UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Logic Operators for Active Contours on Multi-Channel Images

Berta Sandberg
Tony F. Chan

March 2002

CAM Report 02-12

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

http://www.math.ucla.edu/applied/cam/index.html

Logic Operators for Active Contours on Multi–Channel Images *

Berta Sandberg† Tony F. Chan ‡§

May 28, 2002

Abstract

We propose a mathematical framework for active contours object detection in multi–channel images using logic operations to combine object information for the different channels. We consider methods which use one initial contour that would evolve from the information given in each channel simultaneously. Differences between boundary based and region based models are discussed, and specific models are derived for the geodesic[10] and the Chan-Vese[4] models respectively. Numerical experiments show that the methods were able to find general intersections, unions, and complements of the boundaries and the regions of objects respectively.

1 Introduction

Much has been written on active contour and segmentation of multi–channel images. There are papers that discuss methods for color images [10],[14], texture images convolved with filters [3],[12], multispectral images with occlusion in some channels and noise in others [2], and movie sequences [6]. Many of these models attempt to extract parts of an object from each of the channels and to recombine this information in a logical fashion. In most of these cases, the right segmentation is some combination of occluded objects, or a combination of noisy images.

An example of occluded channels is given in Figure 1. Most models for multi–channel segmentation would find a triangle that is the union of both channels as the "correct" segmentation.

The union is just one of several possible logical operations for multi–channel segmentation. There are other possible segmentations like intersection or difference. For example, a sequence of the brain images of the same subject taken

*This work was supported in part by ONR Contract N00014-96-1-0277 and NSF Contract DMS-9973341.

†bsand@math.ucla.edu.

‡chan@math.ucla.edu.

§Address: UCLA, Mathematics Departement, 405 Hilgard Avenue, Los Angeles, California 90095-1555.

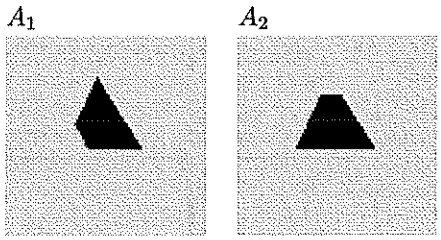


Figure 1: A synthetic example of an object (a triangle) in two different channels. In A_1, the lower left corner is missing. In A_2 the upper corner is missing.

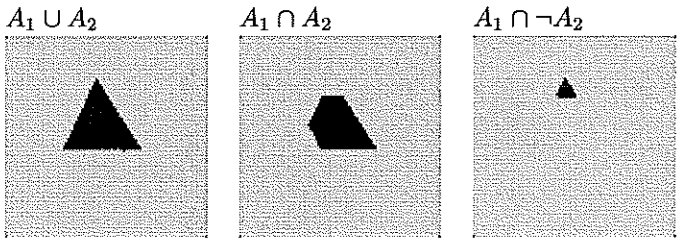


Figure 2: Different logical combinations for the sample image, $A_1 \cup A_2$ is the union of the objects in each channel, $A_1 \cap A_2$ is the intersection, $A_1 \cap \neg A_2$ the object in A_1 that is not in A_2.

over time, in order to locate possible tumor growths. These segmentations can be described using logic terminology such as intersection, union, and negation of the objects in the images. The derived segmentations are illustrated in Figure 2.

The most direct and obvious way of accomplishing logic segmentations is to segment each channel independently, by the model of your choice, followed by bitwise logic operations. However, there are some drawbacks to this approach. First, if there are many images, it is cumbersome and costly to perform all the segmentations separately. On a deeper level, such an approach often gives undesirable segmentations because the information in the separate channels is not taken together. Therefore by assuming that each image is independent of the other, each channel is segmented separately, causing valuable information may to be lost. In Figure 3, such a situation is illustrated. In the two channels, the images (a triangle) are occluded and noisy. When each one is segmented separately, due to the noise, the segmentation has jagged boundaries. We then do bitwise logic operations $A_1 \cap A_2$, $A_1 \cup A_2$ and $A_1 \cap \neg A_2$. While the first two segmentations are done fairly well, the last one involving negation gives false points because of the jagged boundaries.

In this paper, we would like to investigate the possibility of object detection using a single contour for all the channels in the image set using active contour models. Many active contour models for scalar images have the following

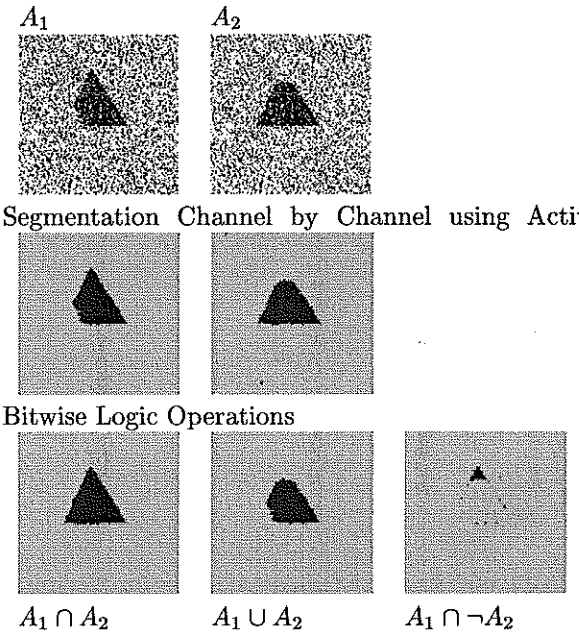


Figure 3: In this example the segmentation is done independently on each channel. Using the active contours without edges model, the two images are combined using bitwise OR and AND. In $A_1 \cap \neg A_2$ spurious points arise as a result of noise in the channels.

variational form:

$$\inf F(u_0, C) = \lambda \int_C J + R,$$

where u_0 is the image, C is the evolving curve, $\int_C J$ is the term related to the image and R is the regularization term. Two examples of these models are the geodesic model[1] and active contours without edges[4]. We will concentrate on the term $\int_C J$. The derivation of the models is based on extending the scalar models to the form of multi–channel models. Implementation of the logic operations is achieved by various combinations of the channels.

We will consider two types of active contour models. Boundary based active contour models such as geodesic and snakes [1, 7] and region based active contours such as active contours without edges[4] and the Dariche-Paragios model[9] will be considered. As specific examples, we will first consider the geodesic model due to Caselles, Kimmel, and Sapiro [1] for the boundary based contours and the active contours without edges for the region based method respectively.

The focus of the papers is on the models. No details of the numerical implementation will be given. These are standard and can be found in [1], [4], for example.

The outline of the paper is as follows: in section 2 we review the boundadry based model, develop the logic model for it, and discuss some problems that arise. In section 3, we will investigate region based models, specifically the active contours without edges model. In Section 4, we compare the logic model of active contours without edges to the vector valued model in [2]. Section 5 is the conclusion which summarizes the work and discusses possible further work on the topic.

2 Logic Operations on a Boundary Based Active Contour Model

2.1 Model

We will start with the following boundary based active contours model[1]:

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

This functional is transformed to a geodesic computation which minimizes the energy:

$$\int_0^1 g(|\nabla I|)|C'|ds$$

where g(s) is an edge detection function (positive decreasing and goes to 0 as $s \to \infty$). An example of such a function is:

$$g(s) = 1/(1 + s^2).$$

Truth Table for 2 Channels				
z_1	z_2	$A_1 \cup A_2$	$A_1 \cap A_2$	$A_1 \cap \neg A_2$
1	1	1	1	1
1	0	0	1	1
0	1	0	1	0
0	0	0	0	1

Table 1: Truth Table for Geodesic model.

We will extend the model from one image u_0 to n images u_0^i, where $i = 1..n$ by changing the functional's boundary detection term to:

$$\int_0^1 f(g(u_0^1), g(u_0^2), ..g(u_0^n))ds. \tag{2}$$

We want the function f to be close to 0 when the curve C is on the prescribed boundary and far from 0 when it's not on the proper boundary. In this model the initial contour surrounds the object and stops once the contour is on the boundary of the detected object. To simplify, we first look at a two channel system.

In keeping with the logic operations, we want to define a truth table. We define an edge indicator logical variable z_i as follows:

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

For the geodesic model described in (1), we can take z to be:

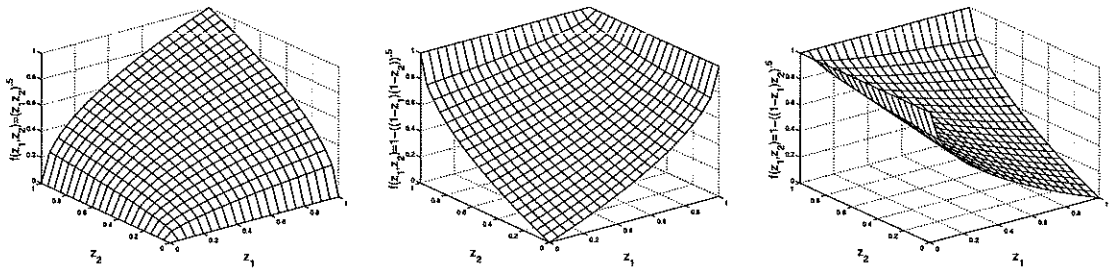
$$z_i = \frac{g(\nabla(u_0^i))}{g(0)}.$$

Notice that $g(0)$ scales z_i so that $0 \leq z_i \leq 1$. Now we combine the z_is to implement the logic operations in the variational model. For the three logic operations done in Figure 2, the desired truth table combination should be:

For the union, $A_1 \cup A_2$, we want the contour C to stop on the boundary in either channel, so if z_i is 0 in either channel, we want the output to be 0. The only time we want the output to be 1 in the truth table is if the contour is not on the boundary in either channel. For the intersection $A_1 \cap A_2$, the truth value should be 0 only if the contour is on the boundary in both of the channels. For the logic operation $A_1 \cap \neg A_2$, we need to take the complement of one of the channels. This can be accomplished by defining:

$$z_2^{'} = 1 - z_2.$$

Now we find a smooth function $f(z_1, z_2)$ which will interpolate the truth table values above. For example, the following functions that give the desired



$f_{A_1 \cup A_2}$ $f_{A_1 \cap A_2}$ $f_{A_1 \cap \neg A_2}$

Figure 4: Plots for Eqns. 3- 5. These figures show that the functions are smooth, continuous interpolations of the truth table with the minimum occuring at $f = 0$.

interpolations:

$$f_{A_1 \cup A_2} = \sqrt{z_1 z_2}, \tag{3}$$

$$f_{A_1 \cap A_2} = 1 - \sqrt{(1 - z_1)(1 - z_2)}, \tag{4}$$

$$f_{A_1 \cap \neg A_2} = 1 - \sqrt{(1 - z_1)(1 - z_2')}. \tag{5}$$

We plot the functions in Figure 4 to show that they have smooth interpolations between the values of the truth table. The z variables are between 0 and 1.

Next, logic operations which combine unions and intersections are considered. A simple example is $A_1 \cup A_2 \cap A_3$. This is calculated from left to right, rewriting it as $(A_1 \cup A_2) \cap A_3$. The corresponding functional:

$$f_{(A_1 \cup A_2 \cap A_3)} = 1 - \sqrt{(1 - \sqrt{z_1 z_2})(1 - z_3)}.$$

More general logical operations can be similarly derived.

These functions can be extend to n–channels and more general logic operations as follows. Consider the unions of objects or their complements in the different channels. Denote the desired logic operations by:

$$L_1(A_1) \cup L_2(A_2) ... \cup L_n(A_n),$$

where $L_i(A_i)$ could be either A_i or $\neg A_i$. The appropriate function for this is:

$$f = (\prod_{i=1}^{n} l_i(z_i))^{\frac{1}{n}},$$

where

$$l_i(z_i) = \begin{cases} z_i & \text{if } L_i(A_i) = A_i \\ z_i' & \text{if } L_i(A_i) = \neg A_i. \end{cases}$$

The $\frac{1}{n}$ term is there to make the value of the function f the same order as the original $g(|\nabla u_0(C(s))|)$. Similarly for the intersection case:

$$L_1(A_1) \cap L_2(A_2) ... \cap L_n(A_n),$$

the appropriate function is:

$$f = 1 - (\prod_{i=1}^{n}(1 - l_i(z_i))^{\frac{1}{n}}.$$

The geodesic functional in (1) can now be extended as follows to implement the desired logical combinations:

$$\begin{aligned} F_{L_1(A_1)\cap..\cap L_n(A_n)} &= \int_0^1 (1 - \prod_{i=1}^{n}(1 - l_i(z_i))^{\frac{1}{n}})|C'|ds, \\ F_{L(A_1)\cup..L_n(A_n)} &= \int_0^1 (\prod_{i=1}^{n} l_i(z_i))^{\frac{1}{n}}|C'|ds. \end{aligned}$$

$$F = \int_0^1 f(z_1, z_2, .., z_n)|C'|ds.$$

For the curve evolution, the level set method in [8] is used. The representation of the evolution of contour C is the function $\phi(t, \cdot)$ at the zero level set. The function evolves over time t with the speed of mean curvature. We calculate the Euler Lagrange formulation for a general function $f(z_1, .., z_n)$. The following partial differential equations give the solution of the above functionals at steady state:

$$\frac{d\phi}{dt} = f(z_1, ...z_n)|\nabla\phi|\kappa + \nabla f(z_1, ..z_n) \cdot \nabla\phi,$$

For the specific f described above the Euler Lagrange are:

$$\frac{d\phi_{L(A_1)\cup..\cup L_n(A_n)}}{dt} = (\prod_{i=1}^{n} l_i(z_i)^{\frac{1}{n}}|\nabla\phi|)\kappa + \nabla(\prod_{i=1}^{n} l_i(z_i))^{\frac{1}{n}} \cdot \nabla\phi.$$

The Euclidean curvature κ, is calculated by the following:

$$\kappa = div(\frac{\nabla\phi}{|\nabla\phi|}).$$

The details of the numerical schemes are standard and can be found in [1].

2.2 Experimental Results

We present some examples using the boundary based active contour models. When these models are implemented on the piecewise constant example in Figure 1, the object for $A_1 \cap A_2$, $A_1 \cup A_2$ and$A_1 \cap \neg A_2$ are found as shown in

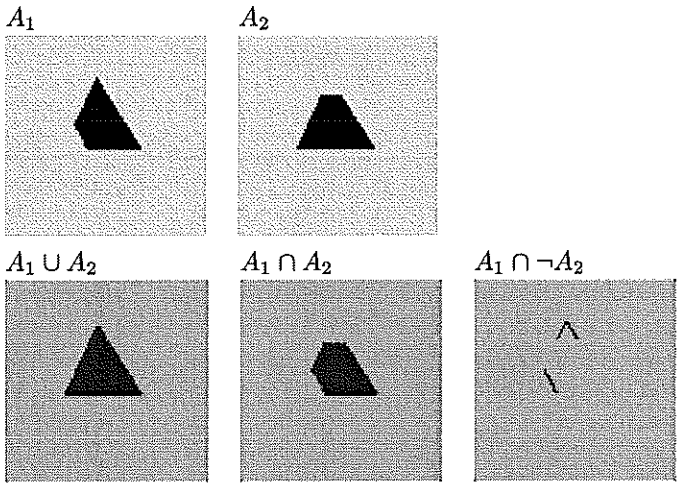


Figure 5: Segmentation using the geodesic model. The objects were found using the functions described in Equations 3- 5. The model finds $A_1 \cup A_2$ and $A_1 \cap A_2$ as expected, however when the segmentation is done for $A_1 \cap \neg A_2$ the intersection of boundaries (correctly) rather than the regions is found.

Figure 5. The union of the objects in two channels is found successfully. Since the contour is from the outside moving in, it stops when it detects the first edge it reaches in either of the images. On the other hand, for the intersection case, the contour reaches an edge but continues inward if the edge is not in both channels. For $A_1 \cap \neg A_2$ the model finds the boundaries that are in the first channel but not second. Thus the model wants to stop on the boundaries of Channel 1, but to minimize the functional excludes the boundary of Channel 2. The model will not work as well when the initial contour is inside rather than outside the object. We give an example of this in Figure 7.

Since this model finds the boundaries rather than the regions there are limitations to this. Previously we had registered images, but if the images aren't registered, unexpected results occur. We show this in Figure 6. The union of the objects is successful, but when we attempt to find the intersection, the contour only finds the points of intersection on the boundary. In analyzing what happened, we look at the function $f_{A_1 \cap A_2} = 1 - \sqrt{(1-z_1)(1-z_2)}$. The minimum occurs when the model finds both of the edges. Since the images are unregistered, the area where both edges intersect is several points. Likewise, when $A_1 \cap \neg A_2$ is segmented, A_1 is found since the model finds all the edges that are not in the same location in the two channels. When noise was added to the channels like the example in Figure 3 the model finds the contour only after the image is de-noised. However, when the image has been de-noised the images become unregistered.

To summarize, in this section we extended the geodesic contours model to multi–channel form by combining the edge detector functions g_i. This was done

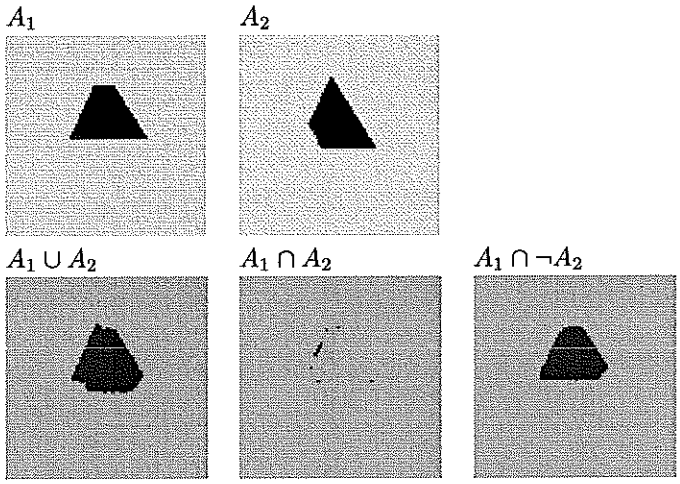


Figure 6: Segmentation of unregistered images. The union of the two images is found for the boundary based method: the initial contour is inside. But once we attempt to find $A_1 \cap A_2$ and $A_1 \cap \neg A_2$, the model attempts to find intersection of boundaries instead, and thus finds only points of intersection for $A_1 \cap A_2$ since at best there are only a couple of boundary points that intersect. In the case of $A_1 \cap \neg A_2$ it finds most of A_1 since most of the boundary points do not intersect.

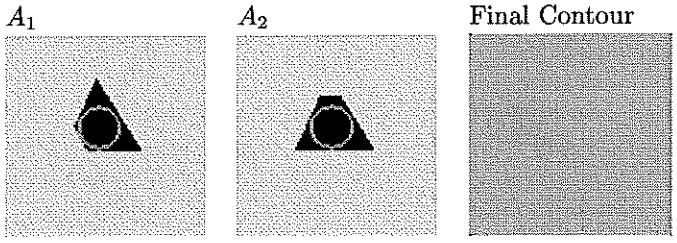


Figure 7: For the boundary based method, the initial contour is inside the object. The contour contracts inside but finds no boundaries, eventually it disappears.

successfully but with some limitations characteristic of boundary based active contours.

3 Logic Operations on Region Based Active Contours

While the boundary based method gives the segmentation as predicted, we would like to see the operations done in a way that is closer to human visual perception. The human perception includes not just boundaries of objects but also the regions and what is inside and outside the region. This is especially important when finding logical combinations of objects, thus we will look at a region based model. As an example of a region based model, we consider the active contour without edges model proposed by Chan and Vese [4]. The scalar model can be written as follows:

$$\inf F(C, c_1, c_2) = \mu(length(C)) + \lambda \int_{inside(C)} |u_0 - c_1|^2 dx + \gamma \int_{outside(C)} |u_0 - c_2|^2 dx.$$

The first term on the right denotes the length of C and is used as a regularization for C. The constants c_1 and c_2 are the average values of the image inside and outside C respectively. The object in the image is defined as the image that is located inside C. The minimum of the functional occurs when the curve C is on the object. This model does not use the edge function as described in Section 2. The model has several advantages: it detects edges both with and without gradient, it can be made to automatically detect interior contours; the initial curve does not necessarily have to start around the objects to be detected and instead can be placed anywhere in the image, and it gives in addition a partition of the image into two regions, one formed by the set of the detected objects, while the second one gives the background.

We want to find an analog for the multi–channel logic operation case that is an extension of F. As in the previous section, we define a parameter z_i for image i that is 0 when the region is enclosed and 1 otherwise:

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

Just as in the previous section we look to the scalar model for the definitions of the parameter z. In this model how well the curve C fits the inside and outside of the object are calculated separately. Thus our parameter should reflect this.

We now redefine our parameters such that the information inside and outside the contour C is calculated separately. We define two separate parameters z_i^{in} and z_i^{out}:

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

Truth Table for 2 Channels							
	z_1^{in}	z_2^{in}	z_1^{out}	z_2^{out}	$A_1 \cup A_2$	$A_1 \cap A_2$	$A_1 \cap \neg A_2$
$(x,y) \in C$	1	1	0	0	1	1	1
	1	0	0	0	0	1	1
	0	1	0	0	0	1	0
	0	0	0	0	0	0	1
$(x,y) \in \Omega \backslash C$	0	0	1	1	1	1	0
	0	0	1	0	1	0	1
	0	0	0	1	1	0	0
	0	0	0	0	0	0	0

Table 2: The Truth Table for the Active Contours without Edges Model

$$z_i^{out}(u_0^i, x, y, C) = \begin{cases} 1 & \text{if } (x,y) \text{ inside}(C), \\ 0 & \text{otherwise.} \end{cases}$$

A natural way to define z_i^{in} and z_i^{out} is as follows:

$$z_i^{in}(u_0^i, x, y, C) = \frac{|u_0^i(x,y) - c_+^i|^2}{\max_{(x,y)\in u_0^i} u_0^i}$$

$$z_i^{out}(u_0^i, x, y, C) = \frac{|u_0^i(x,y) - c_-^i|^2}{\max_{(x,y)\in u_0^i} u_0^i}$$

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

$$\begin{aligned} z_i^{in'} &= 1 - z_i^{in}, \\ z_i^{out'} &= 1 - z_i^{out}. \end{aligned}$$

Now we need a way to incorporate the fit of the contour both inside the object and outside. As an example we will discuss the intersection of the two objects in Figure 2. To find the intersection of an object one can take the intersection of the objects designated as the black values in Figure 8a. In our example this is done by taking $1 - \sqrt{(1-z_1^{in})(1-z_2^{in})}$ as explained in the previous section. But another way to look at it is to look at outside of the objects. If we take the union of the outside of C which is $z_1^{out} z_2^{out}$, we also get the intersection of the objects Figure 8. Adding the two different sections together will give us the intersection of the object from the prospective of inside C, as well as outside C.

The desired truth table can then be described using the z_i^{in}s and z_i^{out}s. We show the table for the 2 channel case.

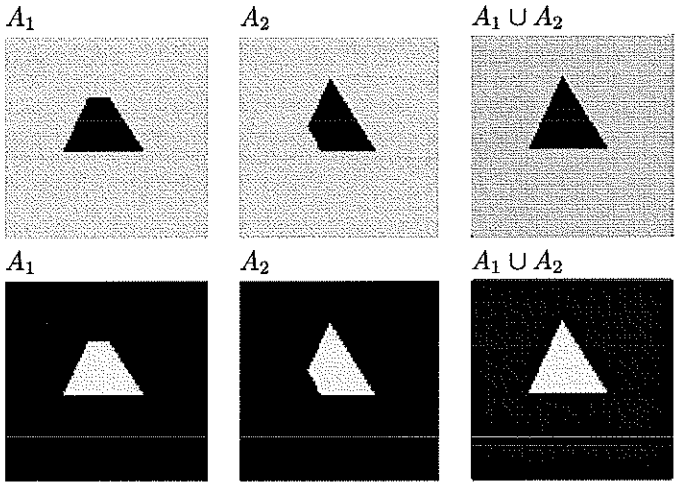


Figure 8: Logic Operations inside and outside the object. The upper triple of images show that the union of the inside (black) region gives the union of the 2 objects in A_1 and A_2. The bottom triple shows that the intersection of the outside (black) region gives the correct complement to the union of two objects.

Now we choose the functions to interpolate the truth table above. There are many functions that could fit for the parameters chosen. Two possibilities are:

$$f_{A_1 \cup A_2}(x,y) = (z_1^{in}(x,y) z_2^{in}(x,y))^{\frac{1}{2}} + (1 - \sqrt{(1 - z_1^{out}(x,y))(1 - z_2^{out}(x,y))}),$$

$$f_{A_1 \cap A_2}(x,y) = 1 - \sqrt{(1 - z_1^{in}(x,y))(1 - z_2^{in}(x,y))} + (z_1^{out}(x,y) z_2^{out}(x,y))^{\frac{1}{2}}).$$

The square roots are taken of the products to keep them of the same order as the original scalar models. An example of a function using the negation is $A_1 \cap \neg A_2 \cup A_3$. The function is found the same way as in Section 2.

$$f_{A_1 \cap \neg A_2 \cup A_3} = 1 - \sqrt{(1 - \sqrt{z_1^{in}(1 - z_2^{in})})(1 - z_3^{in})} + \sqrt{1 - \sqrt{(1 - z_1^{out}) z_2^{out}} z_3^{out}}.$$

It is straightforward to extend the two channel case to n channels. We will use the terminology that was defined in Section 2. First consider the union case. Let the logic operation be expressed as:

$$L_1(A_1) \cup L_2(A_2) ... \cup L_n(A_n),$$

where $L_i(A_i)$ is either A_i or $\neg A_i$. The function for this is as follows:

$$f_{L_1(A_1) \cup .. \cup L_n(A_n)} = 1 - \prod_{i=1}^{n} ((1 - l_i(z_i^{in})))^{\frac{1}{n}} + (\prod_{i=1}^{n} l_i(z_i^{out}))^{\frac{1}{n}},$$

where

$$l_i(z_i^{in}) = \begin{cases} z_i^{in} & \text{if } L_i(A_i) = A_i, \\ z_i^{in'} & \text{if } L_i(A_i) = \neg A_i. \end{cases}$$

Similarly, for the intersection case:

$$L_1(A_1) \cap L_2(A_2) ... \cap L_n(A_n),$$

the function is:

$$f_{L_1(A_1) \cap .. \cap L_n(A_n)} = (\prod_{i=1}^{n} l_i(z_i^{in}))^{\frac{1}{n}} + 1 - (\prod_{i=1}^{n} (1 - l_i(z_i^{out})))^{\frac{1}{n}}.$$

The functionals to be minimized in the model can now be written as:

$$\begin{aligned} F_{L_1(A_1) \cap .. \cap L_n(A_n)} &= (length(C)) + \lambda \Big[\int_{inside(C)} (\prod_{i=1}^{n} l_i(z_i^{in}))^{\frac{1}{n}} dx + \\ & \int_{outside(C)} (1 - (\prod_{i=1}^{n} (1 - l_i(z_i^{out}))^{\frac{1}{n}}) dx \Big], \end{aligned}$$

and

$$\begin{aligned} F_{L_1(A_1) \cup .. \cup L_(A_n)} &= (length(C)) + \lambda \Big[\int_{inside(C)} (1 - (\prod_{i=1}^{n} (1 - l_i(z_i^{in})))^{\frac{1}{n}}) dx \\ & \int_{outside(C)} (\prod_{i=1}^{n} l_i(z_i^{out}))^{\frac{1}{n}} dx \Big]. \end{aligned}$$

The functional may be written using the level set formulation as described in the boundary based method. Thus C is represented by the function ϕ at the zero level set. Inside C, $\phi > 0$ and outside $\phi < 0$. To describe the difference of inside and outside of C we will introduce the Heaviside function H and the Dirac delta function δ.

$$H(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & z < 0, \end{cases}$$

and

$$\delta(z) = \frac{d}{dz} H.$$

Now we can rewrite the functional F for a general $f(z_1^{in}, z_1^{out}, ..)$ using the level set function ϕ. The function f is in the form

$$f(z_1^{in}, z_1^{out}, ..) = f_{in}(z_1^{in}, z_2^{in}, ..) + f_{out}(z_1^{out}, z_2^{out} ..).$$

$$F(\phi, c^+, c^-) = \mu length(\phi = 0) + \int_{\Omega} f_{in}(z_1^{in}, .. z_n^{in}) H(\phi) + f_{out}(z_1^{out}, .., z_n^{out})(1 - H(\phi)) dx. \quad (6)$$

Derivation of the Euler-Lagrange equation are similar to that of the scalar model and yield the following differential equation (which at steady state gives the solution):

$$\frac{\partial \phi}{\partial t} = \delta_(\phi)\Big[\mu div\Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) - \lambda(f_{in}(z_1^{in},...,z_n^{in}) - f_{out}(z_1^{out},..,z_n^{out}))\Big]$$

with the boundary condition:

$$\frac{\delta(\phi)}{|\nabla \phi|}\frac{\partial \phi}{\partial \vec{n}} = 0$$

on $\partial\Omega$, where $\vec{n}$ denotes the unit normal at the boundary of Ω. For example, for the two logic models presented earlier, the corresponding Euler-Lagrange equations are:

$$\frac{\partial \phi_{L_1(A_1)\cup..\cup L_n(A_n)}}{\partial t} = \delta_\epsilon(\phi)\Big[\mu div\Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) - \lambda((\prod_{i=1}^{n} l_i(z_i^{in}))^{\frac{1}{n}} + 1 - (\prod_{i=1}^{n}(1 - l_i(z_i^{out})))^{\frac{1}{n}})\Big],$$

$$\frac{\partial \phi_{L_1(A_1)\cap..\cap L_n(A_n)}}{\partial t} = \delta_\epsilon(\phi)\Big[\mu div\Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) - \lambda(1 - (\prod_{i=1}^{n}(1 - l_i(z_i^{in})))^{\frac{1}{n}} + (\prod_{i=1}^{n} l_i(z_i^{out}))^{\frac{1}{n}}).\Big]$$

Even though the form is complicated, the implementation is very similar to that of the scalar model. The details for this scheme can be found in [4].

3.1 Experimental Results

In this section, we show some examples of the performance of the logical active contours models described in Section 3.

Figure 9 shows two different occlusions of a triangle. The logic operations are done successfully using our new model. We are able to recover the union, intersection and negation of the objects in the channels using the functionals described above. The constant λ is on the order of the $(\max u_0^i(x,y))^2$ except when there is noise in the image, in which case it is smaller so that the noise is not included in the segmentation. The models converged to the minimum quickly for all the cases.

In Figure 10, a 3 channel example is shown. We ran the model for four cases. The first two are $A_1 \cap A_2 \cap A_3$ which have

$$f = (\prod_{i=1}^{3} z_i^{out})^{\frac{1}{3}} + 1 - (\prod_{i=1}^{3}(1 - z_i^{in}))^{\frac{1}{3}},$$

and $A_1 \cup A_2 \cup A_3$ with the corresponding function:

$$f = (\prod_{i=1}^{3} z_i^{in})^{\frac{1}{3}} + 1 - (\prod_{i=1}^{3}(1 - z_i^{out}))^{\frac{1}{3}}.$$

The other two cases have examples with negation and mixing of unions and intersections. They are $A_1 \cap \neg A_2 \cap A_3$ and $A_1 \cup A_2 \cup A_3$. Their functions are

$$f = (z_1^{in}(1 - z_2^{in})z_3^{in})^{\frac{1}{3}} + 1 - ((1 - z_1^{out})z_2^{out}(1 - z_3^{out}))^{\frac{1}{3}}$$

and

$$f = (z_1^{in} z_2^{in} z_3^{in})^{\frac{1}{3}} + 1 - ((1 - z_1^{out})(1 - z_2^{out})(1 - z_3^{out}))^{\frac{1}{3}}.$$

The contour converged quickly to the proper object.

The initial contour does not have to surround the object in order to find the object desired, as was the case for boundary–based active contours. An example of this can be found in Figure 11. Likewise the model finds an unregistered image segmentation as can be seen from Figure 12. All the examples so far are piecewise constant the model converges quickly in this case because they are piecewise constant images the parameters z_i^{in} and z_i^{out} will have two possible values each, a larger and smaller value. Thus it mimics the truth table values.

In the next two examples the models are not piecewise constant. In Figure 13, we show that unlike the channel by channel segmentation which had spurious data points when it had to segment $A_1 \cap \neg A_2$ using a single contour, our model gives the correct solution. In this case, the λ is smaller, thus the regularization term has a greater weight so that the noise is not included.

Finally, in Figure 14, we have a two channel image of the brain. In one we have a "tumor" with some noise, while the other is clear. The images are not registered. We want to find $A_1 \cap \neg A_2$ so that the tumor can be observed. This happens to be a very complicated example as there are a lot of features and textures. However the model finds the tumor successfully.

4 Comparison of Logic Active Contour Model with the Vector Valued Active Contour Model

The objective of this paper is related to that of vector valued active contours model in [2], both papers try to combine information from different channels of a vector-valued image in order to derive an active contour segmentation. The vector based model in [2] is as follows:

$$\inf F(C, c_1, c_2) = \mu(length(C)) + \lambda \int_{inside(C)} \sum_{i=1}^{n} |u_0^i - c_+^i|^2 dx + \quad (7)$$

$$\gamma \int_{outside(C)} \sum_{i=1}^{n} |u_0^i - c_-^i|^2 dx. \quad (8)$$

Empiraclly, it appears to give the union of several channels. In this section, we will try to compare the two different approaches. While the model in 8 may seem very different from the model presented in Section 3, with a little calculation we will see that it does follow a logic based format. Taking a Taylor

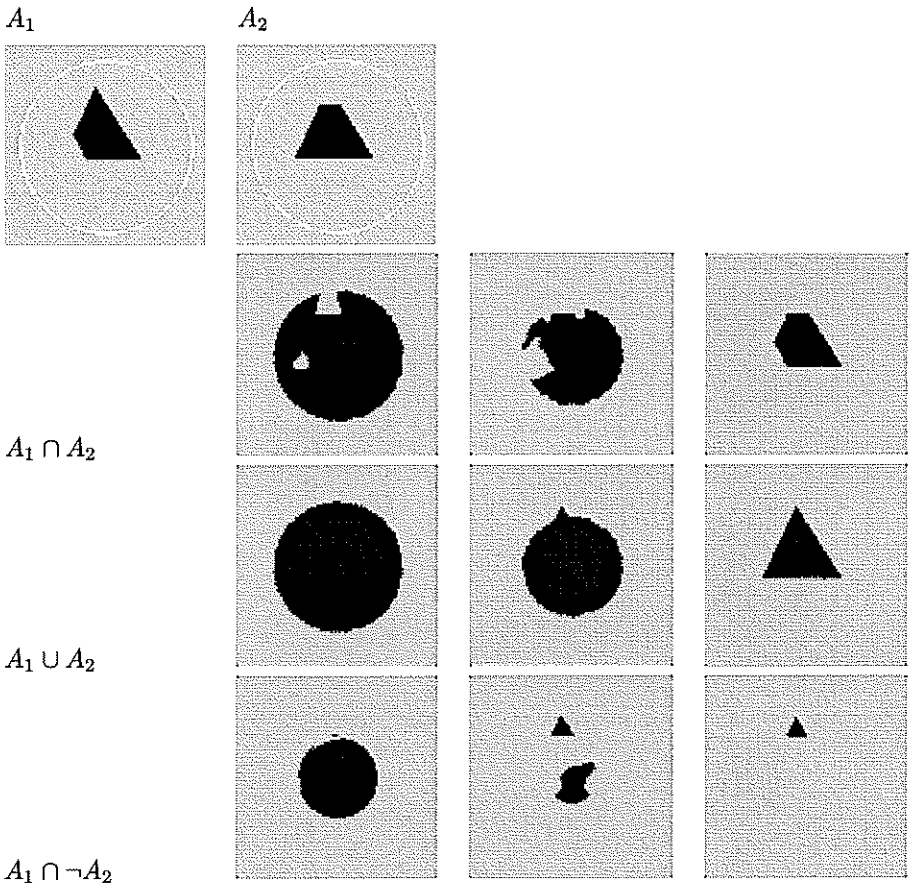


Figure 9: Region based logic model. This is the time evolution from left to right for the contour C evolving until it converges to the desired object.

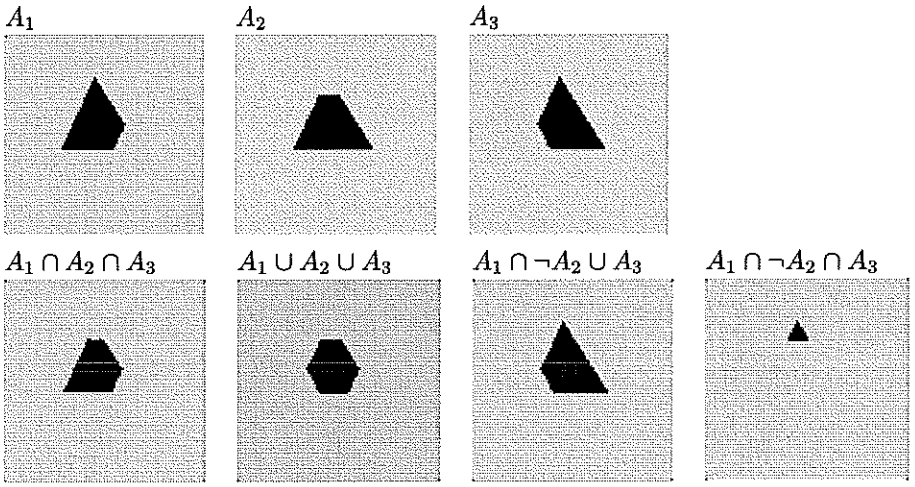


Figure 10: An example for the 3 channel case of the region based logic model. The model found the desired images quickly and correctly.

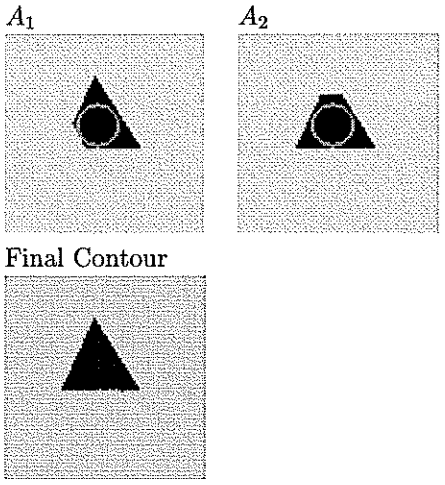


Figure 11: Region based logic model with initial contour inside the object. Even though the initial contour is inside the object, $A_1 \cup A_2$ is found.

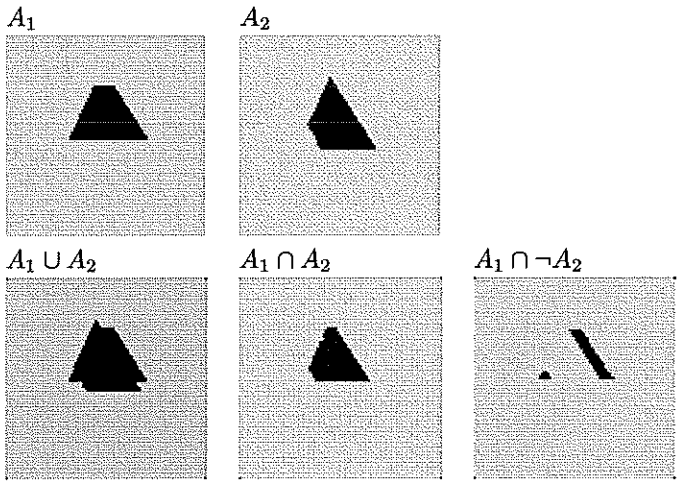


Figure 12: Region based logic model for unregistered images. The unregistered images still find $A_1 \cup A_2$ and $A_1 \cap A_2$ successfully, since the model is looking at the regions of union and intersection. Notice that when $A_1 \cap \neg A_2$ is found, it contains a part of the object that is due to the images being unregistered rather than the intrinsic object difference.

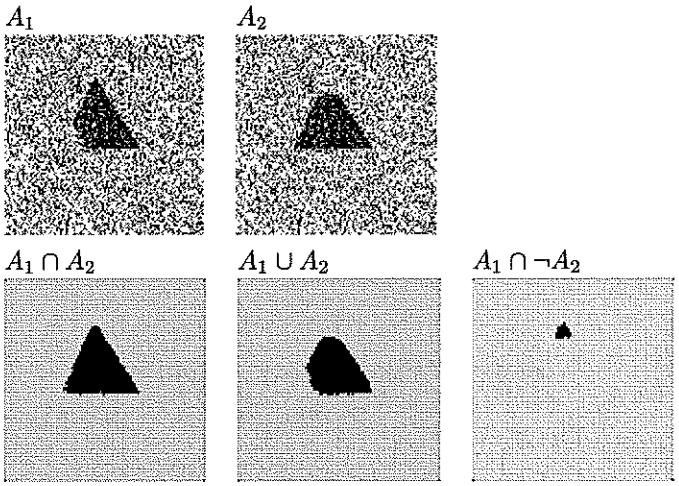


Figure 13: Region based logic model for noisy images. This is the same example as that for the channel–by–channel case (Fig. 3). While the union of the two channels is of comparable quality, the intersection is better when done using the logic operations method.

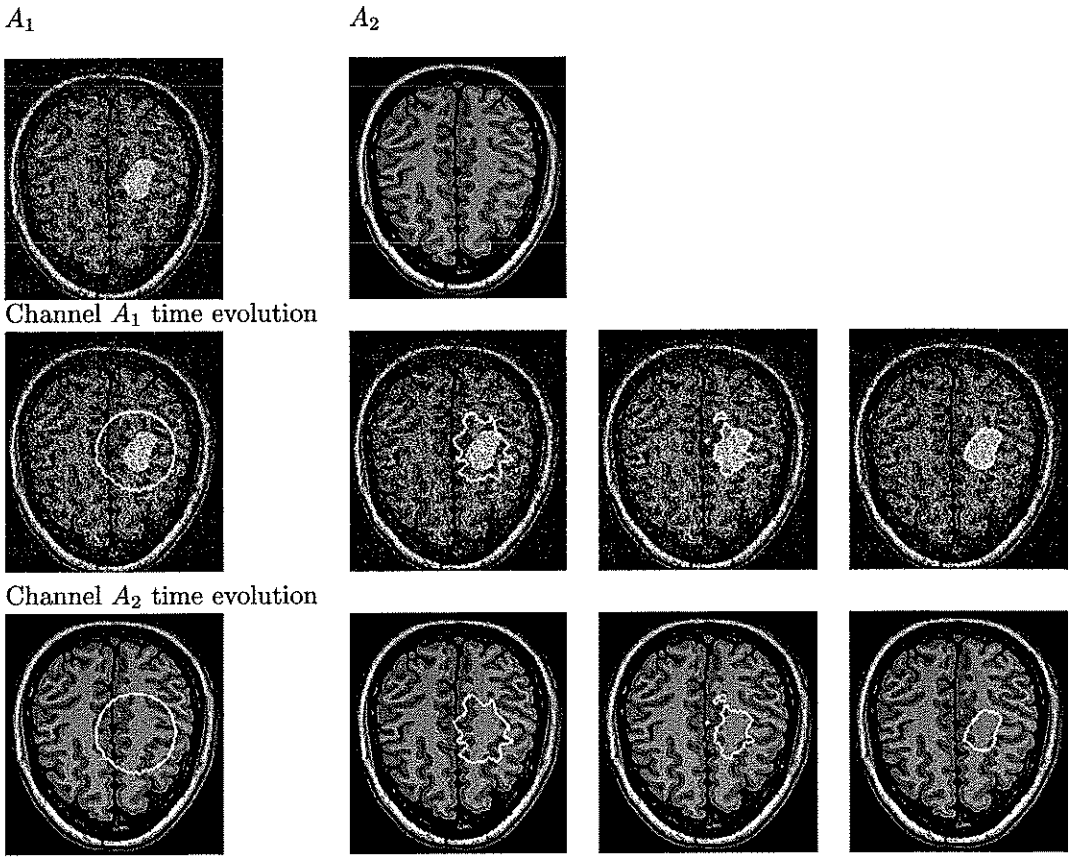


Figure 14: Region based logic model on a medical image. In the first channel A_1, the noisy image has a "brain tumor", while channel A_2 does not have a tumor. The images are not registered. We want to find the tumor that is in channel A_1, but not in A_2, thus we take the intersection of $A_1 \cap \neg A_2$. In the last two rows, we observe the time evolution (from left to right) as the contour deforms to find the "tumor".

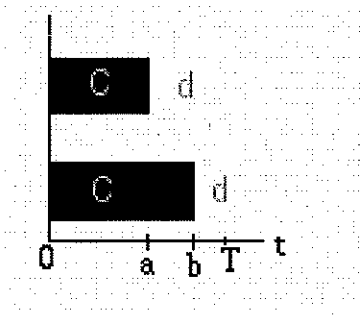


Figure 15: This a 1-D example where channel A_1 and A_2 both start from the same point. a is the length of A_1, b is the length of A_2, T is total length of the channels. The intensity inside the image is c and outside is d. $A_1 \cap A_2 = A_1$, while $A_1 \cup A_2 = A_2$.

expansion of $1 - \sqrt{(1-z_1)(1-z_2)}$

$$1 - \sqrt{(1-z_1)(1-z_2)} = \frac{z_1 + z_2}{2} + O(z_1 z_2).$$

The leading order term $z_1 + z_2$ corresponds to the vector based model. Thus the vector model is similar to taking the intersection of the inside of the contour and the intersection of the outside of the contour. Thus there is a conflicting objective in the vector model. Unlike the logic models, the error term inside the contour is not minimized at the contour as the error term for the outside of the object. The result is that the vector model has multiple minima.

We want to illustrate the difference between the vector model and the logic models. This is done in two ways: first as a 1-D analysis in which we can work the functional out analytically, second, a carefully chosen example to illustrate the differences.

Let us consider a 1-D example in Figure 15. In this example, a is the length of the object in Channel A_1, b is the length of the object in A_2, T is the total length of the channels. We will calculate the functionals for the vector valued, and logic models. In this example, t is the position at which the contour is located. The intensity values are c inside the object, and d outside.

For this example we can find the exact solution for the functionals in terms of a, b, c, d, t, and T. For this simple example the exact form of the variational formulations can be derived for both the vector model and the logic models. We will just compare the fitting terms so we set $\mu = 0$ for the functions in Eqns. 6 and 8. Three segments need to be considered to calculate the functional.
For $t < a$, we have:

$$c_+^1 = c, \quad c_-^1 = \frac{c(a-t)+d(T-a)}{T-t}, \quad c_+^2 = c, \quad c_-^2 = \frac{c(b-t)+d(T-b)}{T-t},$$

$$
\begin{aligned}
F_{vec}(t) &= \frac{(c-d)^2}{T-t}((T-a)(a-t)+(T-b)(b-t)),\\
F_{A_1\cup A_2}(t) &= \frac{(c-d)^2}{T-t}T-\frac{(c-d)^2}{T-t}\sqrt{(T-(a-t)(T-a))(T-(T-b)(b-t))},\\
F_{A_1\cap A_2}(t) &= \frac{(c-d)^2}{T-t}\sqrt{((T-a)(a-t)(T-b)(b-t)}.
\end{aligned}
$$

For $a \le t \le b$, we have:

$$
c_+^1 = \tfrac{ca+d(t-a)}{t}, \quad c_-^1 = d, \quad c_+^2 = c, \quad c_-^2 = \tfrac{c(b-t)+d(T-b)}{T-t},
$$

$$
\begin{aligned}
F_{vec}(t) &= (c-d)^2(\frac{a(t-a)}{t}+\frac{(T-b)(b-t)}{(T-t)}),\\
F_{A_1\cup A_2}(t) &= \frac{(c-d)^2}{T-t}T-\frac{(c-d)^2}{T-t}\sqrt{T(T-(T-b)(b-t))},\\
F_{A_1\cap A_2}(t) &= \frac{(c-d)^2}{t}T-\frac{(c-d)^2}{t}\sqrt{T(T-a(t-a))}.
\end{aligned}
$$

For $b < t < T, wehave:$

$$
c_+^1 = \tfrac{ca+d(t-a)}{t}, \quad c_-^1 = d, \quad c_+^2 = \tfrac{cb+d(t-b)}{t}, \quad c_-^2 = d,
$$

$$
\begin{aligned}
F_{vec}(t) &= \frac{(c-d)^2}{t}(a(t-a)+b(t-b)),\\
F_{A_1\cup A_2}(t) &= \frac{(c-d)^2}{t}T-\sqrt{(ab(t-a)(t-b))},\\
F_{A_1\cap A_2}(t) &= \frac{(c-d)^2}{t}T-\frac{(c-d)^2}{t}\sqrt{(1-a(t-a))(1-b(t-b))}.
\end{aligned}
$$

When we graph the functionals, we find that F_{vec} has two minima: one at $t = a$ the other at $t = b$. One of the minima is global the other is local depending on a, b and T. So if the contour C starts inside the smaller object, the functional will go to the minimum that corresponds to is the intersection of the objects while if the contour starts outside the smaller object, it converges to $t = b$ which corresponds to the union of the two objects. If the contour is in between, it will go to the edge it's closest to. See Figure 16.

Graphing $F_{A_1\cap A_2}$, we see that the only minimum occurs at the intersection of the two objects, in our case it is at $t = a$. Likewise, the minimum for $F_{A_1\cup A_2}$ occurs at the union of the two channels, which is at $t = b$. See Figures 16b and c. The above observations are verified in Figure 17, which shows the actual results of segmentation using the vector model and the two logic models.

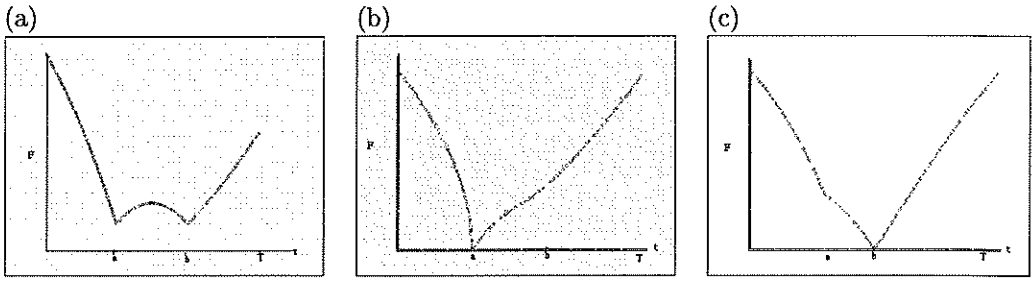


Figure 16: (a)This is the graph of the objective function for the vector valued model. The t-axis denotes the position of the contour in the model. It has 2 local minima, one at the intersection of the two objects located at $(a, F(a))$, and the other is located at $(b, F(b))$ which corresponds the union of the objects. The minima depends on where the initial contour is. If it is inside the smaller object the model converges to the intersection, otherwise it converges to the union. (b)This graph is the region logic model that finds the $A_1 \cap A_2$ of the two objects. Its global and only minimum occurs at $(a, F(a))$. (c) This graph is the region logic model that finds $A_1 \cup A_2$ of the two objects. Its global and only minimum occurs at $(b, F(b))$.

This 1-D example gives some insight into the differences between the vector and logic models. We now see that the vector model segments the image depending on the initial contour, rather than according to a global logic criteria. The logic models do not depend on the initial contour to find the global results, in a piecewise constant image.

However, to use the logic models properly, one needs to know which logic operator to use ahead of time. If the initial contour is chosen carefully, the vectore model can do a good job in choosing a desirable solution. An example of such a situation is given in Figures 18, 19. This is an example of the Kanisza face/vase image. Looking at the dark inside object one can see that it is a vase, but looking at the outside object it is two faces. So here we have information we want to preserve that is outside the object, as well as inside the object. The occlusion can be of the "inside" object, which we will define as the vase, when we would want the intersection of the outside object (i.e. faces). However, the occlusion can also be of the "outside" (faces) object, in which case we want the intersection of the inside. We can make the vector model act as the intersection of the vase (i.e. $A_1 \cap A_2$) if the initial contour is small and inside both channels. If the initial contour surrounds the objects in both channels it will act as the logic model for the union of the vase ($A_1 \cup A_2$), or intersection of the faces, see Figure 19. Since the vector model depends on the initial contour, we can choose an initial contour close to the boundary of the inside object (vase) or the outside object face to get the desired affect. The vector model always tries to compute the logical "AND" (i.e. intersection) of the objects in the two channels. it will go to the intersection of the two faces, in one set of images, and the intersection

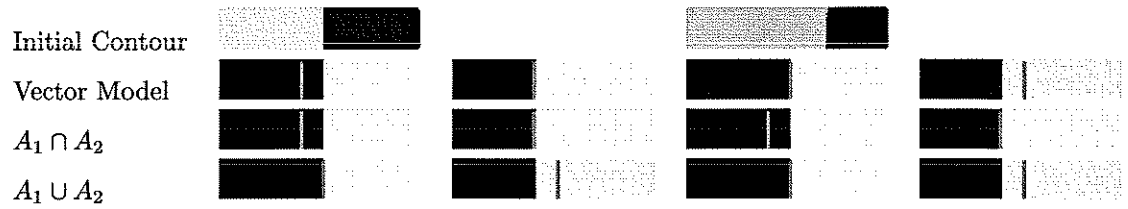


Figure 17: This is a simple example that shows the object found for the vector model depends on the initial contour for the segmentation, if the contour begins inside the object, it will find $A_1 \cap A_2$ of the images, otherwise it will find $A_1 \cup A_2$ of the images. Calculating the logic model using $A_1 \cap A_2$, the same segmentation is found for the initial contour inside or outside the object. Likewise $A_1 \cup A_2$ is found for the initial contour inside and outside the object.

of the vase in the other.

Rather than choosing which logic operation, a careful choice of the initial contour for the vector model can give the desired result, which can correspond to either the union or the intersection of the inside object.

The logic models will give a robust segmentation. If for example, the intersection of the objects is desired, one only needs to choose the model that calculates the intersection. However in situations when one doesn't know ahead of time which model is preferable, a careful guess of the initial contour using the vector model may give the best compromise.

5 Conclusion

A generalized model for multi–channel images has been presented. It allows the user to choose the information that is extracted from the set of images using general logic combinations of union, intersection and negation. This was presented for both boundary based and region–based models. We found that the region–based models using the particular example of the Chan-Vese [4] active contours without edges model, was able to find the unions and intersections of regions, rather than boundaries, and hence is closer to human perception. The boundary based method found boundaries where there were no regions.

Experiments on two channel and three channel systems were presented. They demonstrated the ability of the models to detect the objects as described by the logic operations accurately and quickly.

We compared the logic models to the vector model derived previously. This showed some interesting elements to the concept of a global minimum versus multiple minima. An example was made that shows that perhaps sometimes a local minimum, i.e. a vector model is preferable.

It would be interesting and straightforward to apply this model to object tracking in movie sequences, and in registration of multi–channel images.

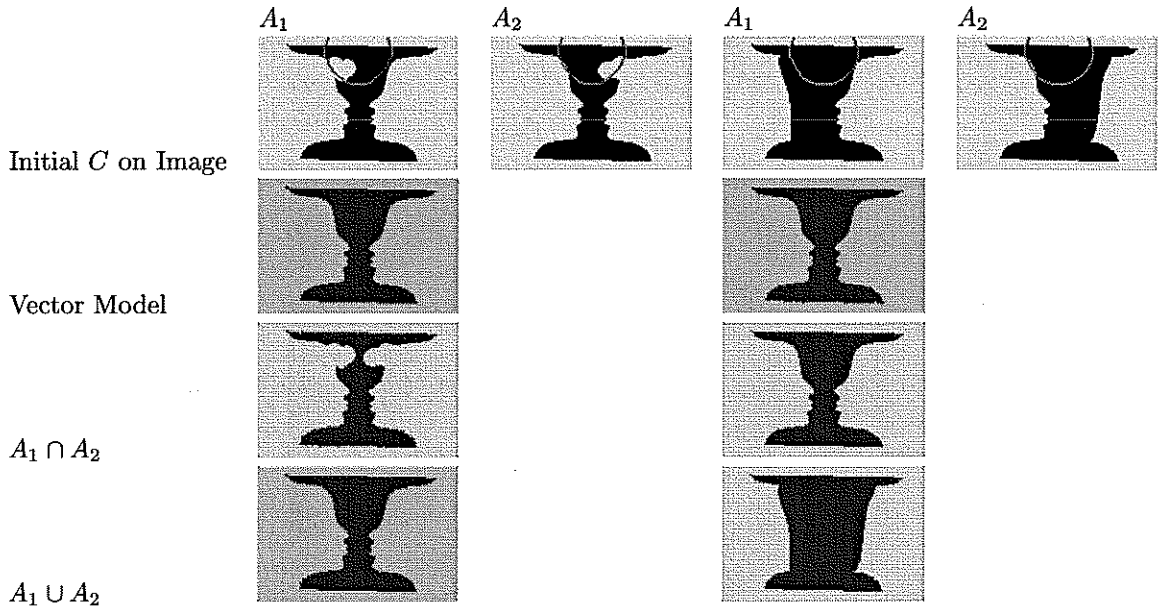


Figure 18: In this example, we show when the vector model might be more appropriate than the logic model. In the first set of images, we need to take $A_1 \cup A_2$ to recover the Kanizsa face vase image, while in the second we need to take $A_1 \cap A_2$ to get the same image back. This requires a priori knowledge at the time the image is being segmented, while the vector model finds the desired object using the same initial contours in both cases.

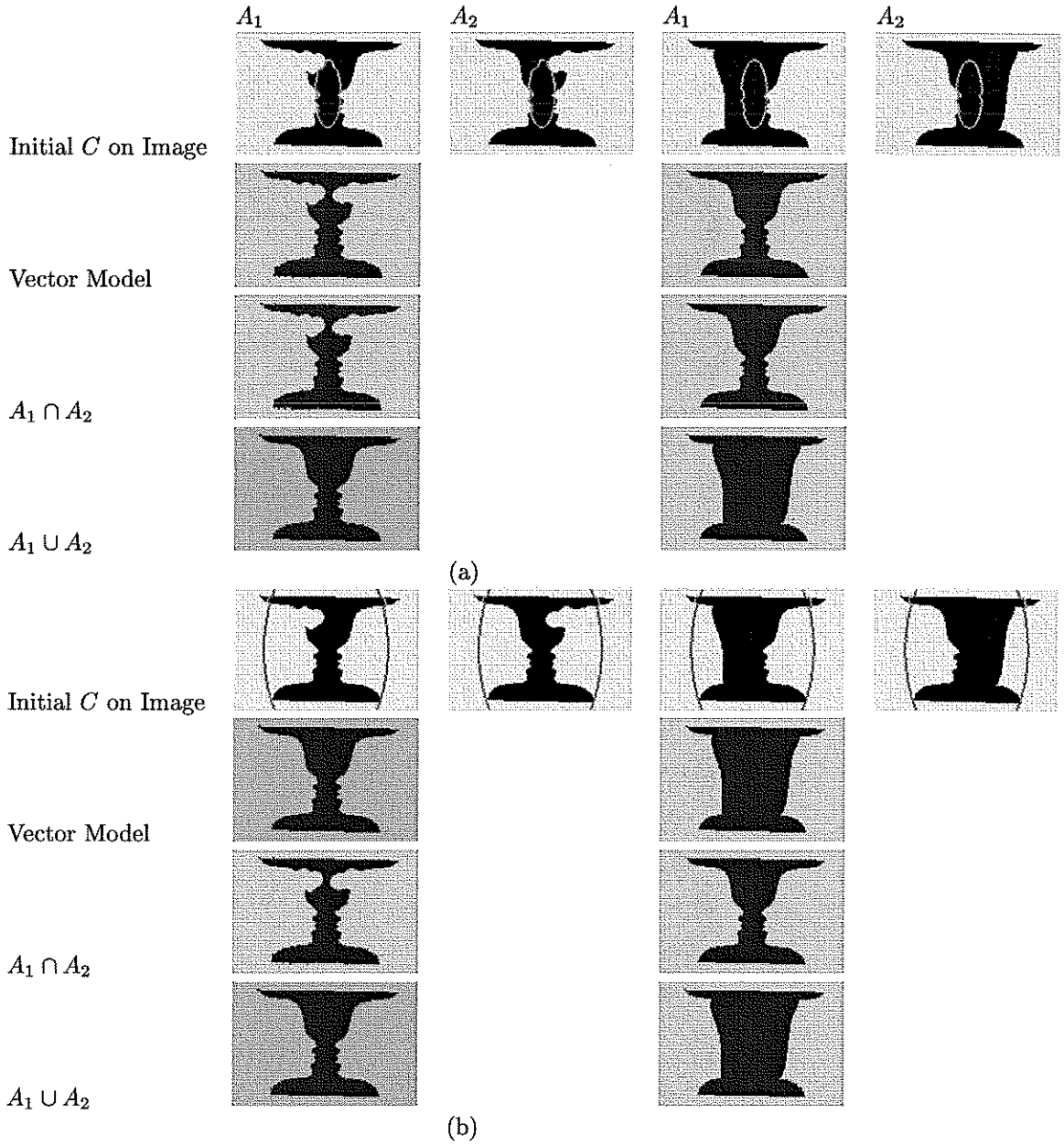


Figure 19: In this example we show how using the vector model one can get different logic models through the manipulation of the initial contours. For (a) the initial contour is inside the object in both cases, and the vector model computes the intersection of the inside objects which is the logic equivalent of $A_1 \cap A_2$. For (b) the initial contour is outside the object in both cases and the vector model computeses the the intersection of the outside objects which is the logic model equivalent to $A_1 \cup A_2$.

References

[1] V. Caselles, R. Kimmel, and G. Sapiro. Geodisic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[2] T. Chan, B. Sandberg, and L. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11:130–141, 1999.

[3] T. Chan, B. Sandberg, and L. Vese. Active contours without edges for textured images. *CAM report 02-28*, 2002.

[4] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 16(2):266–277, 2000.

[5] F. Dibos and G. Koepfler. Color segmentation using a variational formulation. *Actes du 16me Colloque GRETSI*, pages 367–370, 1997.

[6] F. Guichard. A morpholocial affine and galilean invarient scale space for movies. *IEEE Image Processing*, 7(3):444–456, 1998.

[7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[8] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[9] N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. *Proceedings of the 7th International Conference on Computer Vision*, pages 100–115, 1999.

[10] G. Sapiro. Color snakes. *Computer Vision and Image Understanding*, pages 247–253, 1997.

[11] G. Sapiro. Vector (self) snakes: a geometric framework for color, texture, and multiscale image segmentation. *Proc. IEEE ICIP, Lausanne*, I:817–820, September 1996.

[12] G. Sapiro and D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Transactions on Image Processing*, 5:1582–1586, 1996.

[13] H.K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, 1996.

[14] 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*, 18(9):884–900, 1996.