JOINT SEGMENTATION AND REGISTRATION USING LOGIC MODELS†

Mark Moelich and Tony Chan

Mathematics Department, UCLA
405 Hilgard Avenue, Los Angeles, CA 90095
{mmoelich, chan}@math.ucla.edu

February 10, 2003

ABSTRACT An algorithm is presented for the joint segmentation and registration of images using logic models. The algorithm is based on the ideas presented by Yezzi, Zöllei, and Kapur [23]. We introduce three improvements. The logic models of Sandberg and Chan [17] are used for segmentation, instead of the multi-channel Chan-Vese algorithm, because they allow better control of the segmentation and provide a richer context in which to interpret dissimilar images. The second change is to use two parameterized mappings instead of one for registration. The third change is to use an initial registration phase to handle poor initial guesses of the registration parameters. All three changes improve both the robustness and fidelity of the algorithm. This algorithm also significantly enlarges the scope of applications for the logic models because it relaxes the requirement that the images be registered.

KEYWORDS Segmentation, registration, logic, multi-channel, active contour, level-set.

1 INTRODUCTION

It is often difficult to isolate an object in an image through segmentation when the image of the object is either incomplete or is in close proximity to a similar object. A second image of the object can often be used to fill in the missing parts of the object or to act as a template. Combining images in this way usually requires us to register (align) them before applying a segmentation algorithm. But this, in turn, requires some sort of preliminary segmentation or matching of landmarks.

Recently Yezzi, Zöllei, and Kapur presented a unified approach to this problem in their pioneering paper [23]. Their algorithm jointly segments and registers two images by "interleaving" the processes of segmentation and registration. We build upon their approach in three ways.

The logic models of Sandberg and Chan [17] are used for segmentation instead of the multi-channel Chan-Vese algorithm [6]. The multi-channel Chan-Vese algorithm has multiple minima when the objects do not appear identical in each channel. This may not be a problem for many applications, but is a potential problem for a joint segmentation and registration algorithm because the segmentation contour can get trapped in the wrong minima before registration is complete. The logic models, on the other hand, do not have this problem due to the way the energies are defined.

We improve the registration process by using two parameterized mappings, instead of one, and by introducing an initial registration phase. The use of two mappings allows the algorithm greater latitude in avoiding local minima. The initial registration phase allows the algorithm to handle large errors in the registration parameters, when an estimate of the intensity of the object of interest is available.

These three changes improve the robustness and fidelity of the algorithm. The most important change is the use of logic models because they allow greater control over the registration process and provide a richer context in which to interpret dissimilar images. This algorithm also enlarges the scope of applications for the logic models because it relaxes the requirement that the images be registered.

† This work has been supported in part by ONR contract N00014-96-1-0277, NSF contract DMS-9973341, and NIH contract P20 MH65166.

2 REGION-BASED SEGMENTATION

2.1 Overview

The scalar Chan-Vese algorithm, the multi-channel Chan-Vese algorithm, and the logic models are all region-based segmentation algorithms. These models segment an image by finding a contour C that minimizes and energy of the form

$$E(C) \;=\; E^{in}(C) \;+\; E^{out}(C) \;+\; \mu \,\text{length}\,(C), \tag{1}$$

where

$$\begin{aligned} E^{in}(C) &= \int_{\text{int}(C)} F^{in}(\mathbf{x})\,d\mathbf{x} \\ E^{out}(C) &= \int_{\text{ext}(C)} F^{out}(\mathbf{x})\,d\mathbf{x}, \end{aligned} \tag{2}$$

where $\text{int}(C)$ and $\text{ext}(C)$ refer to the regions interior and exterior to C, respectively, and where μ is a design parameter. The forms of the functions $F^{in}(\mathbf{x})$ and $F^{out}(\mathbf{x})$ define the segmentation algorithm. For these algorithms, the gradient flow associated with the energy (1) is given by

$$\frac{\partial C(\mathbf{x})}{\partial t} = (\,F^{in}(\mathbf{x}) \;-\; F^{out}(\mathbf{x}) \;+\; \mu\,\kappa\,)\,\mathbf{N}, \tag{3}$$

where κ is the curvature at point $\mathbf{x}$ and $\mathbf{N}$ is the inward pointing unit normal.

To minimize the energy (1), we can begin with an initial contour C, and evolve it along the gradient flow (3) until a minima is achieved. Figure 1 below illustrates this evolution for the scalar Chan-Vese algorithm [7]. In practice, this equation can be solved using the level set theory of Osher and Sethian, where the contour C is modeled as the zero level set of a Lipschitz continuous function ϕ. The evolution of C is computed by solving a partial differential equation in ϕ. Further details can be found in [15], [19], [16], [7], [18], and [2].

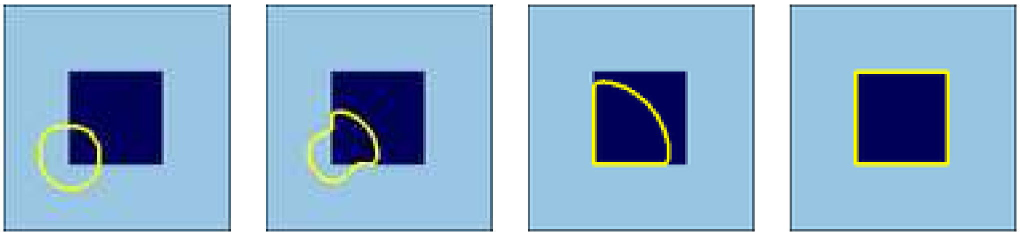


Figure 1: Example of contour evolution in the Chan-Vese algorithm ($\approx$ 10 iterations/frame).

2.2 Scalar and Multi-Channel Chan-Vese Algorithms

In the scalar Chan-Vese algorithm [7],

$$F^{in}(\mathbf{x}) = \lambda^{in}\left(\,I(\mathbf{x}) - c^{in}\,\right)^2 \quad \text{and} \quad F^{out}(\mathbf{x}) = \lambda^{out}\left(\,I(\mathbf{x}) - c^{out}\,\right)^2, \tag{4}$$

where $I(\mathbf{x})$ is the intensity of the image at point $\mathbf{x}$, and c^{in} and c^{out} are the average intensities in the regions $\text{int}(C)$ and $\text{ext}(C)$, respectively, and λ^{in} and λ^{out} are design parameters.

The scalar Chan-Vese algorithm makes a piecewise constant approximation to the image. The functions F^{in} and F^{out} measure the error in this approximation. Considering the second image in the sequence shown in Figure 1 (see Figure 2 below), we see that the intensity at point a better matches c^{out} than c^{in}, so we have $F^{out}(a) < F^{in}(a)$ and the contour moves inward at point a according to the gradient flow (3). At point b, on the other hand, the intensity better matches c^{out} than c^{in}, so we have $F^{in}(b) < F^{out}(b)$ and the contour moves outward. In either case the contour moves toward the boundary of the object.

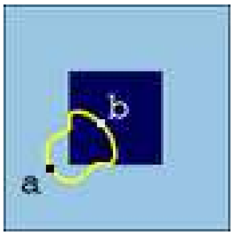


Figure 2: Contour moves inward at a and outward at b. In either case, the contour moves toward the toward the boundary.

The multi-channel Chan-Vese algorithm [6] uses the same concepts to segment composite images. A composite, or multi-channel, image is a collection of images that are registered, that is, the object of interest has a consistent position in each image. For the multi-channel Chan-Vese algorithm,

$$F^{in}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} F_i^{in}(\mathbf{x}) \qquad \text{and} \qquad F^{out}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} F_i^{out}(\mathbf{x}), \tag{5}$$

where

$$F_i^{in}(\mathbf{x}) = \lambda_i^{in}\left(I_i(\mathbf{x}) - c_i^{in}\right)^2 \qquad \text{and} \qquad F_i^{out}(\mathbf{x}) = \lambda_i^{out}\left(I_i(\mathbf{x}) - c_i^{out}\right)^2.$$

Note that although there are N channels, there is a single segmenting contour C. In this paper, we consider only the case where $N = 2$.

The strength of the multi-channel Chan-Vese algorithm is its ability to reconstruct an object from partial information. For example, if we consider the two images in Figure 3 to be incomplete images of a square, and if the initial contour contains the union of both objects, then the multi-channel Chan-Vese algorithm recovers the square. The contour C is drawn in yellow in both images.

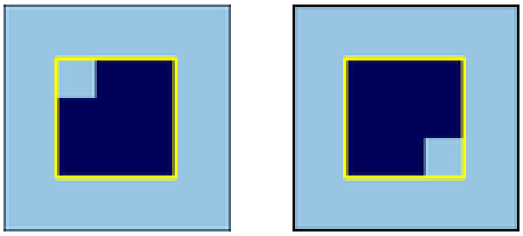


Figure 3: The contour was initially contained both objects and converged to the union (logical OR) of the objects.

2.3 Logical Interpretation of Images

When there is a disparity between the appearance of the object in each of the channels, as in Figure 3 above, there is more than one valid interpretation of the actual object. Figure 4 below illustrates two such interpretations of the images given in Figure 3. These alternate interpretations correspond to different logical interpretations of the images. The image on the left is the union (logical OR) of the objects in Figure 3, while the image on the right is the intersection (logical AND) of the objects.

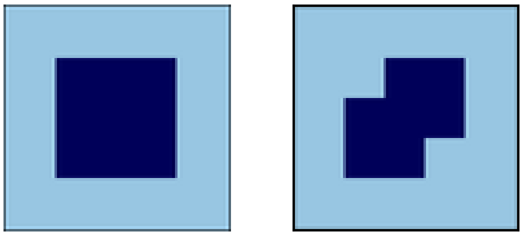


Figure 4: Two alternate interpretations of images in Figure 3. The union (logical OR) is on the left and the intersection (logical AND) is on the right.

In the multi-channel Chan-Vese algorithm, each of these interpretations correspond to different local minima in the segmentation energy. The multi-channel Chan-Vese algorithm does not differentiate between these logical interpretations, but simply converges to the minima which is "nearest" the initial contour. In Figure 3 above, the contour was initially contained in the union of the objects in each channel and converged to the union (logical OR). In Figure 5 below, on the other hand, the contour was initially contained in the intersection of the objects and converged to the intersection (logical AND). The resulting segmentation, then, depends upon the initial contour. It also depends upon the intensities of the image and upon the design parameters. This behavior may not be a problem for many applications, but is a potential problem for joint segmentation and registration since it can cause the contour to get trapped in the wrong minima before registration is complete. Because of this, we focus on the logical models of Sandberg and Chan [17] for segmentation.

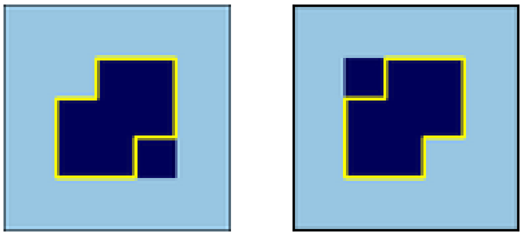


Figure 5: Initial contour was contained in the intersection (logical AND) of the objects.

2.4 Logic Models

The logic models, developed by Sandberg and Chan [17], are designed to segment multi-channel images according to logical interpretations. Selecting a particular logic model is the same as selecting the minima to which the contour will converge. In this paper, we consider the logical OR and the logical AND models, although this method extends to the other logic models as well. A comprehensive description of various logic models can be found in [17].

The logical OR and logical AND models are built from the two logical elements

$$\begin{aligned} \text{OR element:} \quad & \sqrt{F_1(\mathbf{x})\,F_2(\mathbf{x})} \\ \text{AND element:} \quad & 1 - \sqrt{(1-F_1(\mathbf{x}))\,(1-F_2(\mathbf{x}))}\,. \end{aligned} \tag{6}$$

These logical elements are continous analogues of the OR and AND operations of binary logic. The arguments F_1 and F_2 take on any real value between 0 and 1, where a value of 0 corresponds to no error and a value of 1 corresponds to the maximum error. By construction, the logical elements also take on values between 0 and 1. The truth table of these logical elements, for limiting values of the arguments, is given by

(a) OR element

F_1 \ F_2	0	1
0	0	0
1	0	1

(b) AND element

F_1 \ F_2	0	1
0	0	1
1	1	1

Table 1: Truth tables for the logical OR element (a) and logical AND element (b) for limiting values of the arguments. A value of 0 corresponds to no error and a value of 1 corresponds to the maximum error.

Note that if we consider no error (0) to be true and the maximum error (1) to be false, then these tables are the familiar truth tables of binary logic.

For the logical OR segmentation model, we apply the logical OR element to the interior of C and apply the logical AND element to the exterior of C. That is, we take the union of the images inside the contour and the intersection outside:

$$\begin{aligned} F^{in}(\mathbf{x}) &= \lambda^{in}\sqrt{F_1^{in}(\mathbf{x})\,F_2^{in}(\mathbf{x})} \\ F^{out}(\mathbf{x}) &= \lambda^{out}\left(1 - \sqrt{(1-F_1^{out}(\mathbf{x}))\,(1-F_2^{out}(\mathbf{x}))}\right), \end{aligned} \tag{7}$$

where

$$F_i^{in}(\mathbf{x}) = k_i^{in}\,\left(I_i(\mathbf{x}) - c_i^{in}\right)^2 \qquad \text{and} \qquad F_i^{out}(\mathbf{x}) = k_i^{out}\,\left(I_i(\mathbf{x}) - c_i^{out}\right)^2,$$

and where $k_{in}^{(i)}$ and $k_{out}^{(i)}$ are normalizing scale factors, chosen so that $F_i^{in} \le 1$ and $F_i^{out} \le 1$, and where c_i^{in} and c_i^{out} are the average intensities of image I_i in regions $\mathrm{int}(C)$ and $\mathrm{ext}(C)$, respectively. The square roots are used to retain a consistent order of magnitude.

For example, consider the pair of images and contour shown in Figure 6 below, where image I_1 is on the left and image I_2 is on the right. At point a, the intensity $I_1(a)$ is close to c_1^{out} but is not close to c_1^{in}, so F_1^{out} is small (≈ 0) and F_1^{in} is large (≈ 1). The intensity $I_2(a)$, on the other hand, is close to c_2^{in}, but not to c_1^{out}, so F_2^{in} is small (≈ 0) and $F_2^{out}(a)$ is large (≈ 1). The value of $F^{in}(a)$ is small because it is based on the logical OR element and $F_2^{in}(a)$ is small. The value of $F^{out}(a)$ is large because it is based on the logical AND element and $F_2^{out}(a)$ is large. Since $F^{out}(a) > F^{in}(a)$, the contour moves outward at point a.

At point b, both intensities $I_i(b)$ are close to c_i^{out} but not to c_i^{in}, so in both cases $F_i^{in}(b)$ is large and $F_i^{out}(b)$ is small. As a consequence $F^{in}(b) > F^{out}(b)$ and the contour moves inward at point b. In either case, the contour moves toward the union of the objects.

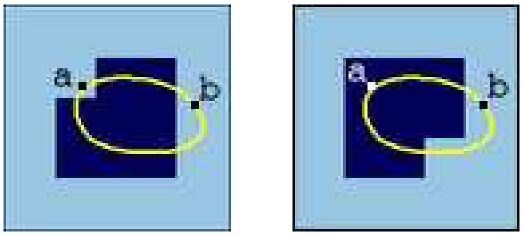


Figure 6: The contour moves outward at a and inward at b. In both cases, toward the union (logical OR) of the objects.

For the logical AND segmentation model we apply the logical AND element to the interior of C and apply the logical OR element to the exterior of C. That is, we take the intersection of the images inside the contour and the union outside:

$$\begin{aligned} F^{in}(\mathbf{x}) &= \lambda^{in}\left(1 - \sqrt{(1-F_1^{in}(\mathbf{x}))(1-F_2^{in}(\mathbf{x}))}\right) \\ F^{out}(\mathbf{x}) &= \lambda^{out}\sqrt{F_1^{out}(\mathbf{x})\,F_2^{out}(\mathbf{x})}\,, \end{aligned} \tag{8}$$

where F_i^{in} and F_i^{out} are defined as above. In this case, we are simply reversing the roles of the regions $\text{int}(C)$ and $\text{ext}(C)$, as compared to the logical OR model. Applying the logical AND model to the images and contour in Figure 6, we see that $F^{in}(a) > F^{out}(a)$ and $F^{in}(b) > F^{out}(b)$, so the contour moves inward at both points a and b. That is, toward the logical AND of the objects.

3 JOINT SEGMENTATION AND REGISTRATION ALGORITHM

3.1 Description of Model

Let $I_1 : \Omega \to R^+$ and $I_2 : \Omega \to R^+$ denote two possibly unregistered images, where each contains a common object of interest. Let C_1 and C_2 denote the contours which are to segment the object in images I_1 and I_2, respectively. The key idea of Yezzi, Zöllei, and Kapur is to relate these two contours through a parameterized mapping g. In their approach, $C_2 = g(C_1; p)$, where p are the parameters for g. We use the same basic approach in our algorithm; however, we consider C_1 and C_2 to be the images of a contour C under g for two different sets of registration parameters p_1 and p_2. That is,

$$C_1 = g(C;p_1) \quad \text{and} \quad C_2 = g(C;p_2)\,. \tag{9}$$

The use of two transformations gives the algorithm more flexibility in avoiding local registration minima.

The goal of the algorithm is to select a contour C and parameters p_1 and p_2 that minimizes the energy

$$E(C;p_1,p_2) = E^{in}(C;p_1,p_2) + E^{out}(C;p_1,p_2) + \mu\,\text{length}(C), \tag{10}$$

where

$$\begin{aligned} E^{in}(C;p_1,p_2) &= \int_{\text{int}(C)} F^{in}(\mathbf{x};p_1,p_2)\,d\mathbf{x} \\ E^{out}(C;p_1,p_2) &= \int_{\text{ext}(C)} F^{out}(\mathbf{x};p_1,p_2)\,d\mathbf{x}, \end{aligned} \tag{11}$$

and where the expressions for $F^{in}(\mathbf{x};p_1,p_2)$ and $F^{out}(\mathbf{x};p_1,p_2)$ depend upon which segmentation model is chosen.

3.2 Segmentation Models

For segmentation, we write the equations for the multi-channel Chan-Vese algorithm (5) and for the logic models (7) and (8) in this new setting. In each case, we replace $I_i(\mathbf{x})$ with $I_i(g(\mathbf{x};\, p_i))$.

For the multi-channel Chan-Vese algorithm (where N = 2), we have

$$F^{in}(\mathbf{x};\, p_1, p_2) = \frac{1}{2}\sum_{i=1}^{2} F_i^{in}(\mathbf{x};\, p_i) \qquad \text{and} \qquad F^{out}(\mathbf{x};\, p_1, p_2) = \frac{1}{2}\sum_{i=1}^{2} F_i^{out}(\mathbf{x};\, p_i), \tag{12}$$

where

$$F_i^{in}(\mathbf{x};\, p_i) = \lambda_i^{in}\left(I_i(g(\mathbf{x};\, p_i)) - c_i^{in}\right)^2 \qquad \text{and} \qquad F_i^{out}(\mathbf{x};\, p_i) = \lambda_i^{out}\left(I_i(g(\mathbf{x};\, p_i)) - c_i^{out}\right)^2.$$

For the logical OR model, we have

$$\begin{aligned} F^{in}(\mathbf{x};\, p_1, p_2) &= \lambda^{in}\sqrt{F_1^{in}(\mathbf{x};\, p_1)\, F_2^{in}(\mathbf{x};\, p_2)} \\ F^{out}(\mathbf{x};\, p_1, p_2) &= \lambda^{out}\left(1 - \sqrt{(1 - F_1^{out}(\mathbf{x};\, p_1))(1 - F_2^{out}(\mathbf{x};\, p_2))}\right), \end{aligned} \tag{13}$$

where

$$F_i^{in}(\mathbf{x};\, p_i) = k_i^{in}\left(I_i(g(\mathbf{x};\, p_i)) - c_i^{in}\right)^2 \quad \text{and} \quad F_i^{out}(\mathbf{x};\, p_i) = k_i^{out}\left(I_i(g(\mathbf{x};\, p_i)) - c_i^{out}\right)^2. \tag{14}$$

For the logical AND model, we have

$$\begin{aligned} F^{in}(\mathbf{x};\, p_1, p_2) &= \lambda^{in}\left(1 - \sqrt{(1 - F_1^{in}(\mathbf{x};\, p_1))(1 - F_2^{in}(\mathbf{x};\, p_2))}\right) \\ F^{out}(\mathbf{x};\, p_1, p_2) &= \lambda^{out}\sqrt{F_1^{out}(\mathbf{x};\, p_1)\, F_2^{out}(\mathbf{x};\, p_2)}, \end{aligned} \tag{15}$$

where $F_i^{in}(\mathbf{x};\, p_i)$ and $F_i^{out}(\mathbf{x};\, p_i)$ are defined in equation (14).

3.3 Registration Model

There are many valid choices for the mapping g. For simplicity, we use the Euclidean transformation

$$g(\mathbf{x}) = MR\,\mathbf{x} + T, \tag{16}$$

where

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

The parameters of g are $p = (\Delta x, \Delta y, \Delta\theta, m)$, where Δx and Δy are the translation, $\Delta\theta$ is the rotation about the center of C, and m is the magnification. When $p = (0, 0, 0, 1)$, the transformation $g(\,\cdot\,; p)$ is the identity map.

3.4 Description of Algorithm

The energy (10), which depends upon the segmentation contour C and upon the registration parameters p_1 and p_2, is minimized by "interleaving" segmentation and registration, as proposed by Yezzi, Zöllei, and Kapur. During each iteration of the algorithm, we alternately improve the estimate of C and of p_1 and p_2.

We have found that starting the cycle with registration seems to reduce the sensitivity of the algorithm to local minima. During each iteration, we first hold C fixed and improve the estimates of the registration parameters. In our implementation, we use a line search. The direction of the search is found by taking numerical derivatives of the energy $E^{in}(C;\, p_1, p_2)$ with respect to the components of p_1 and p_2. For example, the components of p_1 are updated by numerically integrating

$$\begin{aligned} \partial \Delta x_1 \,/\, \partial t &= -\partial E^{in}(C;\, p_1, p_2) \,/\, \partial \Delta x_1 \\ \partial \Delta y_1 \,/\, \partial t &= -\partial E^{in}(C;\, p_1, p_2) \,/\, \partial \Delta y_1 \\ \partial \Delta q_1 \,/\, \partial t &= -\partial E^{in}(C;\, p_1, p_2) \,/\, \partial \Delta q_1 \\ \partial m_1 \,/\, \partial t &= -\partial E^{in}(C;\, p_1, p_2) \,/\, \partial m_1, \end{aligned} \tag{17}$$

where the partial derivatives are approximated by finite difference derivatives. Since p_1 and p_2 are independent, it is useful to update each set of parameters separately. Also, since translation, rotation, and magnification have different scales, it is useful to have different time steps for these components.

Next we hold p_1 and p_2 fixed and improve the estimate of C by evolving it along the gradient flow (3), which in this context, is given by

$$\frac{\partial C(\mathbf{x})}{\partial t} = (\, F^{in}(\mathbf{x};\, p_1, p_2) \;-\; F^{out}(\mathbf{x};\, p_1, p_2) \;+\; \mu\, \kappa \,)\, \mathbf{N}. \tag{18}$$

We continue this process of alternately improving the segmentation and the registration until the algorithm converges.

3.5 Initial Registration Phase

The line search proposed above and the method proposed by Yezzi, Zöllei, and Kapur, use only local information about the energy to update the registration parameters. These local methods work well in a neighborhood of the optimal value, but can fail for larger errors. Figure 7 below illustrates this situation. In the figure, image I_1 and its segmenting contour C_1 appear on the left and image I_2 and its contour C_2 are on the right. The errors in the registration parameters p_1 are not zero, but are small enough for the line search to find the optimal value. The errors in p_2, although not much larger, are too big for local methods to work. Small changes in p_2 do not affect the energy. For images with more content, the parameters can be updated incorrectly, even causing the contour C_2 to move away from the object of interest.

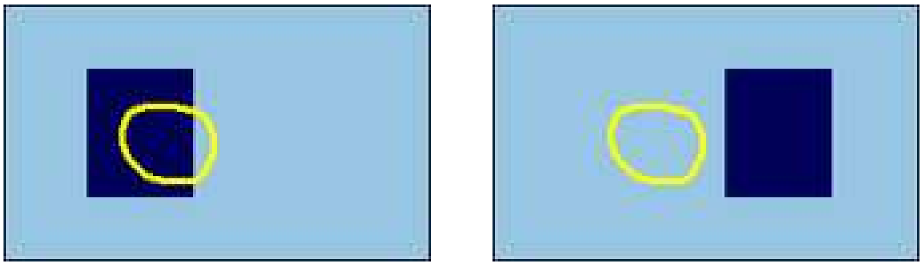


Figure 7: Registration fails due to large error in the registration parameters p_2 in image I_2 (right).

Since poor estimates of at least one set of registration parameters are likely, we add an initial phase to the registration process. The purpose of the initial phase is to bring the registration parameters into a neighborhood of the optimal value, where local methods are effective. In Figure 7 above, the initial phase drives contour C_2 to the object in image I_2. The initial phase can be applied to either, or both images. The method requires an estimate of the average intensity of the object(s) of interest. For sake of the discussion, we assume the scenerio illustrated in Figure 7: that the initial guess of p_1 is reasonable, but that the error in the initial guess for p_2 can be large. In this case, we can often use the value of c_1^{in} as the estimate of c_2^{in}.

We use the estimate $\hat{c}_2^{in}$ of the average intensity of the object of interest in image I_2 to construct the initial registration phase energy

$$E_\psi(C_2) \;=\; \frac{1}{m}\int_{C_2} \psi^m(\mathbf{x})\,ds, \tag{19}$$

where $\psi(\mathbf{x})$ is the distance from $\mathbf{x}$ to the object of interest in I_2. The value of m can be taken as either 1 or 2. A value of $m = 1$ usually gives a faster convergence, although using $m = 2$ gives better behavior near the minima. The function ψ is the solution of

$$\psi_t \;=\; \operatorname{sign}(\psi)\,(\,1 \;-\; |\nabla\psi|\,) \tag{20}$$

$$\psi_0 \;=\; \begin{cases} 0, & \text{when } \left|\, I_2(\mathbf{x}) \;-\; \hat{c}_2^{in} \,\right| \;<\; \epsilon \\ 1, & \text{otherwise,} \end{cases}$$

where ϵ is a small value, related to the uncertainty in the estimate of c_2^{in}. Figure 8 shows an image (left) and the corresponding distance function (right), where $\hat{c}_2^{in}$ is the average intensity of the object.

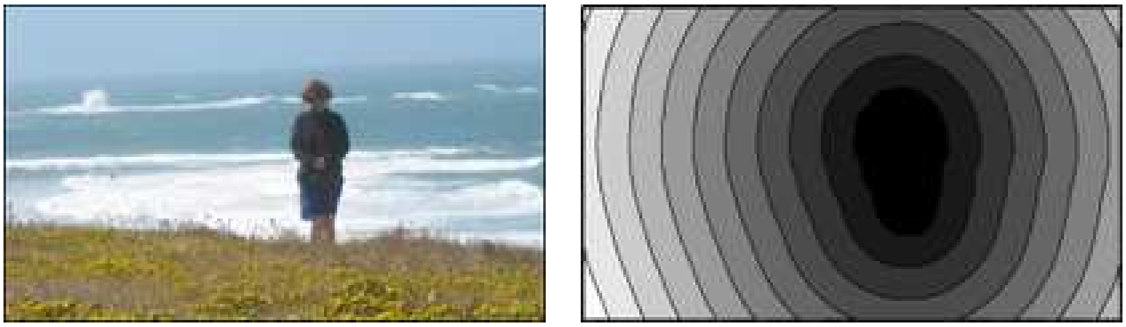


Figure 8: An image (left) and a shaded contour plot of the corresponding ψ function (right). Shading represents the function values. Black corresponds to 0 and white corresponds to the maximum value.

The energy E_ψ is minimized when the average distance of the contour to the object is minimized. This occurs when the contour C_2 is either within, or centered about, the object.

We use a gradient descent to minimize the energy E_ψ. During this process, only the values of Δx and Δy are updated, as we are only interested in bringing the contour C_2 into contact with the object of interest. The equations for the gradient descent are give by

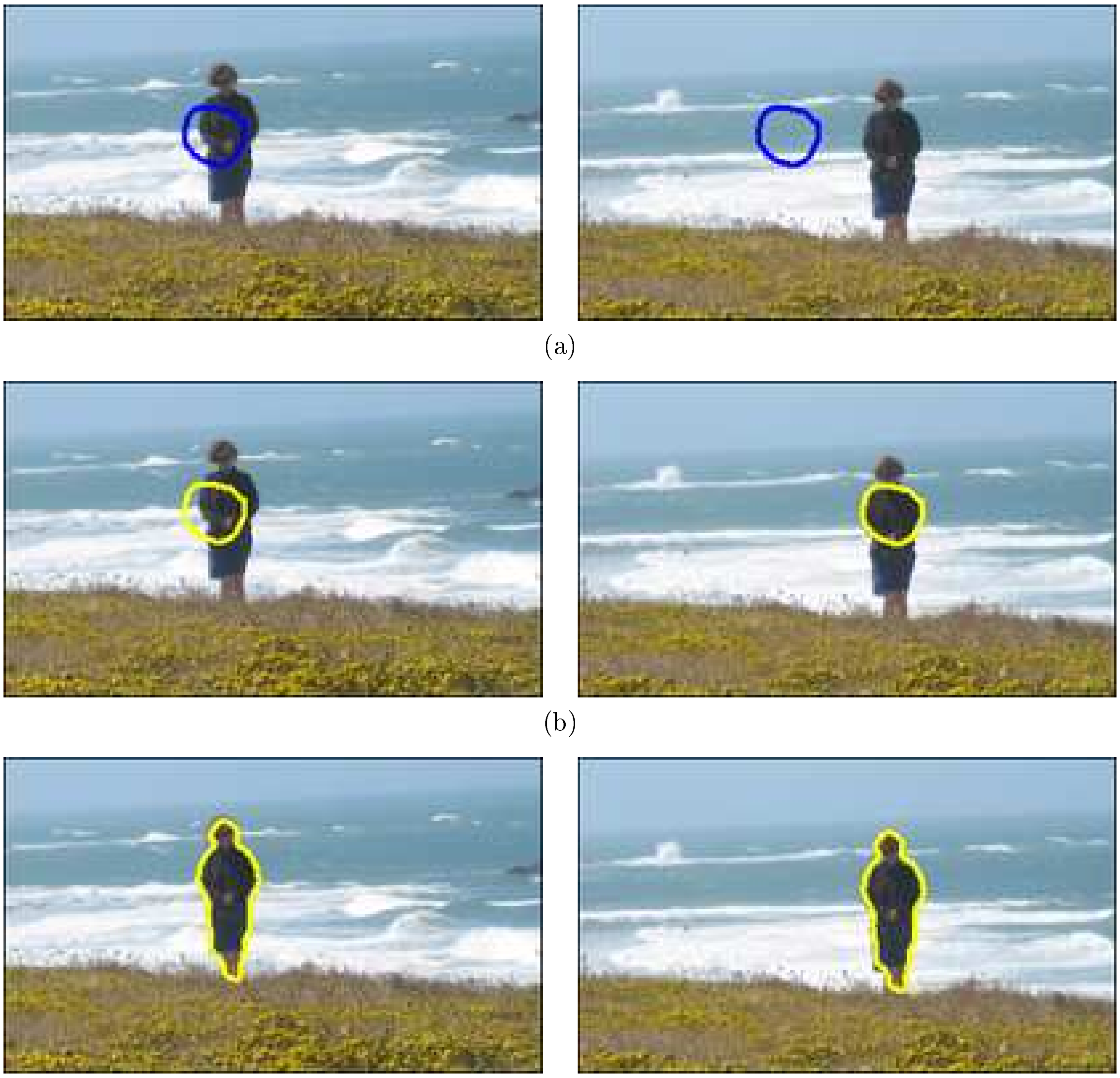
$$\begin{aligned} \frac{\partial \Delta x}{dt} &= -\frac{\partial E_\psi}{\partial x} = -\int_{C_2} \psi^{m-1} \frac{\partial \psi(\mathbf{x})}{\partial x}\, ds \\ \frac{\partial \Delta y}{dt} &= -\frac{\partial E_\psi}{\partial y} = -\int_{C_2} \psi^{m-1} \frac{\partial \psi(\mathbf{x})}{\partial y}\, ds\,. \end{aligned} \tag{21}$$

There are many ways to implement the initial phase. For example, one can segment image I_1 using the scalar Chan-Vese algorithm [7] until a good estimate of c_1^{in} is obtained, then use this as the value of $\hat{c}_2^{in}$. In difficult cases, one can periodically recalculate the distance function ψ, but in our examples this was not necessary.

4 SAMPLE RESULTS

For the examples shown in this section, image I_1 and its segmentation contour C_1 are shown on the left and image I_2 along with C_2 are shown on the right. During the initial registration phase, the contours are shown in blue, and in yellow afterwards. In our code, we assume that the initial estimate of the registration parameters p_1 are reasonable and apply the initial registration phase to image I_2, were we use the value of c_1^{in} as the estimate $\hat{c}_2^{in}$.

Figure 9 below shows the normal behavior of the algorithm. The algorithm is initially in the initial registration phase, as shown in Figure 9(a). During the initial phase, C_2 is driven toward the object of interest, as shown in Figure 9(b). The algorithm then jointly segments and registers the images until it converges, as shown in Figure 9(c).



(a)

(b)

(c)

Figure 9: Usual behavoir of the algorithm. The algorithm begins in the initial registration phase (a). After the contour C_2 is driven to the object of interest, the algorithm transitions out of the initial phase (b). The algorithm then jointly segments and registers the images until it converges (c).

Figure 10 illustrates the need for an initial registration phase. The initial phase was not used in this example. Initially contour C_1 is in contact with the object of interest, but C_2 is not. Note that a horizontal translation of C_2 toward the object does not change the energy (10). Due to image content, the contour moves upward, toward a region with a darker background, since this decreases the energy. The initial contour is given in Figure 10(a) and the resulting segmentation is given in Figure 10(b). Note that this scenario is similar to the case illustrated by Figure 7.

(a)

(b)

Figure 10: Without an initial registration phase, contour C_2 (on right) never finds the object of interest. Initially the contour C_2 is not in contact with the object (a), and moves upward (b) in an attempt to locally reduce the energy.

Figure 11 illustrates the need two g transformations. In this case, only one transformation g is used, that is, $C_2 = g(C_1)$, as proposed by Yezzi, Zöllei, and Kapur [23]. Since contour C_2 is correctly placed, there is little change in the registration parameters. The contour C_1, however, is not correctly placed and the segmentation gets trapped in a segmentation local minimum. This behavior is similar to the case illustrated in Figure 5. The initial contour is shown in Figure 11(a). The resulting segmentation for the multi-channel Chan-Vese algorithm and the logical AND model is given in Figure 11(b) and Figure 11(c), respectively.

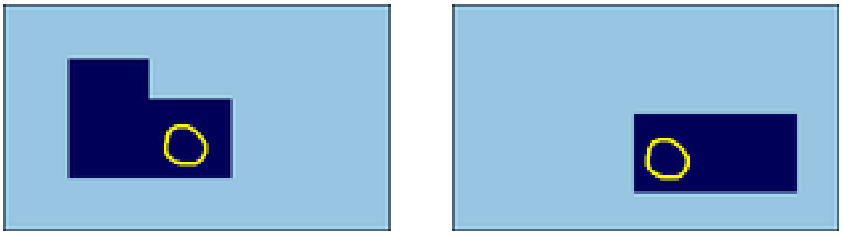
(a)

(b)

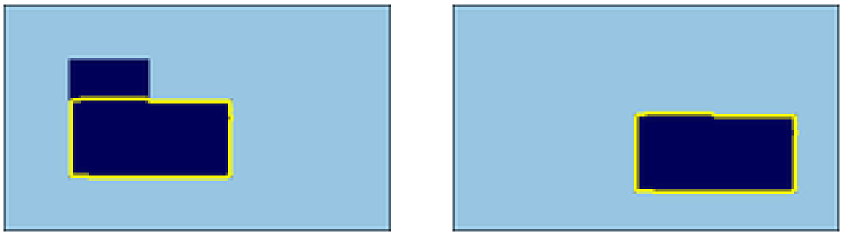
(c)

Figure 11: Without two g transformations, the algorithm can get trapped in a local minimum. Since only contour C_2 (right) can be moved by the registration algorithm and is correctly placed, the registration never gets updated. The segmentation converges to a local minimum. The initial contour is given in (a). The resulting segmentatin for the multi-channel Chan-Vese algorithm and logical AND model are given in (b) and (c), respectively.

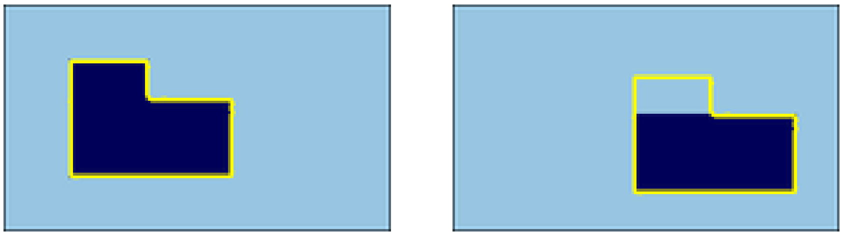
Figure 12 illustrates the behavior of the algorithm when applied to dissimilar objects. We compare all three multi-channel segmentation algorithms discussed in this paper. The initial contour is given in Figure 12(a). The result of applying the multi-channel Chan-Vese algorithm is given in Figure 12(b). As expected, the algorithm finds the correct registration and the segmentation contour is the intersection (logical AND) of the objects. If the initial contour had contained both objects, then the algorithm would have converged to the union (logical OR) of the objects. Figures 12(c) and 12(d) show the results of applying the logical OR and logical AND models, respectively.



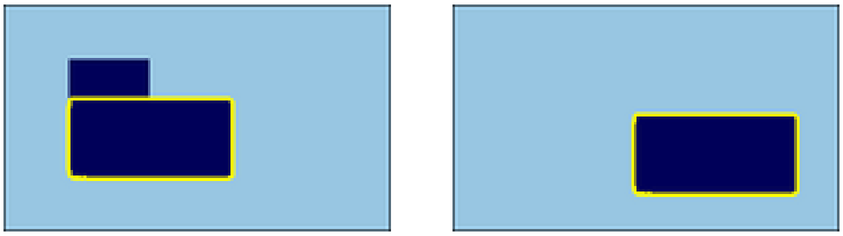
(a): Initial contour



(b): Multi-channel Chan-Vese algorithm - final contour



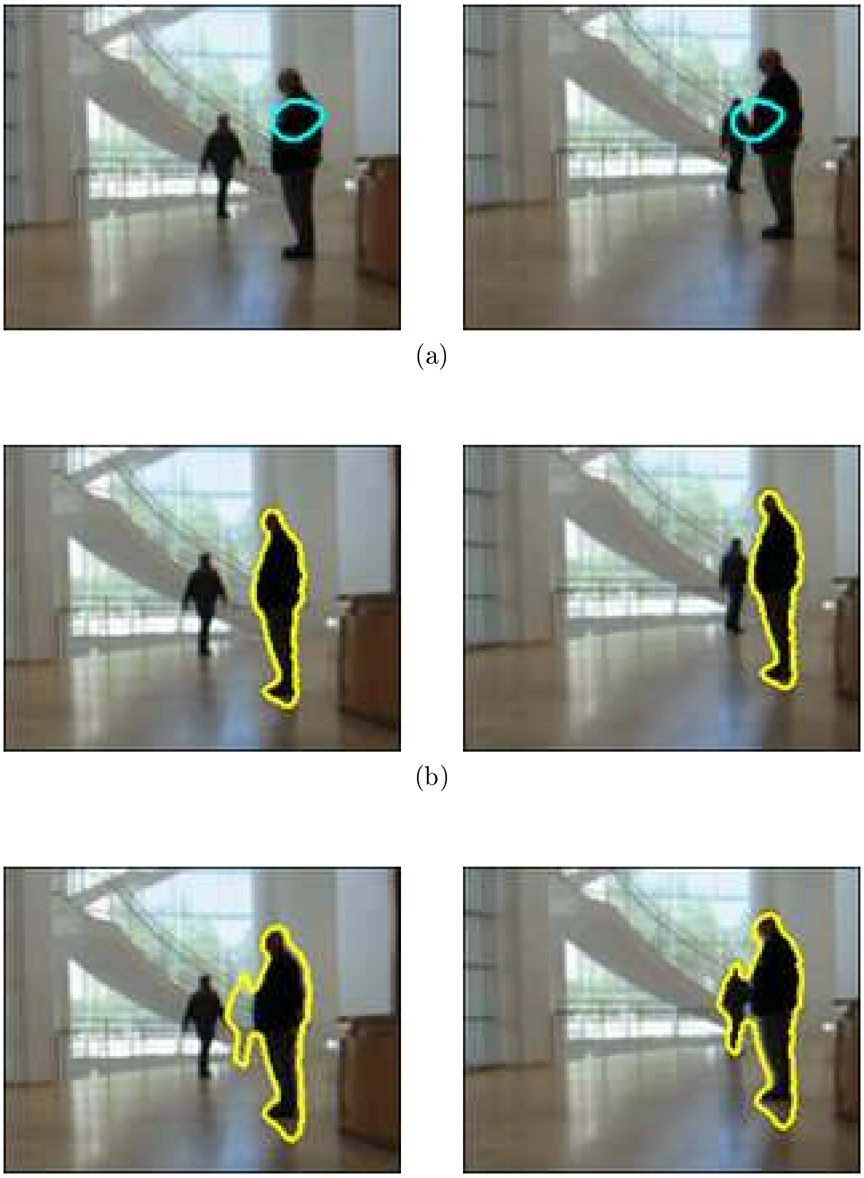
(c): Logic OR model - final contour



(d): Logic AND model - final contour

Figure 12: Behavior of different segmentation models when applied to dissimilar objects. The original contour is given in (a), result of multi-channel Chan-Vese algorithm is given in (b), result of logic OR model is given in (c), and result of logic AND model is given in (d).

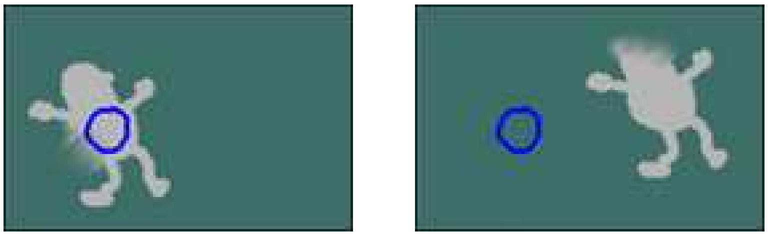
Figure 13 illustrates the usefulness of the logical AND model. Since the two people on the right are of similar intensity, they appear to be a single object. The initial contour is given in Figure 13(a). The result of applying the logical AND model is given in Figure 13(b). We also show the result of applying the logical OR model in Figure 13(c).



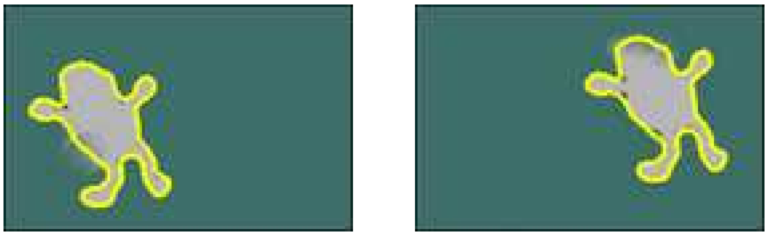
(a) (b) (c)

Figure 13: Usefulness of logical AND model. The two people of the right appear as a single object. The initial contour is given in (a), the result of applying the logical AND model is given in (b). The result of the logical OR model is given in (c).

Figure 14 illustrates the usefulness of the logical OR model. Both of the objects have parts missing which are compensated by the other image. The logical OR model is able to reconstruct the object. The initial contour is given in Figure 14(a) and the resulting segmentation is given in Figure 14(b).



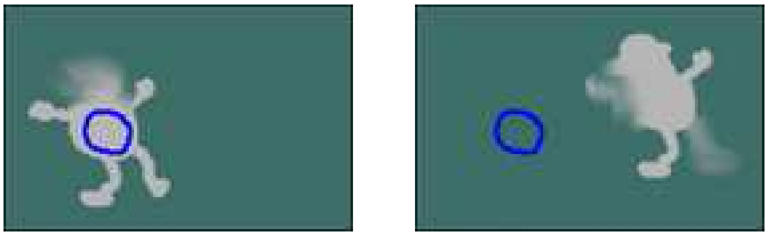
(a)



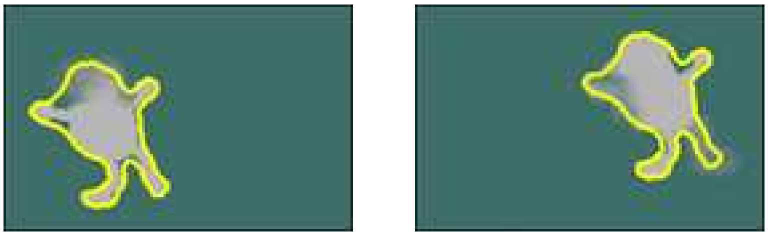
(b)

Figure 14: Usefulness of logical OR model. Each image has missing parts which are compensated by the other object. The initial contour is given in (a) and the resulting segmentation is given in (b).

Figure 15 is similar to Figure 14, but more extreme. Note that, given the data, the algorithm is able to make a good reconstruction of the object. The initial contour is given in Figure 15(a) and the resulting segmentation is given in Figure 15(b).



(a)



(b)

Figure 15: A more extreme example to illustrate the usefulness of the logical OR model. Given the data, the algorithm makes a reasonable reconstruction of the object. The initial contour is given in (a) and the resulting segmentation is given in (b).

5 CONCLUDING REMARKS

We improved the joint segmentation and registration algorithm of Yezzi, Zöllei, and Kapur by using logic models for segmentation, by using two registration transformations, and by using an initial registration phase. All three changes add robustness and fidelity to the algorithm. The introduction of the logic models of Sandberg and Chan is the most significant change because the logic models allow for better control of the segmentation and provide a richer context in which to interpret dissimilar objects. This algorithm also significantly enlarges the scope of applications for the logic models, because it relaxes the requirement that the images be registered.

REFERENCES

[1] O. Amadieu, E. Debreuve, M. Barlaud, and G. Aubert. Inward and outward curve evolution using level set method. In *Int. Conf. on Image Processing*, Kobe, Japan, 1999.

[2] G. Aubert and P. Kornprobst. *Mathematical problems in image processing*, Springer, New York, 2002.

[3] M. Bazaraa, H. Sherali, and C. Shetty. *Nonlinear Programming*, John Wiley and Sons, New York, 1993.

[4] A. Blake and M. Isard. *Active Contours* Springer, New York, 2000.

[5] A. Blake and A. Yuille (eds.). *Active Vision*, MIT Press, Cambridge, 1992.

[6] T. Chan, B. Sandberg, and L. Vese. Active contours without edges for vector-valued images. *J. of Visual Comm. and Image Rep.*, 11:130-141,1999.

[7] T. Chan and L. Vese. An active contour model without edges. *Int. Conf. Scale-Space Theories in Computer Vision*, 16(2):266-277, 1999.

[8] L. Evans and R. Gariepy. *Measure Theory and Fine Properties of Functions*, CRC Press, 1992.

[9] P. Gill, W. Murray, and M. Wright. *Practical Optimization*, Academic Press, Berkeley, 1981.

[10] M. Kass, A. Witkin, and D. Terzopoulos. *Snakes: Active contour models*, International Journal of Computer Vison, No.1, 1988.

[11] R. LeVeque. *Numerical Methods for Conservation Laws*, Birkhäuser, 1992.

[12] J. Morel and S. Solimini. *Variational methods in image segmentation*, Birkhäuser, 1995.

[13] J. Munkres. *Analysis on Manifolds*, Addison Wesley, 1991.

[14] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems, *Comm. Pure Appl. Math.* 42, 1989.

[15] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, Springer, 2003.

[16] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulation, *Journal of Computational Physics*, 79, 1988.

[17] B. Sandberg and T. Chan. *Logic Operators for Active Contours on Multi-Channel Images*, UCLA CAM Report 02-12, March 2002.

[18] G. Sapiro. *Geometric partial differential equations and image analysis*, Birkhäuser, 1995.

[19] J. Sethian. *Level set methods and fast marching methods*, Cambridge University Press, 1999.

[20] M. Spivak. *Calculus on Manifolds*, Benjamin, New York, 1965.

[21] S. Soatto and A. Yezzi. *Deformation: Deforming Motion, Shape Average and the Joint Registration and Segmentation of images.* UCLA Cam Report 01-32, 2001.

[22] C. Xu and J. Prince. *Snakes, Shapes and Gradient Vector Flow*, IEEE Transactions on Image Processing, Vol. 7, No. 3, 1998.

[23] A. Yezzi, L. Zöllei, and T. Kapur. A variational approach to joint segmentation and registration. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, 2001.