



Active contours without edges

Tony F. Chan & Luminita A. Vese

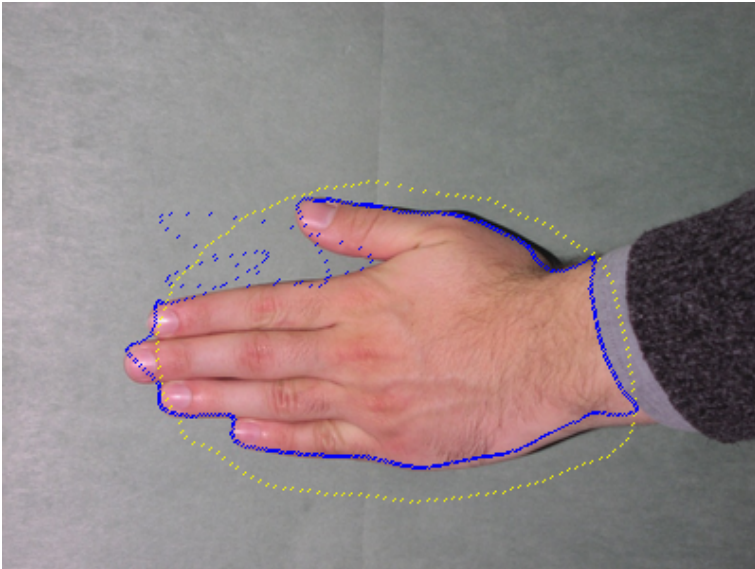
A logic framework for active contours on multi-channel images

Berta Sandberg & Tony F. Chan

Presented by: Sarah Nadi & Karim Ali

Oct. 28Th 2010 – CS 870

Big Picture

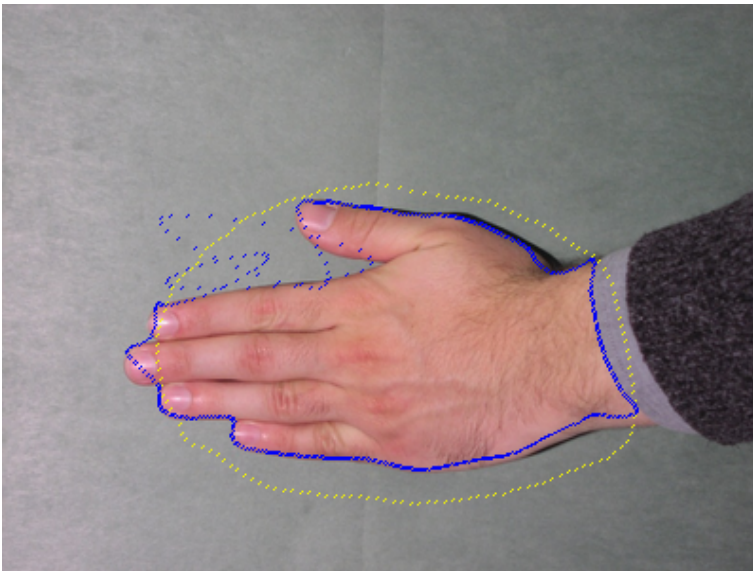


Detecting objects in an image



Combining multiple channels

Big Picture



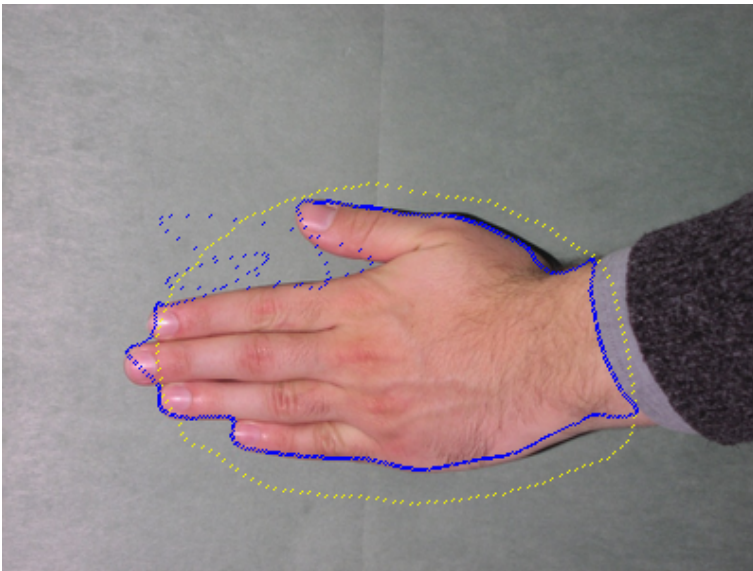
Detecting objects in an image

Chan-Vese model



Combining multiple channels

Big Picture



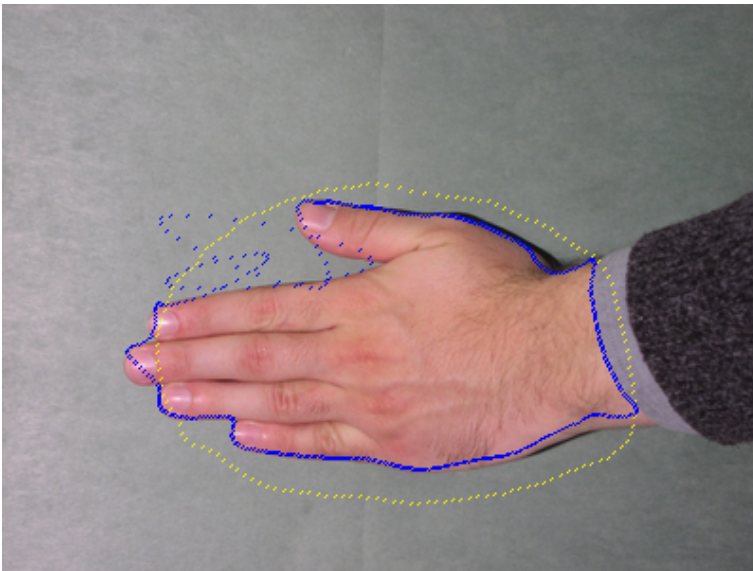
Detecting objects in an image

Chan-Vese model



Combining multiple channels

Big Picture



Detecting objects in an image

Chan-Vese model



Combining multiple channels

Sandberg-Chan framework

Outline

- Active contours without edges
 - Motivation
 - Model
 - Level set formulation
 - Experimental results
- Logic framework for multichannel images
 - Motivation
 - Logic operations
 - Level set formulation
 - Experimental results
- Conclusions

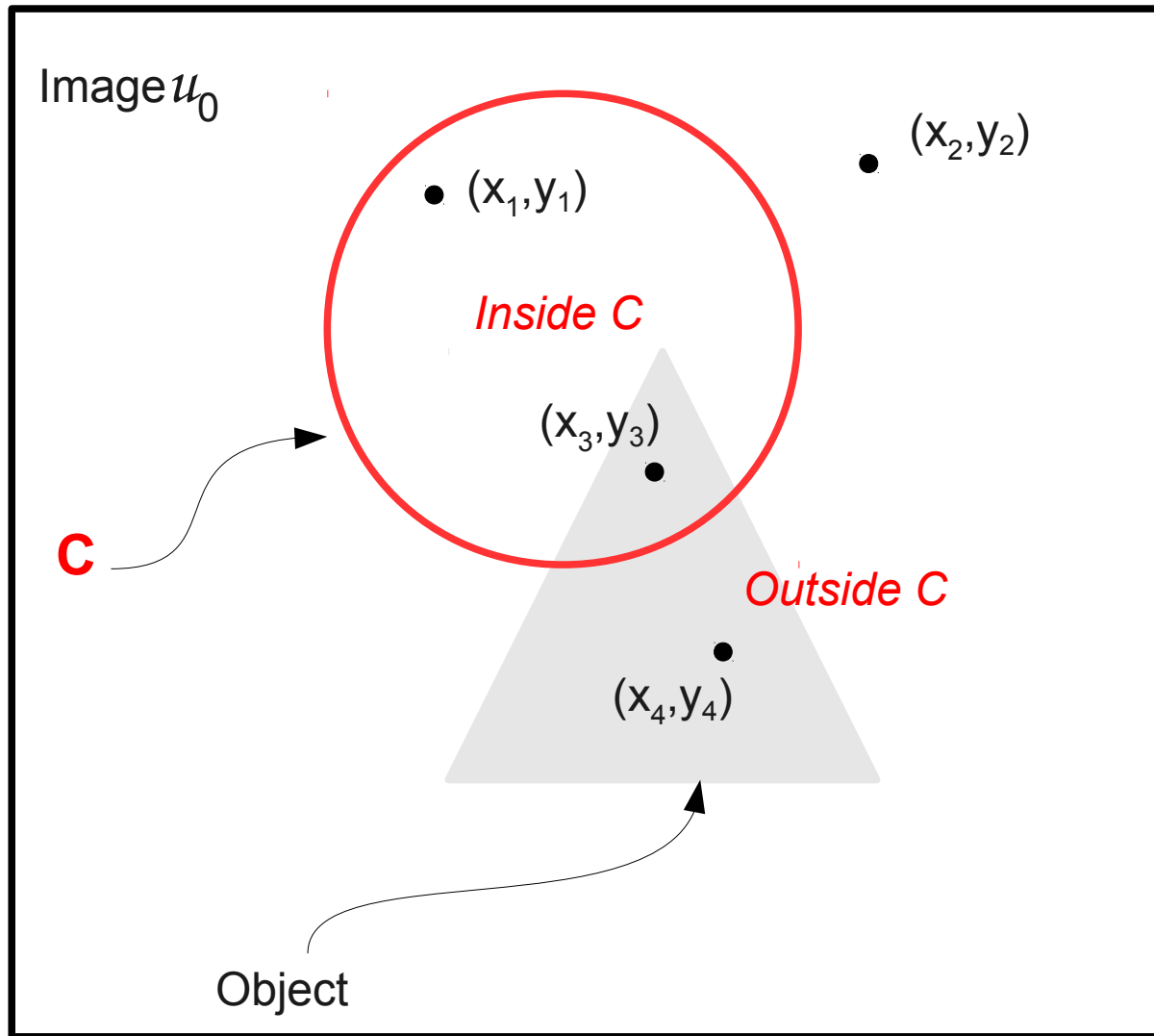
Active Contour Models

- Overview
 - Detect objects in an image
 - Use image gradient to evaluate the stopping condition
- Limitations
 - Can only detect edges defined by a gradient
 - Edges can be smoothed
 - Contour may pass through the boundary

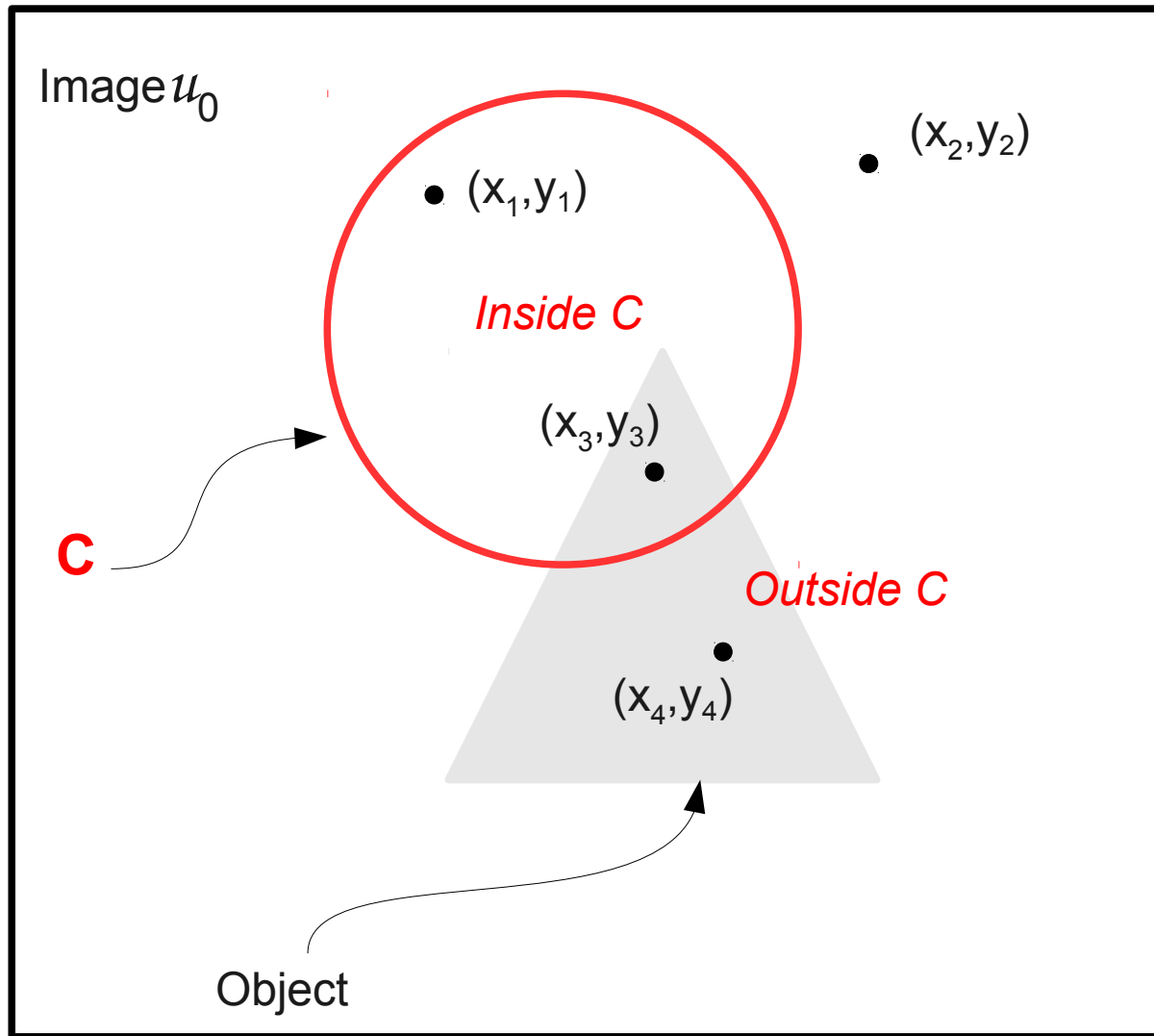
Active Contours Without Edges (Chan-Vese)

- Region based technique
- Stopping is based on Mumford-Shah segmentation techniques
 - No longer depends on gradient
 - Detects smooth or discontinuous boundaries
- Interior contours are automatically detected
- Initial curve can be anywhere in the image

Region Based View



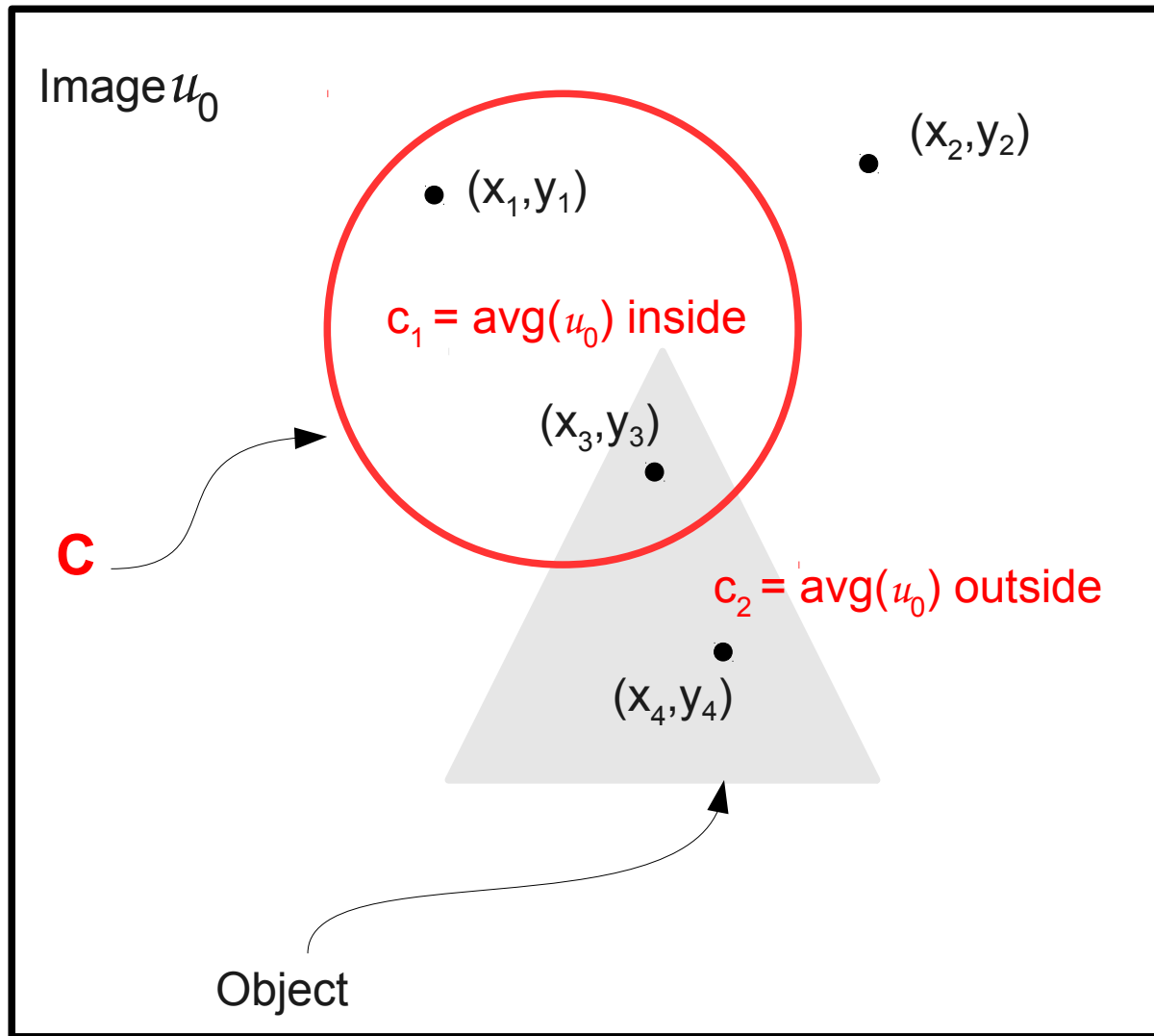
Region Based View



To fit C to our object,
we want:

- (x_1, y_1) to be **outside** C
- (x_2, y_2) to be **outside** C
- (x_3, y_3) to be **inside** C
- (x_4, y_4) to be **inside** C

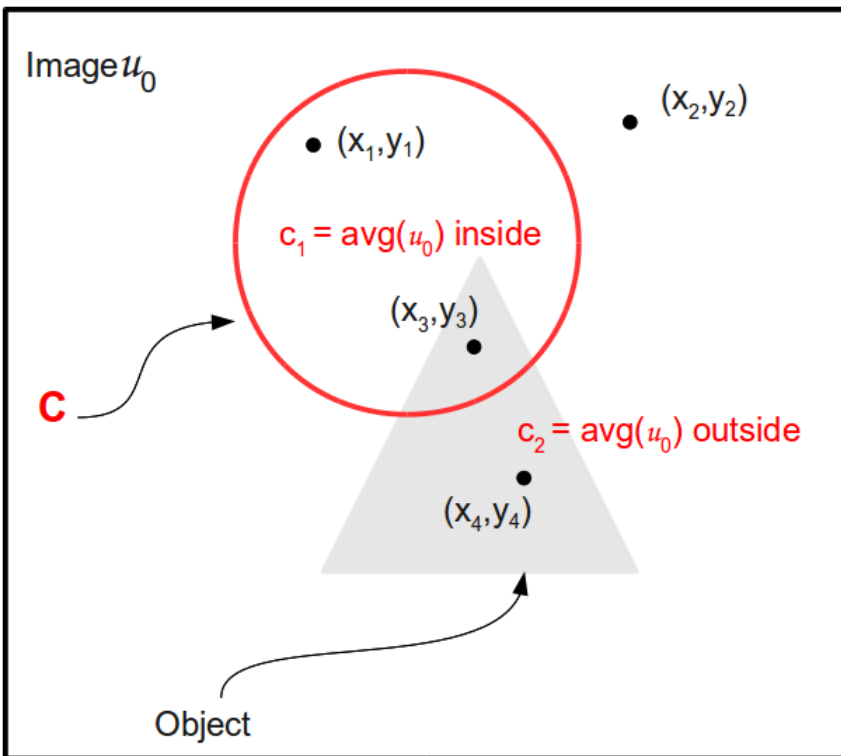
Region Based View



To fit C to our object,
we want:

- (x_1, y_1) to be **outside** C
- (x_2, y_2) to be **outside** C
- (x_3, y_3) to be **inside** C
- (x_4, y_4) to be **inside** C

Curve Fitting



Fitting inside:

$$F_1(C) = \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy$$

Fitting outside:

$$F_2(C) = \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy$$

Overall fitting of C:

$$\begin{aligned} F_1(C) + F_2(C) &= \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy \\ &\quad + \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy \end{aligned}$$

Intuition

- C should stop evolving when $C = \text{Object}$
- To achieve that:
 $\text{Fitting} \approx 0$

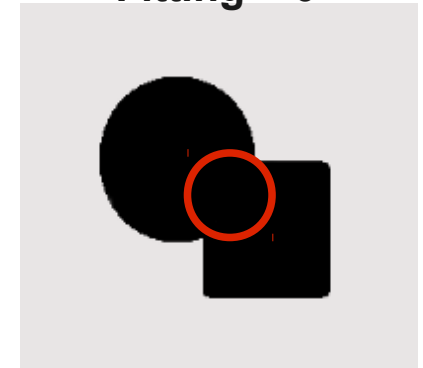
$$F_1(C) > 0, F_2(C) \approx 0$$

$$\text{Fitting} > 0$$



$$F_1(C) \approx 0, F_2(C) > 0$$

$$\text{Fitting} > 0$$



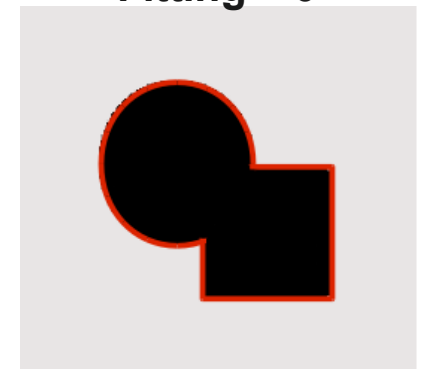
$$F_1(C) > 0, F_2(C) > 0$$

$$\text{Fitting} > 0$$



$$F_1(C) \approx 0, F_2(C) \approx 0$$

$$\text{Fitting} \approx 0$$



Energy Function

$$\begin{aligned} F(c_1, c_2, C) = & \lambda_1 \int_{inside(C)} |u_0(x, y) - c_1|^2 dx dy \\ & + \lambda_2 \int_{outside(C)} |u_0(x, y) - c_2|^2 dx dy \end{aligned}$$

Energy Function

$$\begin{aligned} F(c_1, c_2, C) = & \mu \cdot \text{Length}(C) \\ & + \nu \cdot \text{Area}(\text{inside}(C)) \\ & + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy \\ & + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy \end{aligned}$$

Energy Function

$$\begin{aligned} F(c_1, c_2, C) = & \mu \cdot \text{Length}(C) \\ & + \nu \cdot \text{Area}(\text{inside}(C)) \\ & + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy \\ & + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy \end{aligned}$$

- Look at minimization problem:

$$\inf_{c_1, c_2, C} F(c_1, c_2, C)$$

Level Set Formulation of Energy Function

$$\begin{aligned} F(c_1, c_2, C) = & \quad \mu \cdot \text{Length}(C) \\ & + \nu \cdot \text{Area}(\text{inside}(C)) \\ & + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy \\ & + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy \end{aligned}$$

Level Set Formulation of Energy Function

$$\begin{aligned} F(c_1, c_2, C) = & \mu \int_{\Omega} \delta_0(\phi(x, y)) |\nabla \phi(x, y)| dx dy \\ & + \nu \cdot \text{Area}(\text{inside}(C)) \\ & + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy \\ & + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy \end{aligned}$$

Level Set Formulation of Energy Function

$$\begin{aligned} F(c_1, c_2, C) = & \mu \int_{\Omega} \delta_0(\phi(x, y)) |\nabla \phi(x, y)| dx dy \\ & + \nu \int_{\Omega} H(\phi(x, y)) dx dy \\ & + \lambda_1 \int_{inside(C)} |u_0(x, y) - c_1|^2 dx dy \\ & + \lambda_2 \int_{outside(C)} |u_0(x, y) - c_2|^2 dx dy \end{aligned}$$

Level Set Formulation of Energy Function

$$\begin{aligned} F(c_1, c_2, C) = & \mu \int_{\Omega} \delta_0(\phi(x, y)) |\nabla \phi(x, y)| dx dy \\ & + \nu \int_{\Omega} H(\phi(x, y)) dx dy \\ & + \lambda_1 \int_{\Omega} |u_0(x, y) - c_1|^2 H(\phi(x, y)) dx dy \\ & + \lambda_2 \int_{outside(C)} |u_0(x, y) - c_2|^2 dx dy \end{aligned}$$

Level Set Formulation of Energy Function

$$\begin{aligned} F(c_1, c_2, C) = & \mu \int_{\Omega} \delta_0(\phi(x, y)) |\nabla \phi(x, y)| dx dy \\ & + \nu \int_{\Omega} H(\phi(x, y)) dx dy \\ & + \lambda_1 \int_{\Omega} |u_0(x, y) - c_1|^2 H(\phi(x, y)) dx dy \\ & + \lambda_2 \int_{\Omega} |u_0(x, y) - c_2|^2 (1 - H(\phi(x, y))) dx dy \end{aligned}$$

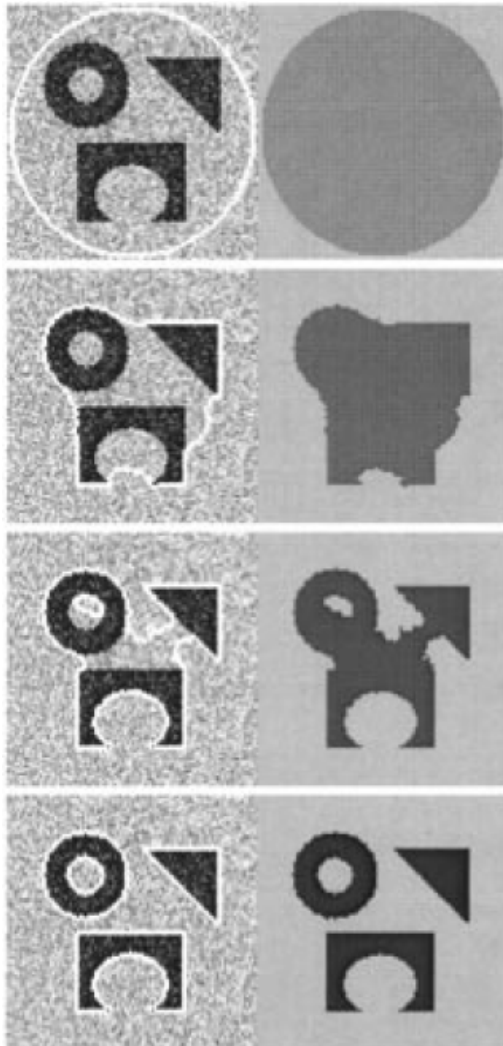
Chan-Vese Technique

$$\frac{\partial \phi}{\partial t} = \delta_{\varepsilon}(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2 \right]$$

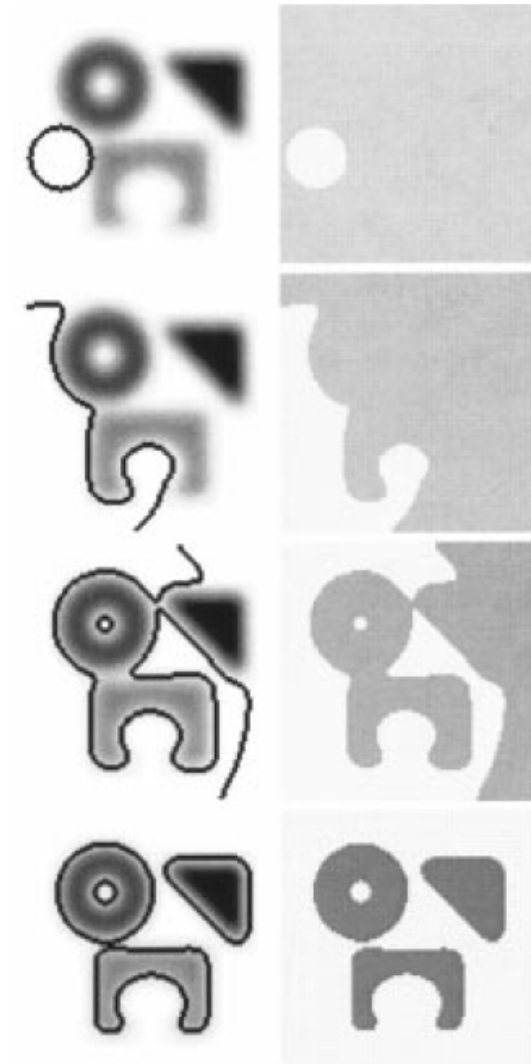
- Initialize ϕ^0 by ϕ_0 , $n = 0$
- Compute $c_1(\phi^n)$ and $c_2(\phi^n)$
- Solve the PDE to obtain ϕ_{n+1}
- Reinitialize ϕ locally to the signed distance function to the curve (optional)
- Check whether the solution is stationary. If not $n = n + 1$ and repeat

Experimental Results

Noisy Image

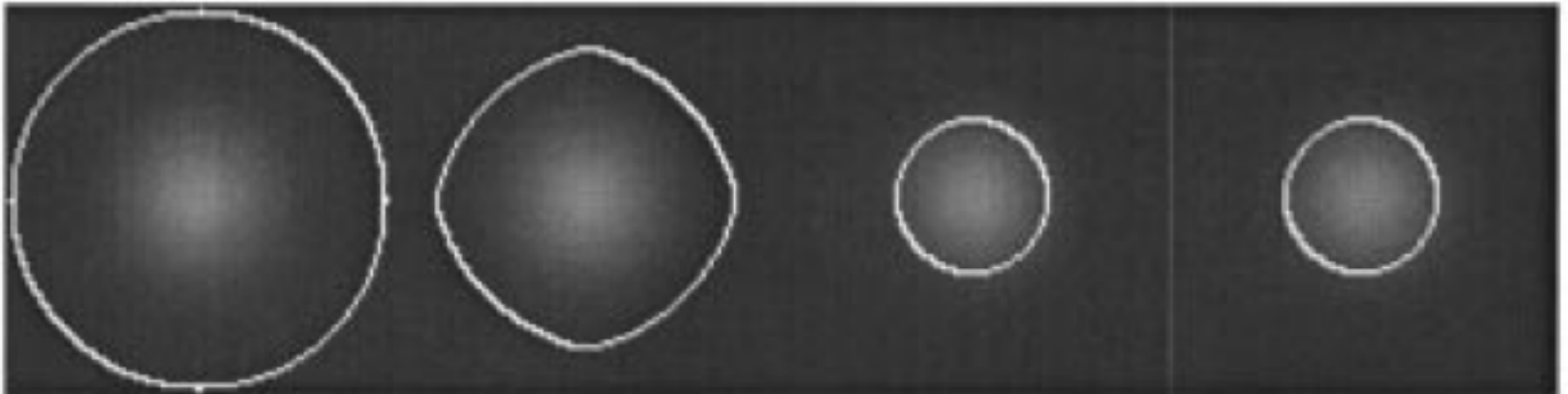
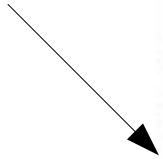


Blurry Image

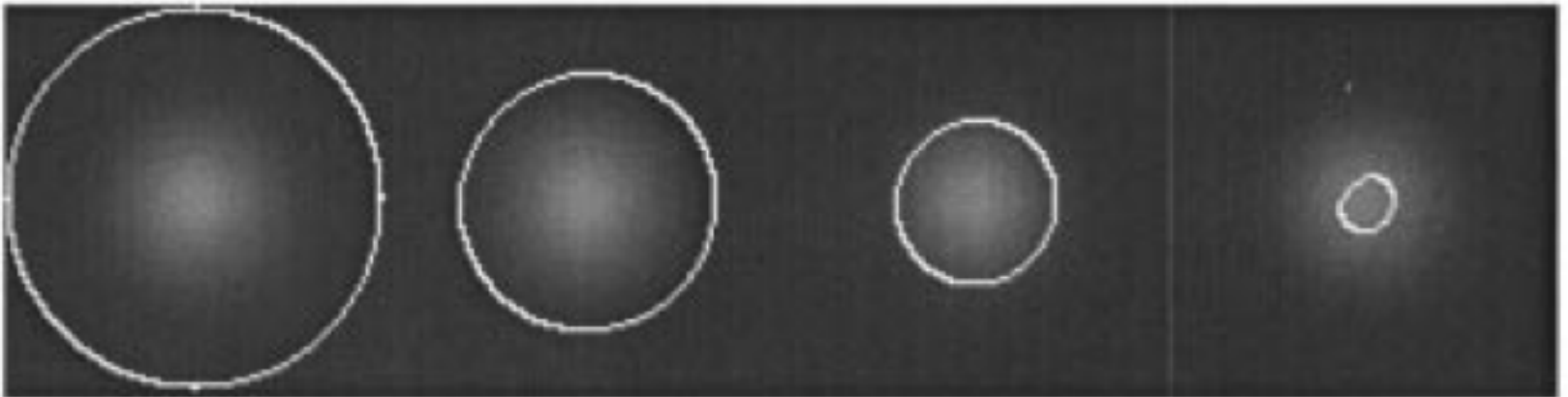
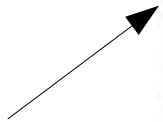


Comparing Models

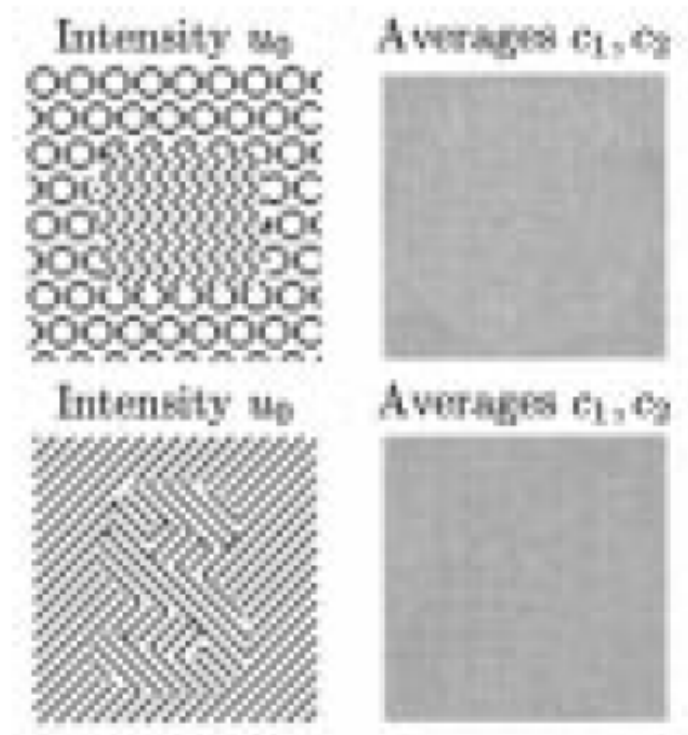
Chan-Vese



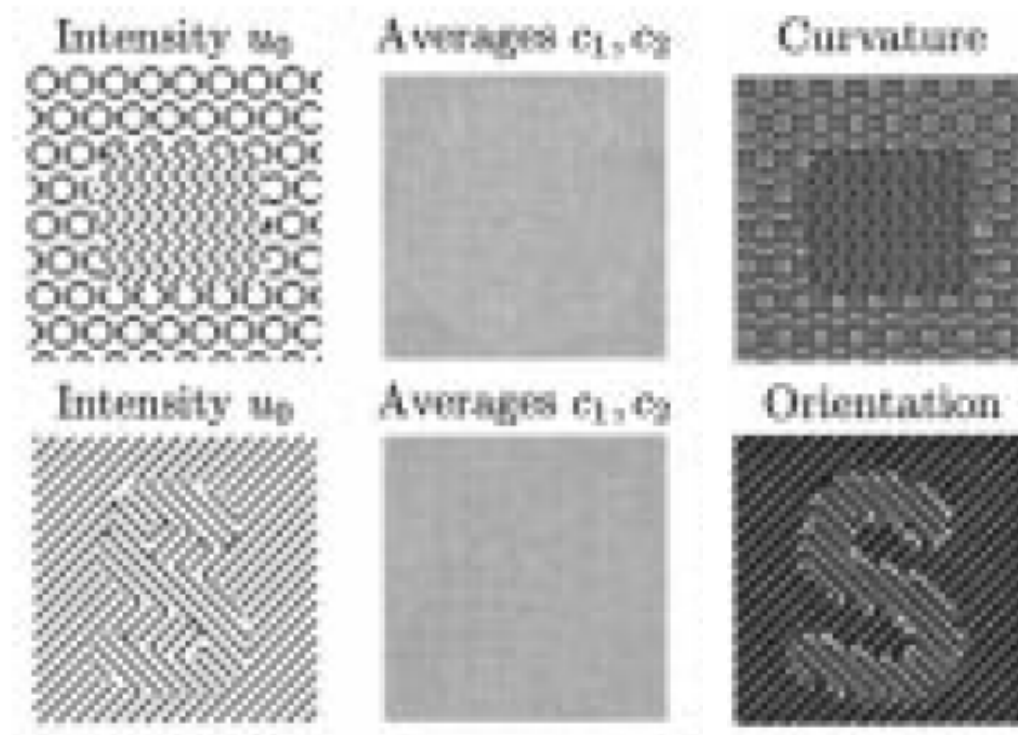
Gradient based
active contour



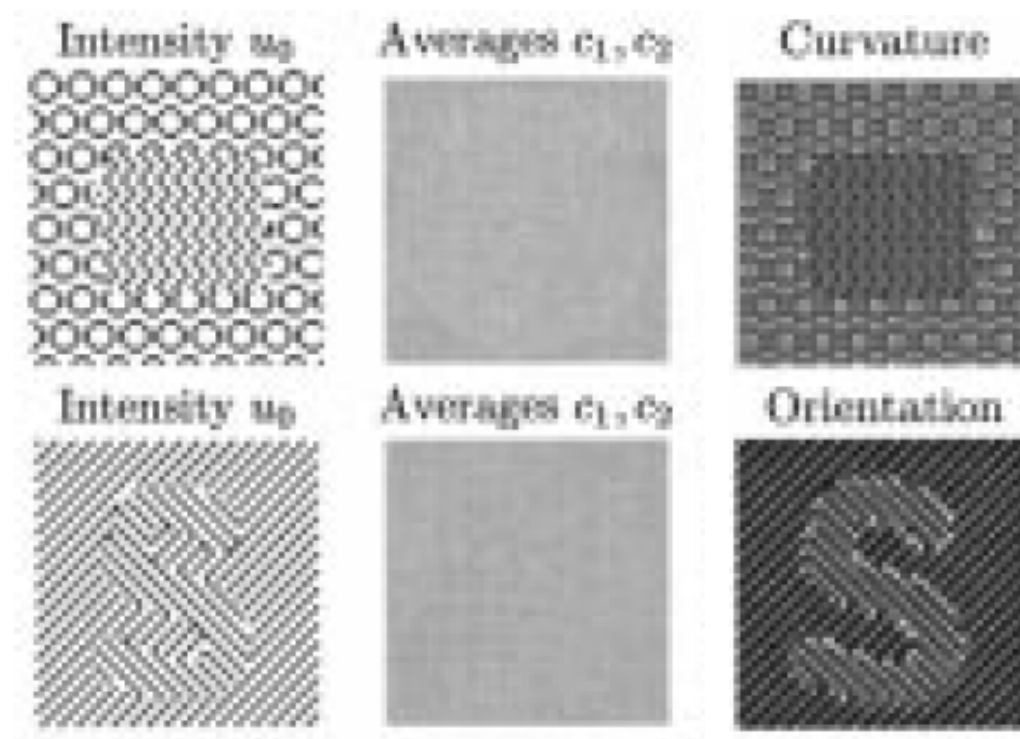
Limitations



Limitations



Limitations

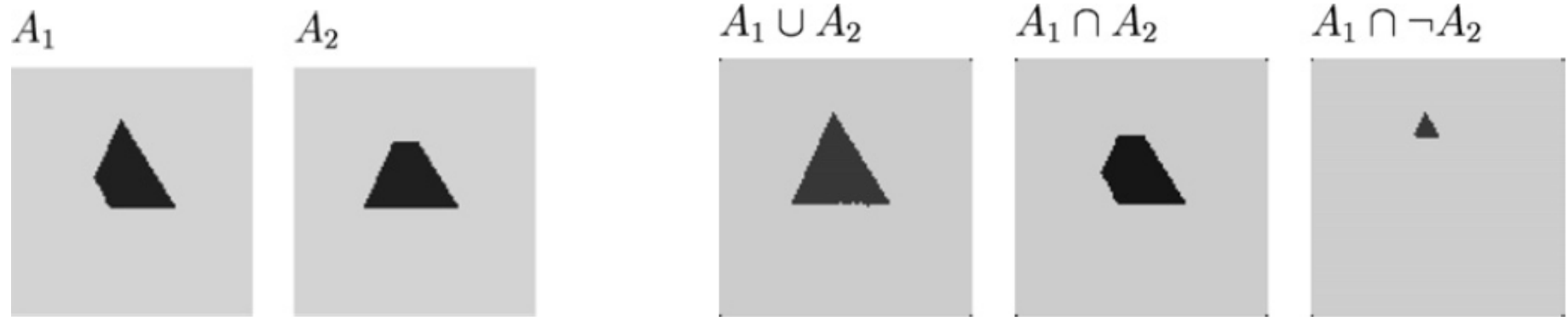


Or implement general case of Mumford-Shah function

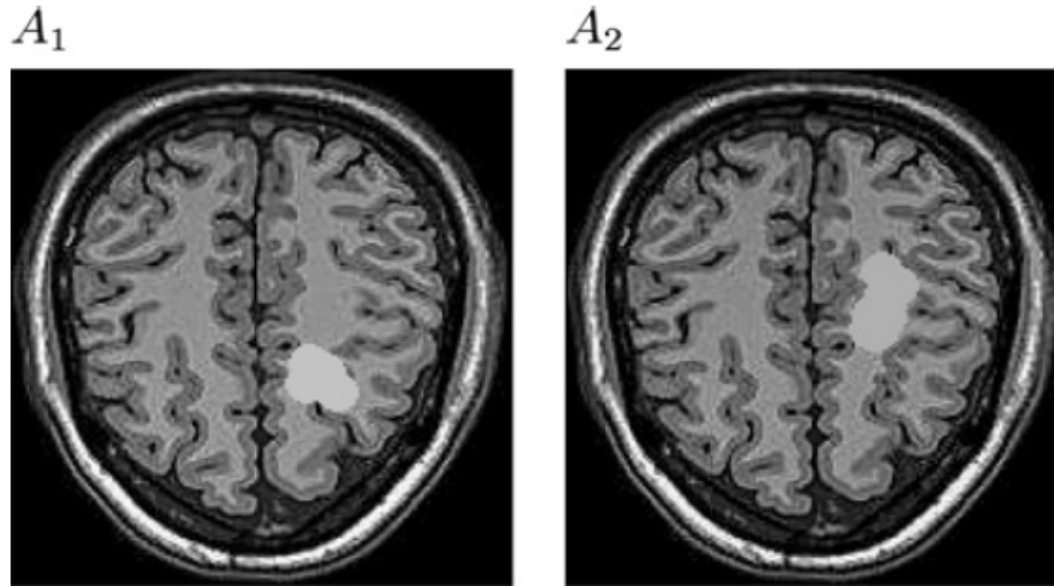
Applying the Chan-Vese Model to Multi-channel Images

The Sandberg-Chan Framework

- Allows the user to produce any logical combination of object channels



Motivating Example



$$\neg A_1 \cap A_2$$

New tumor growth

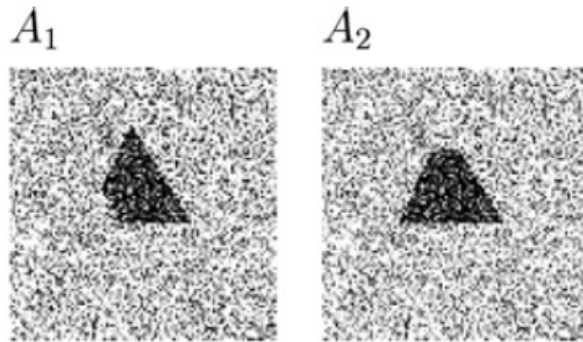
$$A_1 \cap \neg A_2$$

Tumor shrinking

Existing Techniques

- 1) Segment each channel independently followed by bitwise operations
 - Costly
- 2) Perform logic operations on channels forming a combined image followed by segmentation
 - Requires a lot of information about intensity

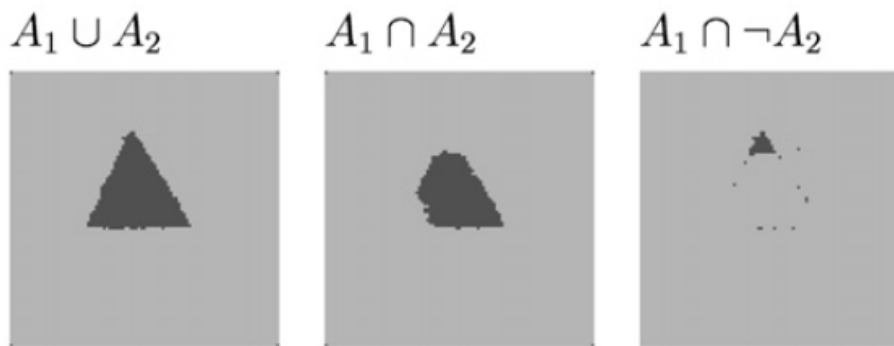
Technique 1



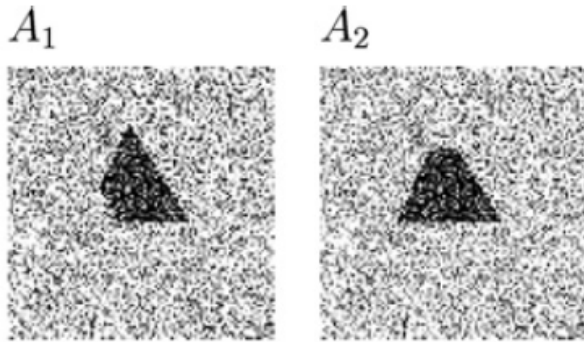
Segmentation Channel by Channel with Active Contours without edges:



Bitwise Logic Operations



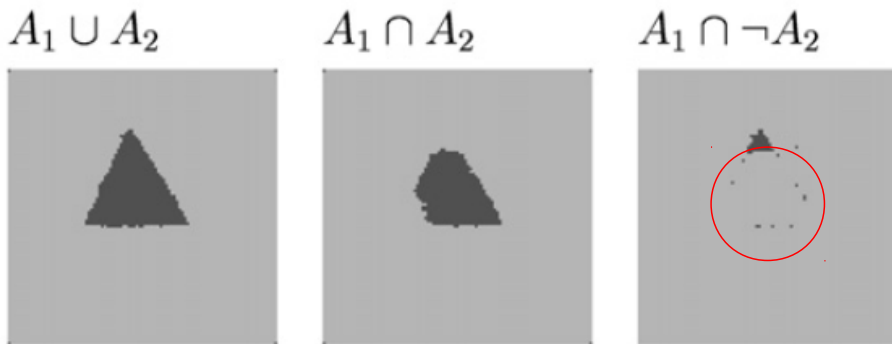
Technique 1



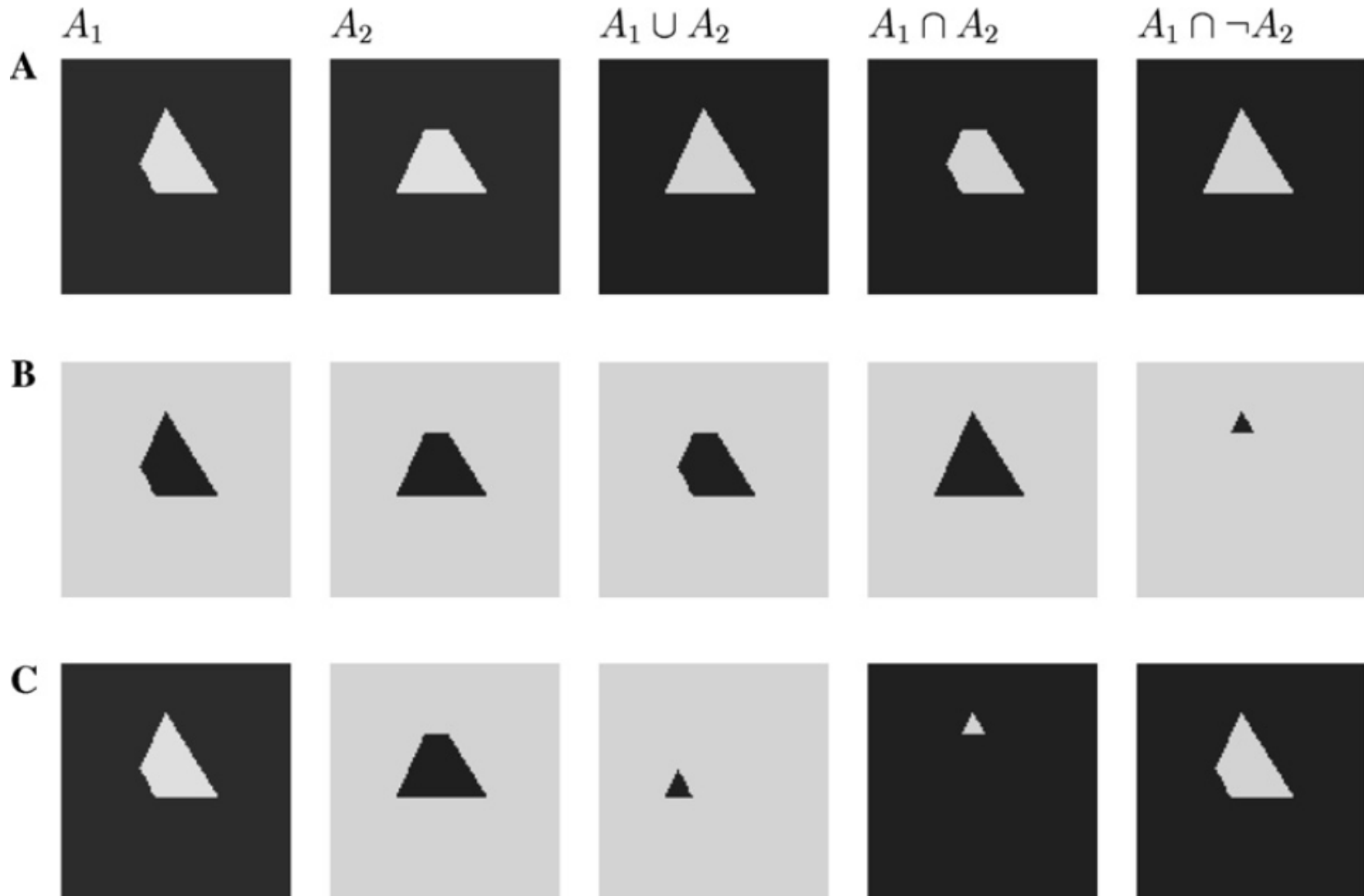
Segmentation Channel by Channel with Active Contours without edges:



Bitwise Logic Operations



Technique 2

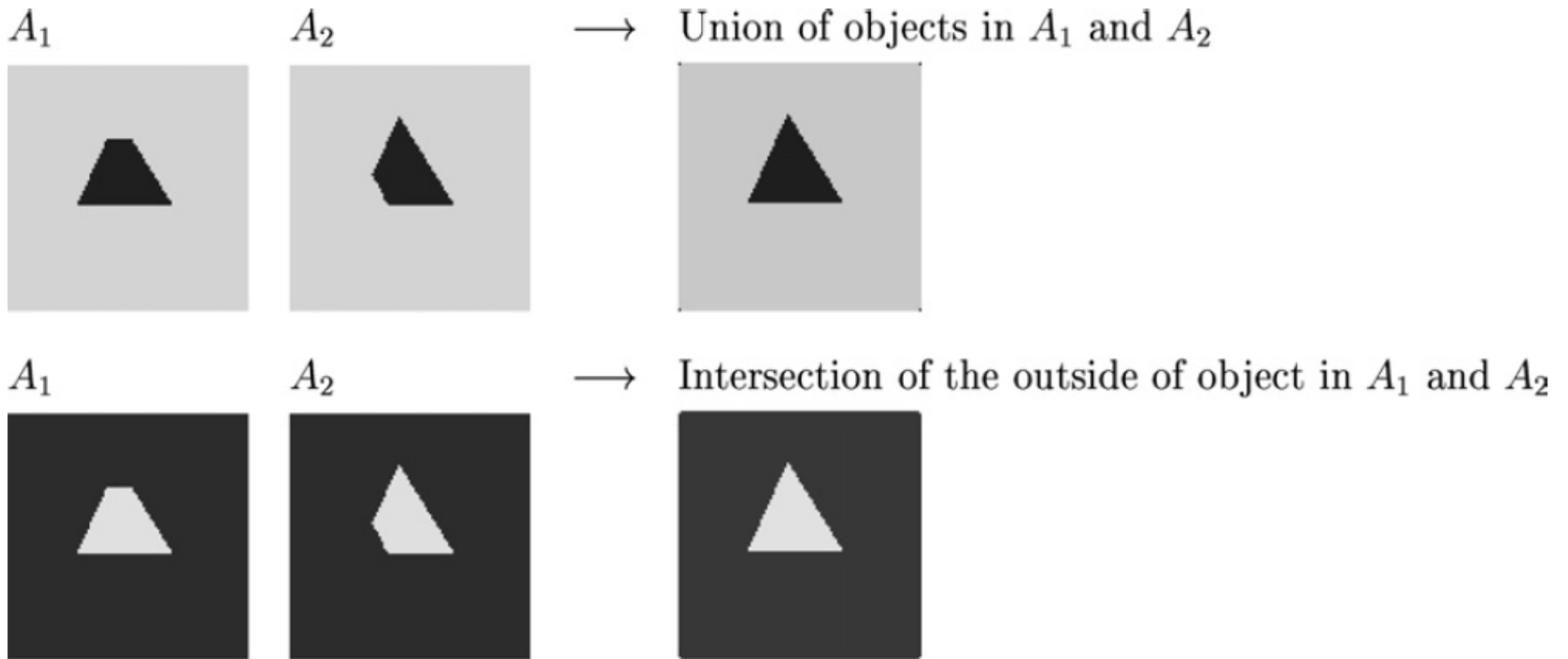


Proposed Solution

- Use the Chan-Vese model, and compare the contour fitting on each channel
 - Start with the same contour on each channel
 - Fit contour to the object on ALL channels according to the logic operator & based on regions



Region Based Logic Operations



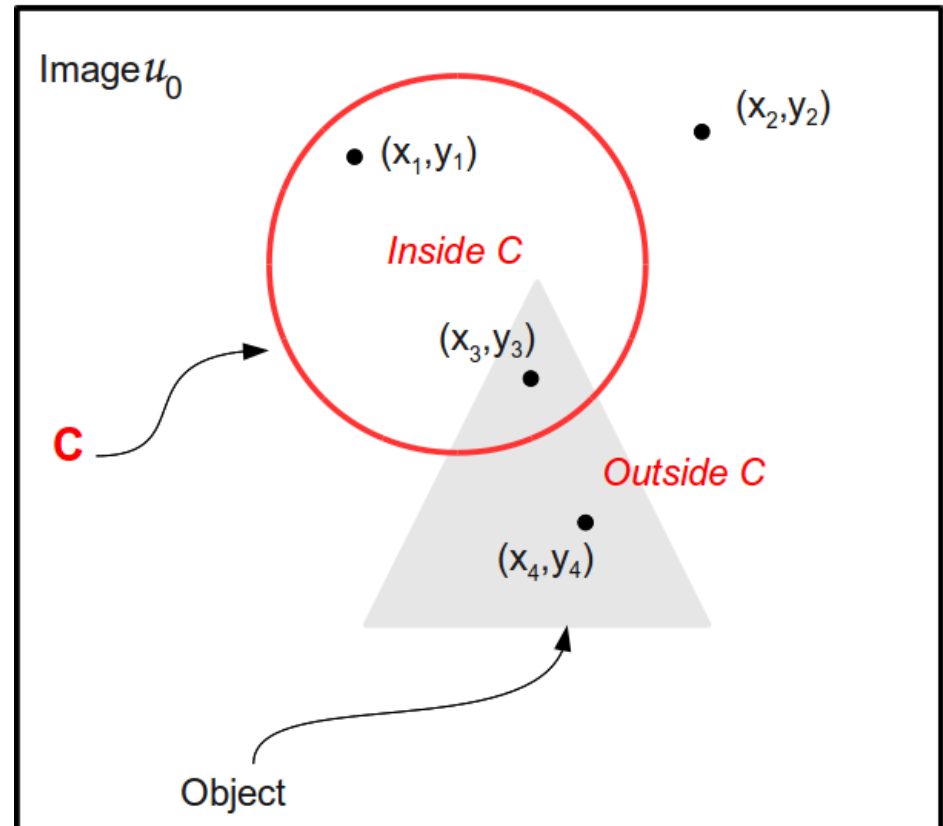
Logic Variables

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

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

- *Note:*

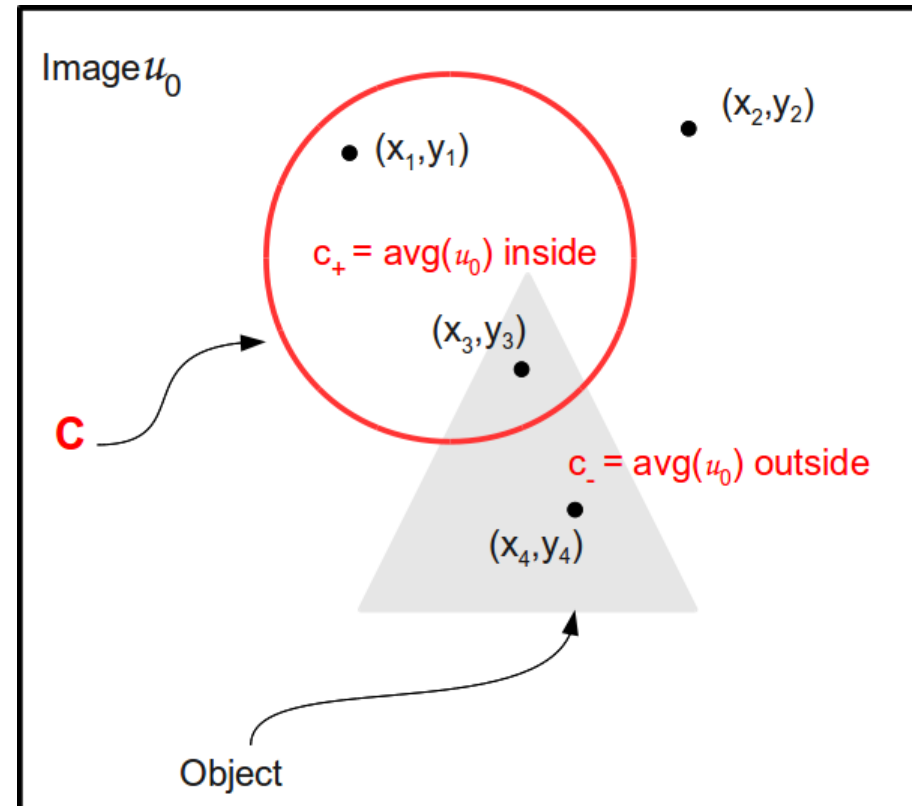
0 is true because
we want to minimize
the fitting term



Chan-Vese Formulation

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

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



Concentrate on Intersection



Interpolation Functions

$$f_{\cup} = (z_1 \cdot z_2)^{1/2}$$

$$f_{\cap} = 1 - ((1 - z_1)(1 - z_2))^{1/2}$$

Intersection of inside

Union of outside

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

General Case

$$L_1(A_1) \cap L_2(A_2) \cap \cdots \cap L_n(A_n)$$

$$F_{L_1(A_1) \cap \cdots \cap L_n(A_n)} = \mu|C| + \lambda \left[\int_{\text{inside}(C)} \left(1 - \left(\prod_{i=1}^n (1 - l_i(z_i^{\text{in}})) \right)^{1/n} \right) dx \right. \\ \left. + \int_{\text{outside}(C)} \left(\prod_{i=1}^n l_i(z_i^{\text{out}}) \right)^{1/n} dx \right].$$

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

Level Set Formulation

- Objective function

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

- We want to minimize F

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

Experiments – 2 Channels



Time Evolution showing contour C evolving to desired solution in both Channels.

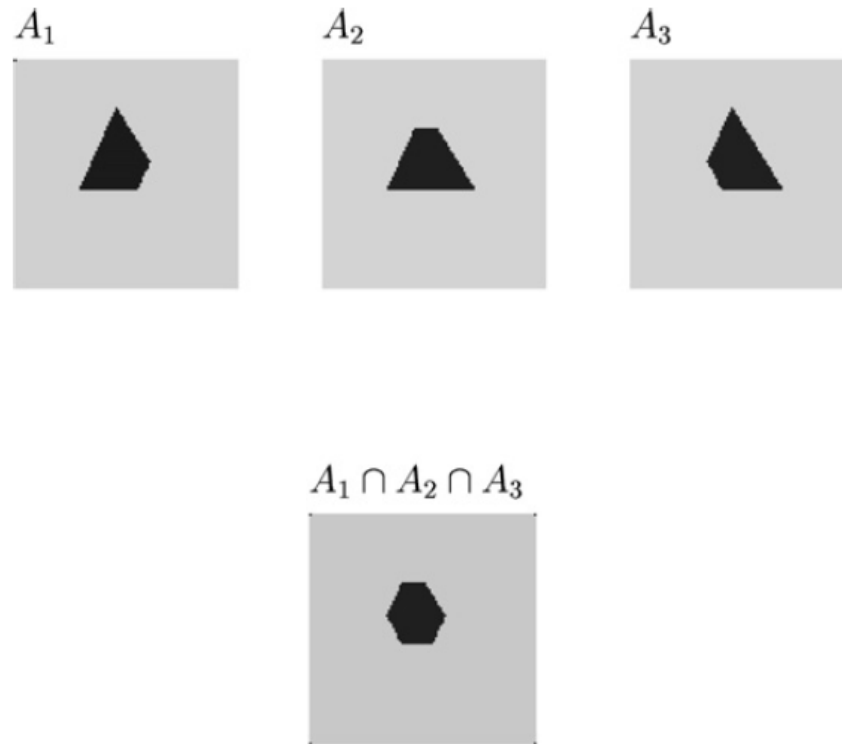
Channel A_1 for $A_1 \cap A_2$



Channel A_2 for $A_1 \cap A_2$



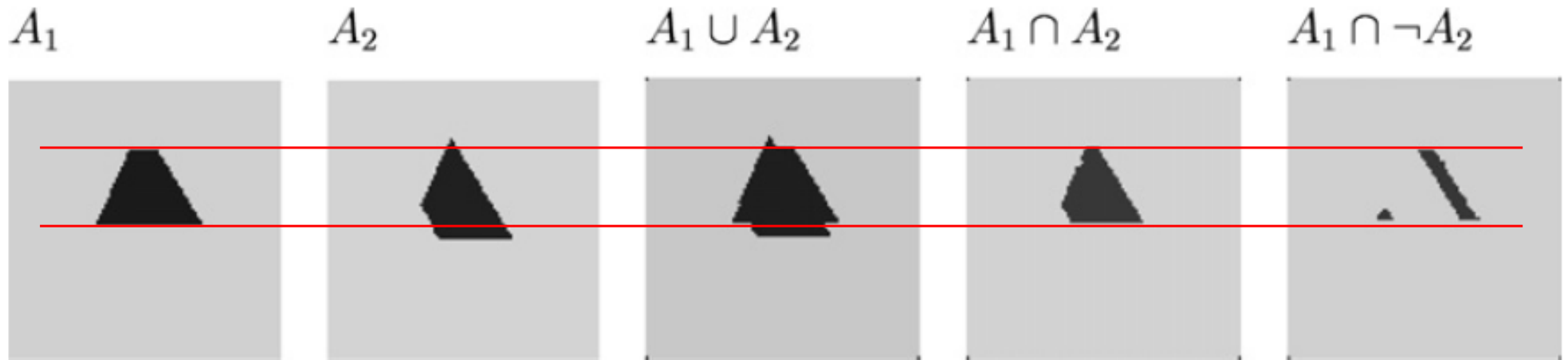
Experiments – 3 Channels



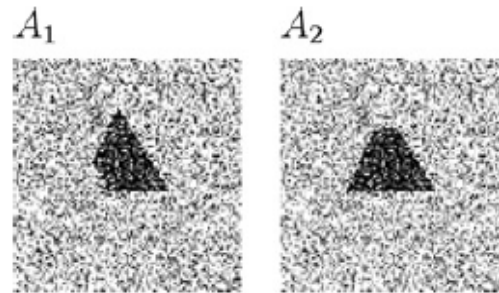
Experiments – Unregistered Images



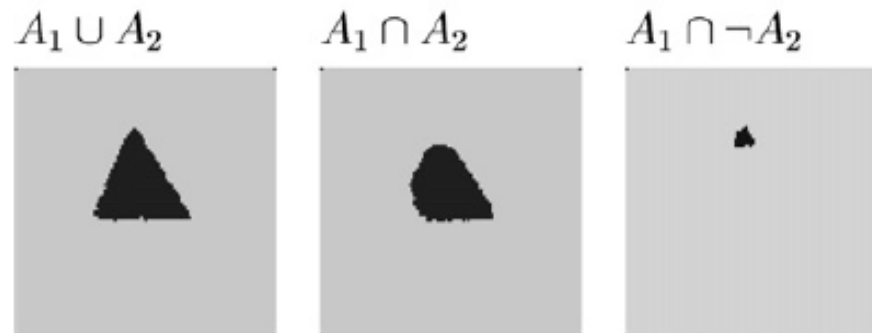
Experiments – Unregistered images



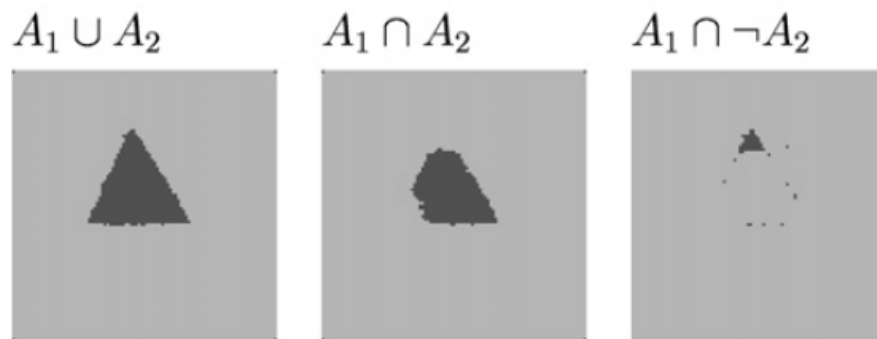
Experiments - Noise



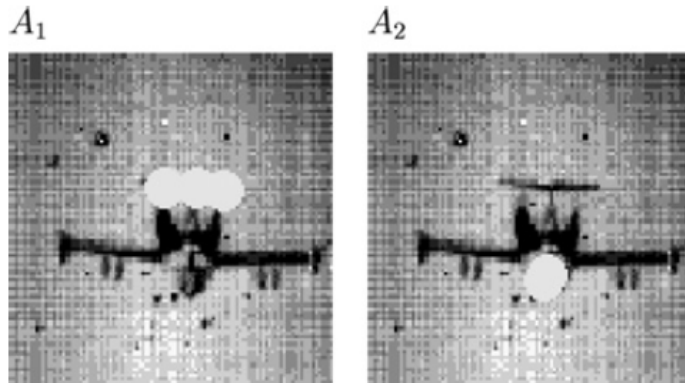
Region based logic
model



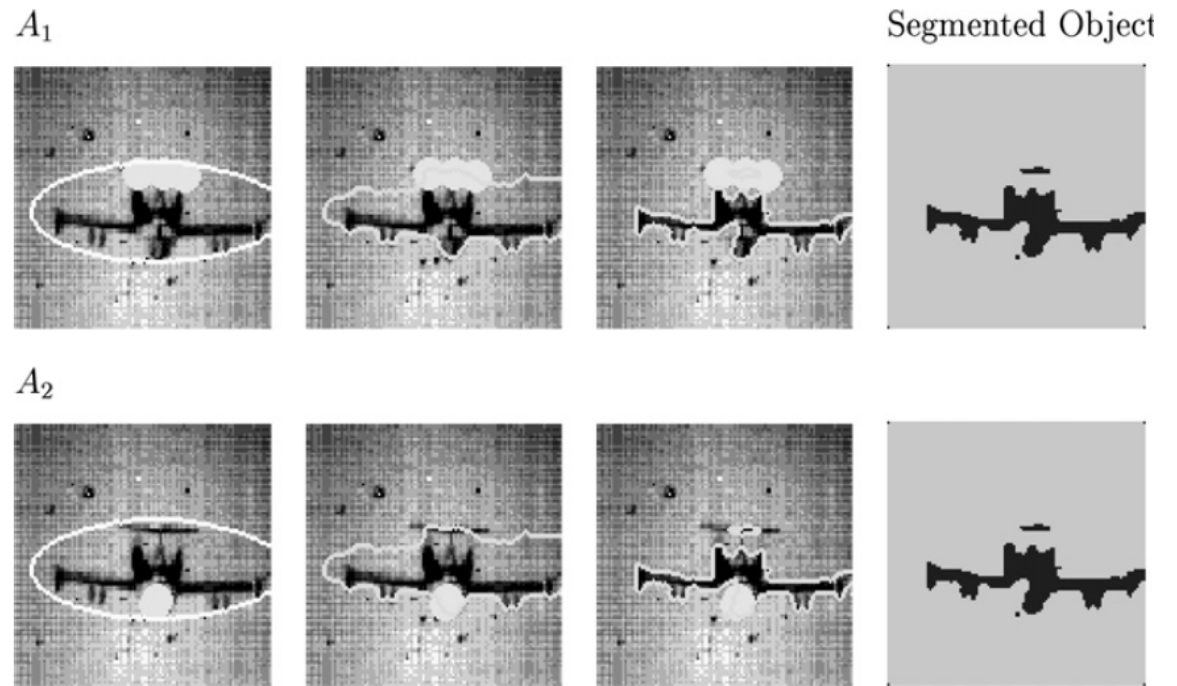
Channel by channel case



Experiments - Real Images



$$A_1 \cup A_2$$



Conclusions

- Chan-Vese model
 - Detects objects without edge information
 - Compares image intensities inside & outside the contour
 - Can detect “holes” in objects
- Sandberg-Chan logic framework
 - Can combine multiple images using different logic operations
 - Performs segmentation & logic operation at the same step

Thank you

???