetri Terzopoulos and Luminita Vese for helpful communications on the segmentation problem. Special thanks go to Stan Osher and Luminita Vese for the images used in the length scale examples, Jeffrey Koseff and Meg Wiley for giving us the plume data, and Eftychios Sifakis for computing the statistical map in the Europe night lights example.

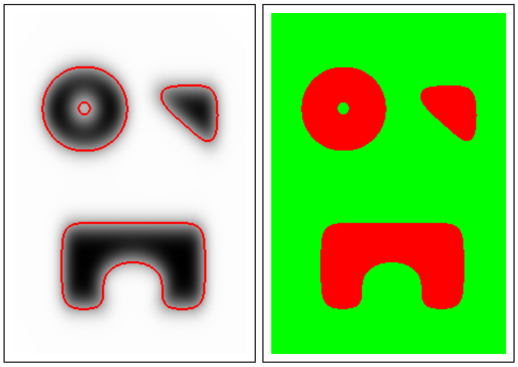

Figure 6: Original blurry image with the segmentation outlined in red (left) and regions defined by the segmentation (right). The image size is 320×220 and the CPU time is .06 seconds.

References

[1] Adalsteinsson, D. and Sethian, J., "A Fast Level Set Method for Propagating Interfaces," *J. Comput. Phys.*, 118 (1), 269-277, 1995.

[2] Adams, R. and Bischof, L., "Seeded Region Growing," *IEEE Trans. Pattern Anal. Machine Intell*, 16 (1), 641-646, 1994.

[3] Canny, J.F., "A Computational Approach to Edge Detection," *Trans. Pattern Anal. Machine Intell*, 8 (1), 679-698, 1986.

[4] Caselles, V., Catte, F. Coll., T. and Dibos F., "A Geometric Model for Active Contours in Image Processing", Numerische Mathematik, 66, 1-31, 1993.

[5] Caselles, V., Kimmel, R. and Sapiro, G., "Geodesic active contours," *International Journal of Computer Vision*, 22 (1), 61-79, 1997.

[6] Chan, T. and Vese, L., "Active Contour Without Edges," *IEEE Transactions on Image Processing*, 10 (2), 266-277, 2001.

[7] Cignoni, P., Ganovelli, F., Montani, C. and Scopigno R., "Reconstruction of Topologically Correct and Adaptive Trilinear Isosurfaces," *Computers & Graphics*, 24 (3), 339-418, 2000.

[8] Cohen, L.D, "On Active Contour Models and Balloons," *CVGIP: Image Understanding*, 53 (2), 211-218, 1991.

[9] Comaniciu, D. and Meer, P., "Mean shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. on Image Processing*, 10, Issue 10, 1467-1475, 2001.

[10] Crimaldi J. and Koseff, J., "High-resolution Measurements of the Spatial and Temporal Scalar Structure of a Turbulent Plume," *Experiments in Fluids*, 31 (1), 90-102, 2001.

[11] Crimaldi J., Wiley, M. and Koseff, J., "The relationship between mean and instantaneous structure in turbulent passive scalar plumes," *Journal of Turbulence* , 3 , 1-24, 2002.

[12] Douglas, J.J. and Dupont, T., "Alternating-Direction Galerkin Methods on Rectangles," Numerical Solution of Partial Differential Equations, Ed B. Hubbard, Academic Press, New York, 133-213, 1971.

[13] Fua, P. and Leclerc, Y. G., "Model Driven Edge Detection," *Machine Vision and Applications*, 3 (1), 45-56, 1990.

[14] Goldenberg, R., Kimmel, R., Rivlin, M. and Rudzsky, M., "Fast Geodesic Active Contour," *IEEE Trans. Pattern Anal. Machine Intell*, 24, 603-619, 2002.

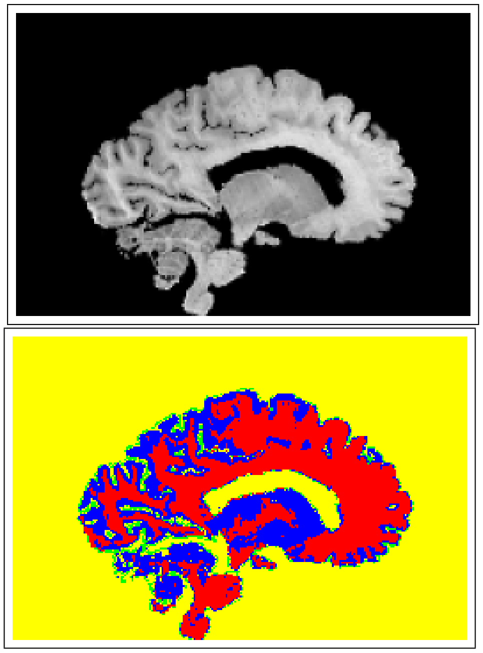

Figure 7: We use two implicit surfaces to separate four different regions of the brain. Original image (top) and segmented regions (bottom). The image size is 256×171 and the CPU time is .16 seconds.

[15] Kass, M., Witkin, A. and Terzopoulos, D., "Snakes: Active Contour Models," *International Journal of Computer Vision*, 1 (4), 321-331, 1987.

[16] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A., "Gradient Flows and Geometric Active Contour Models," *ICCV*, 1995.

[17] Leventon, M., Grimson, W., and Faugeras, O., "Statistical Shape Influence in Geodesic Active Contour," *Comp. Vision and Patt. Recon.*, (CVPR), 2000.

[18] Lorensen, W., and Cline, H., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *SIGGRAPH*, ACM Press/ACM SIGGRAPH, Comp. Graphics Proc., 21, 163-170, 1987.

[19] MacQueen, J., "Some methods for classification and analysis of multivariate observation", *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 281-297, 1967

[20] Malladi, R., Sethian, J. A. and Vemuri B.C., "Shape Modeling with Front Propagation: A Level Set Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 (2), 158-175, 1995.

[21] McInerney, T. and Terzopoulos, D., "Deformable Models in Medical Image Analysis: A Survey," *Medical Image Analysis*, 1 (2), 91-108, 1996.

[22] Molino, N., Bridson, R. and Fedkiw, R., "Robust Volumentric Physics-Based Meshing from Level Set Data," (in preparation).

[23] Mumford, D. and Shah, J., "Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems," *Communication in Pure and Applied Mathematics*, 42 (1), 577-685, 1989.

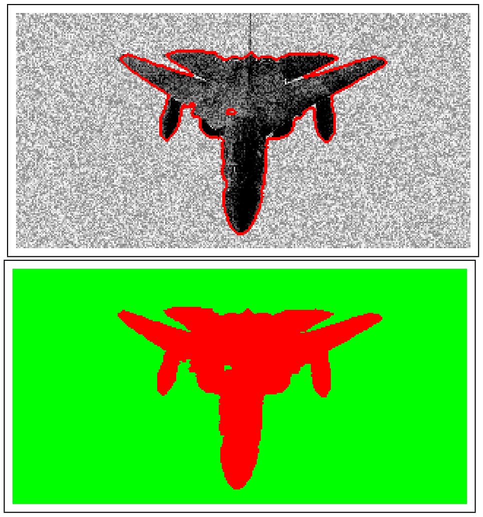

Figure 8: Original noisy image with the segmentation outlined in red (top) and regions defined by the segmentation (bottom). The image size is 311×161 and the CPU time is .14 seconds. In this example, we set $\nu = .4$.

[24] Osher, S. and Fedkiw, R., "Level Set Methods and Dynamic Implicit Surfaces," Springer-Verlag, New York, (2002).

[25] Osher, S. and Sethian, J., "Front Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulation," *J. Comput. Phys.*, 79 (1), 12, 1988.

[26] Paragios, N. and Deriche, R., "A PDE-based Level Set Approach for Detection and Tracking of Moving Objects," *Proceedings of the* 6^{th} *ICCV, Bombay, India*, 1998.

[27] Paragios, N. and Deriche, R., "Geodesic Active Regions for Texture Segmentation," INRIA RR-3440, 1998.

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

[29] Perona, P. and Malik, J., "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 (1), 629-639, 1990.

[30] Rudin, L., "Images, Numerical Analysis of Singularities, and Shock Filters," *Ph.D Thesis*, Comp. Sci. Dept., Caltech #5250:TR:87, 1987.

[31] Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS publishing company, Boston 1996.

[32] Sapiro, G., "Geometric Partial Differential Equations and Image Analysis," Cambridge University Press, New York, 2001.

[33] Sethian, J.A., *A Fast Marching Level Set Method for Monotonically Advancing Fronts*, Proc. Nat. Acad. Sci, 93 (4), 1591-1595, 1996.

[34] Sethian, J.A., *Level Set Methods and Fast Marching Methods*, Second Edition, Cambridge University Press, Cambridge, UK, 1999.

[35] Shi, J. and Malik, J., "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (8), 888-905, 2000.

[36] Sifakis, E., Garcia, C., and Tziritas, G., "Bayesian Level Sets for Image Segmentation," *Journal of Visual Communication and Image Representation* 13 (1), 44-64, 2002.

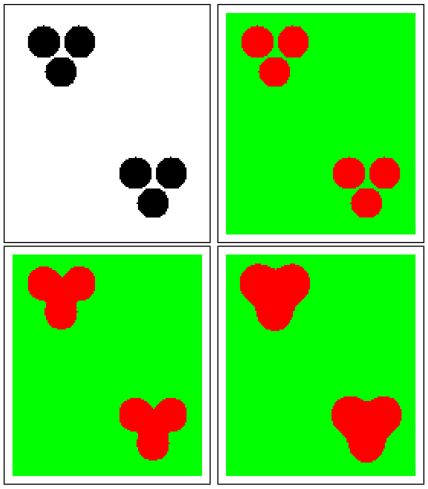

Figure 9: Results showing the effect of varying ν during the nonlinear diffusion preprocessing stage. The original image (top left) is 124×145 in size and the CPU times are .03, .21 and .31 seconds for $\nu = 0$ (top right), $\nu = 1$ (bottom left) and $\nu = 4$ (bottom right) respectively.

[37] Smereka, P., *"Semi-Implicit Level Set Methods for Curvature and Surface Diffusion Motion"*, Journal of Scientific Computing (in press). See www.math.lsa.umich.edu/ psmereka.

[38] Strikwerda, J., "Finite Difference Schemes and Partial Differential Equations," Wadsworth & Brooks/Cole Advanced Books and Software, Pacific Grove, California, (1989).

[39] Terzopoulos, D., and Fleischer, K., "Deformable Models," *The Visual Computer*, 4 (6), 306-331, 1988.

[40] Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K., "Elastically Deformable Models," *SIGGRAPH*, ACM Press/ACM SIGGRAPH, Comp. Graphics Proc., 205-214, 1987.

[41] Terzopoulos, D., Witkin, A. and Kass, M., "Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion," *Artificial Intelligence*, 36 (1), 91-123, 1988.

[42] Unser, M., "Texture Classification and Segmentation Using Wavelet Frames," *IEEE Transactions on Image Processing*, 4 (11), 1549-1560, 1995.

[43] Tsitsiklis, J., "Efficient Algorithms for Globally Optimal Trajectories," *IEEE transactions on Automatic Control* 40 (1), 1528-1538, 1995.

[44] Vese, L. and Chan, T., "A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model," *International Journal of Computer Vision* 50 (3), 271-293, 2002.

[45] Weickert, J., Romeny B. and Viergever, M., "Efficient and Reliable Scheme for Nonlinear Diffusion Filtering," *IEEE Transactions on Image Processing*, 7 (3), 398-410, 1998.

[46] Yezzi, A., Tsai A. and Willsky, A., "A Fully Global Approach to Image Segmentation via Coupled Curve Evolution Equations," *Journal of Visual Communication and Image Representation*, 13, 195-216, 2002.

[47] Zucker, S.W., "Relaxation labeling: 25 years and still iterating," *Foundations of Image Understanding*, L. S. Davis (ed.), Kluwer Academic Publ, Boston, 289-322, 2001.

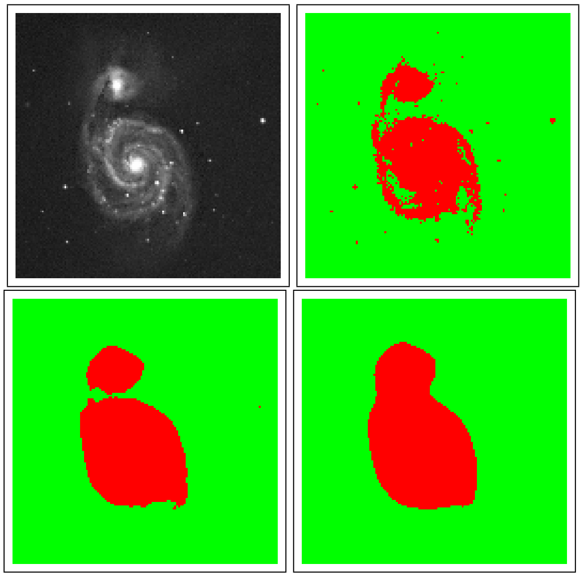

Figure 10: Segmentation of two nearby galaxies (the original image is top left). The tuning of ν allows us to segment each galaxy individually and the stars around them (top right, $\nu = 0$), each individual galaxy but not the isolated stars (bottom left, $\nu = 1$), or both the galaxies as one cluster (bottom right, $\nu = 4$). The image size is 141×141 and the CPU times are .01 seconds, .04 seconds and .05 seconds respectively.

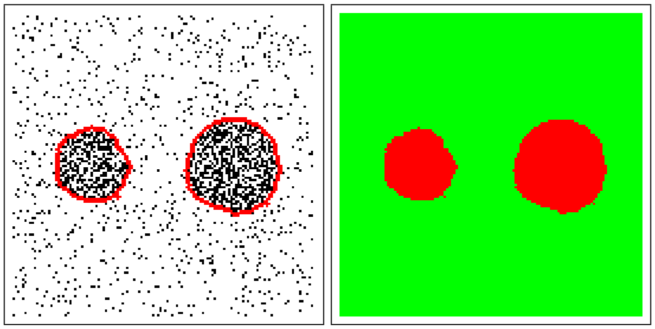

Figure 11: Original image with the segmentation outlined in red (left) and regions defined by the segmentation (right). The image size is 128×128 and the CPU time is .05 seconds. We use the nonlinear diffusion preprocessing with $\nu = 1$.

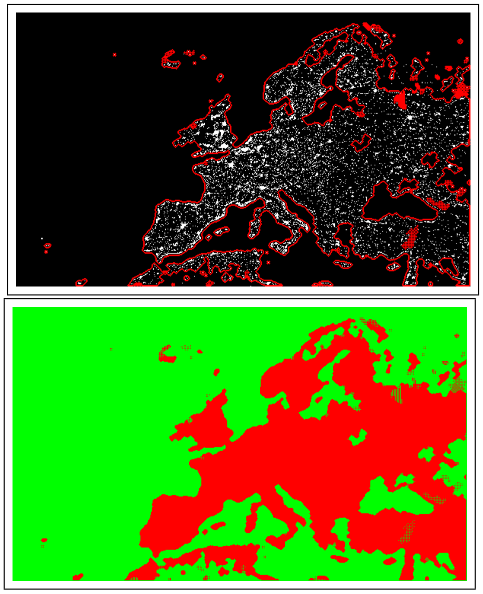

Figure 12: Europe lights segmentation (top) and regions (bottom). We take $\nu = 1$ in the preprocessing stage. The image size is 597×360 and CPU time is .58 second.

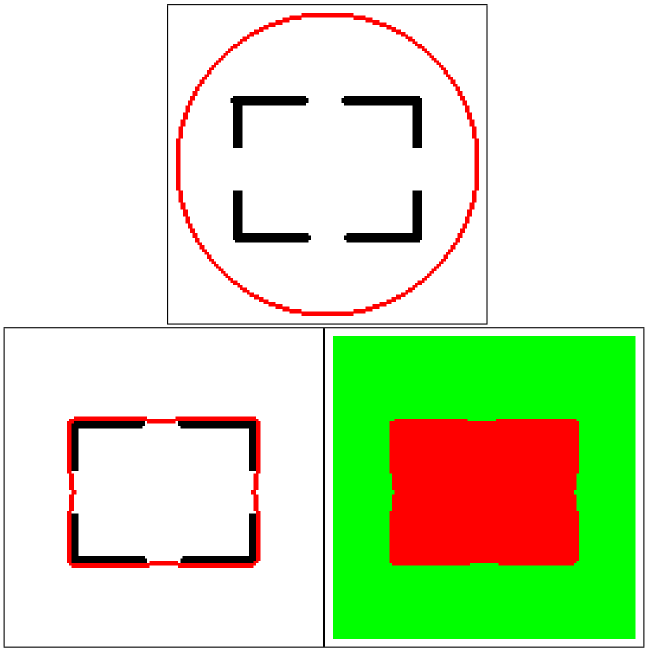

Figure 13: Original image with initial level set around it (top). Segmentation (middle) and regions (bottom). Note that the effect of curvature is desired in this example.

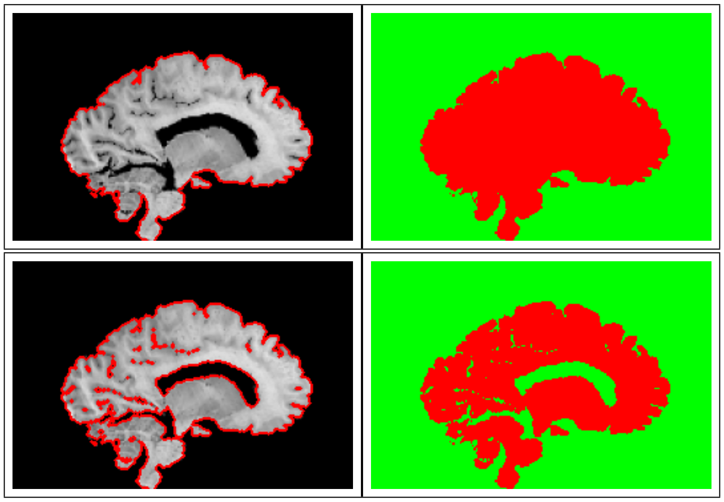

Figure 14: Segmentation of the external surface of the left hemisphere of the brain (top left) and corresponding regions (top right). The image size is 192×128 and the CPU time is 2 seconds. The bottom images are the segmentation (left) and regions (right) obtained using k-Means. In this case, it is not possible to outline *only* the outside.