Active contours without edges

Tony F. Chan & Luminita A. Vese

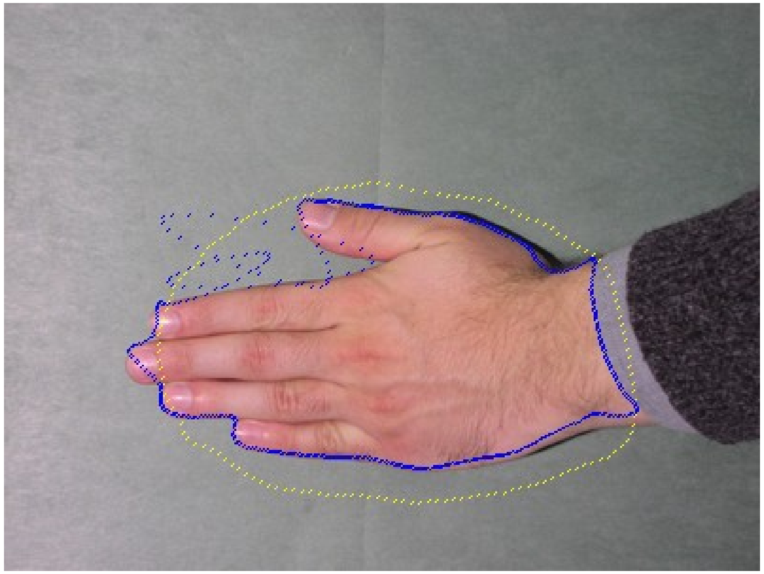
A logic framework for active contours on multi-channel images

Berta Sandberg & Tony F. Chan

Presented by: Sarah Nadi & Karim Ali

Oct. 28Th 2010 – CS 870

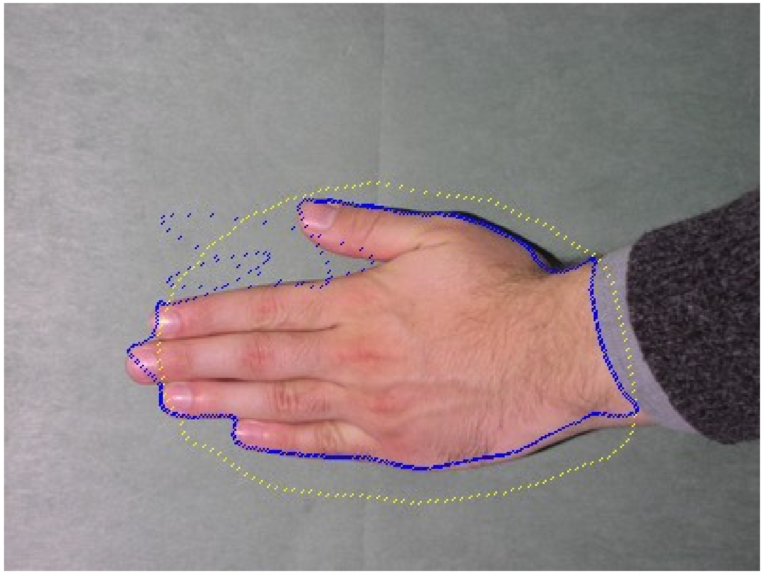
Big Picture



Detecting objects in an image

Combining multiple channels

Big Picture

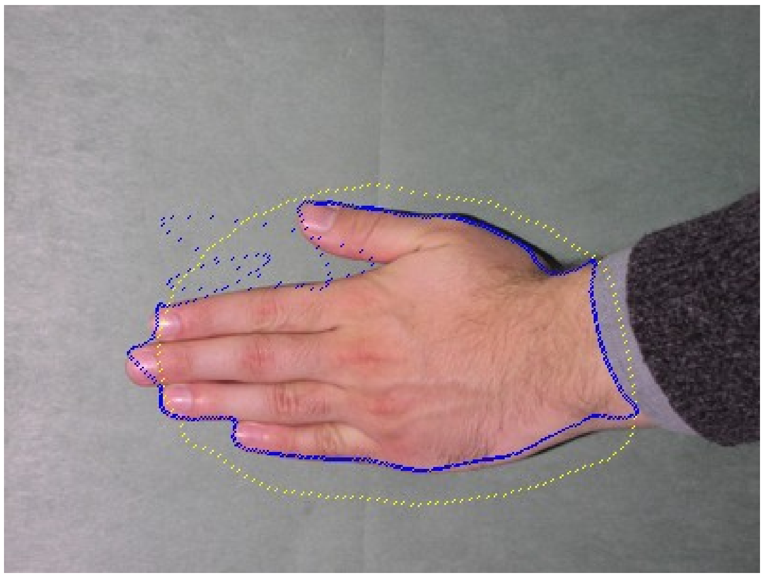


Detecting objects in an image

Combining multiple channels

Chan-Vese model

Big Picture

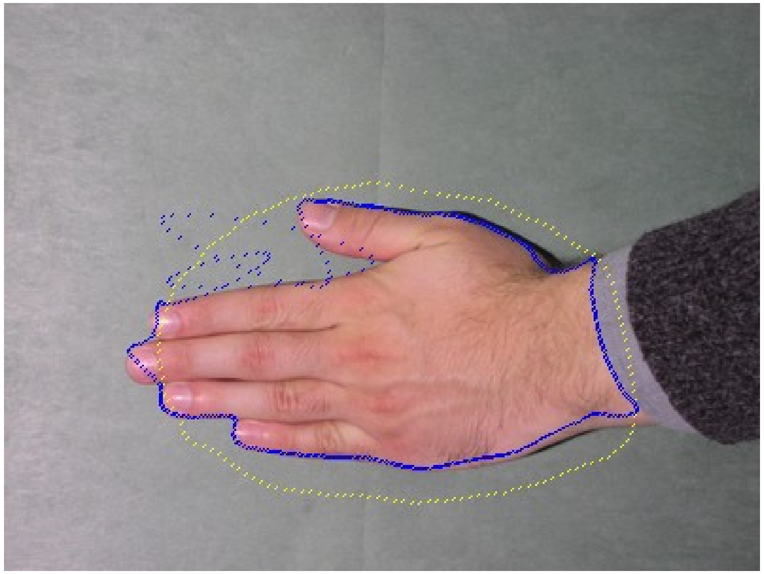


Detecting objects in an image

Combining multiple channels

Chan-Vese model

Big Picture



Detecting objects in an image

Combining multiple channels

Sandberg-Chan framework

Chan-Vese model

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

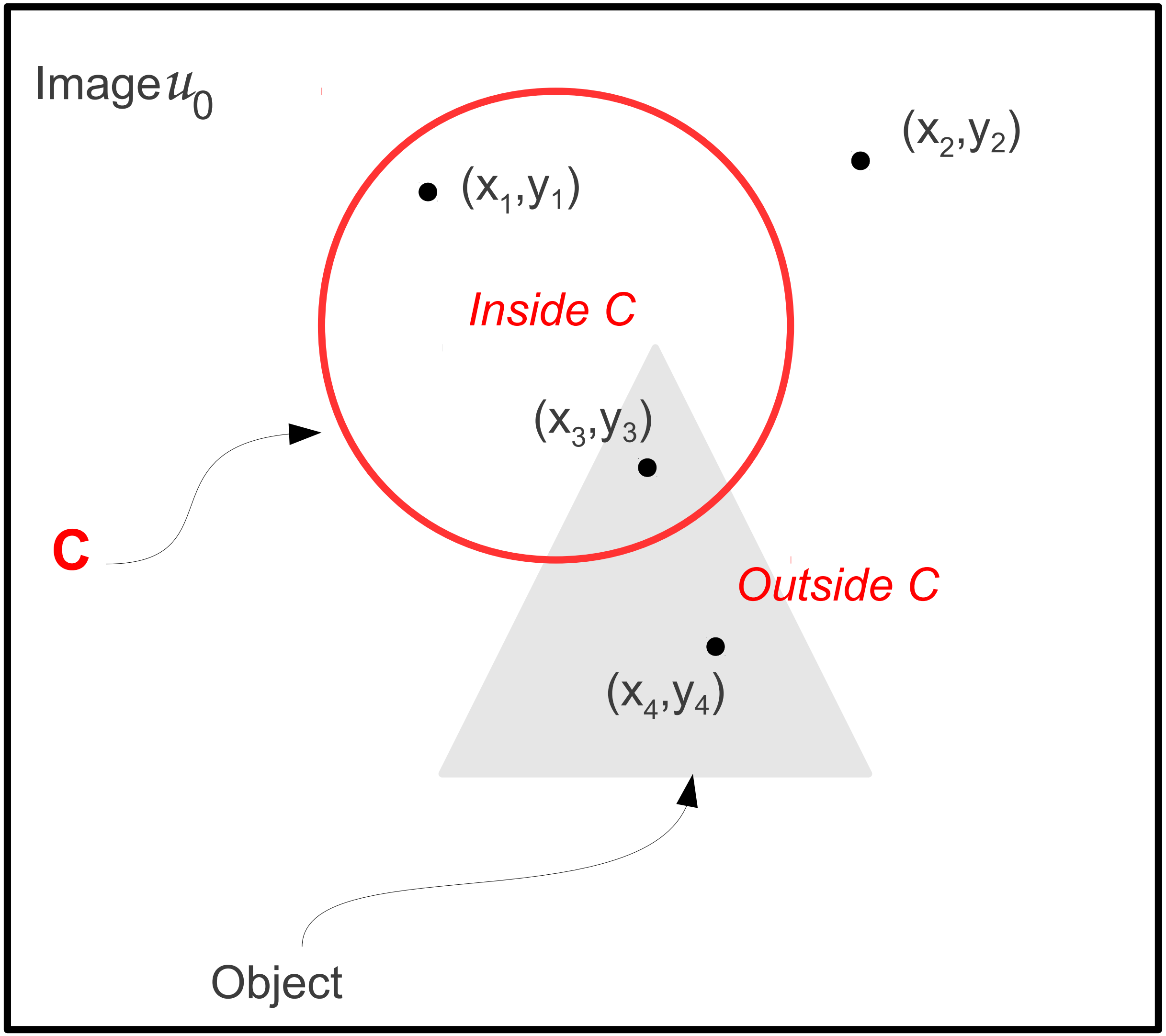
Active Contour Models

- 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

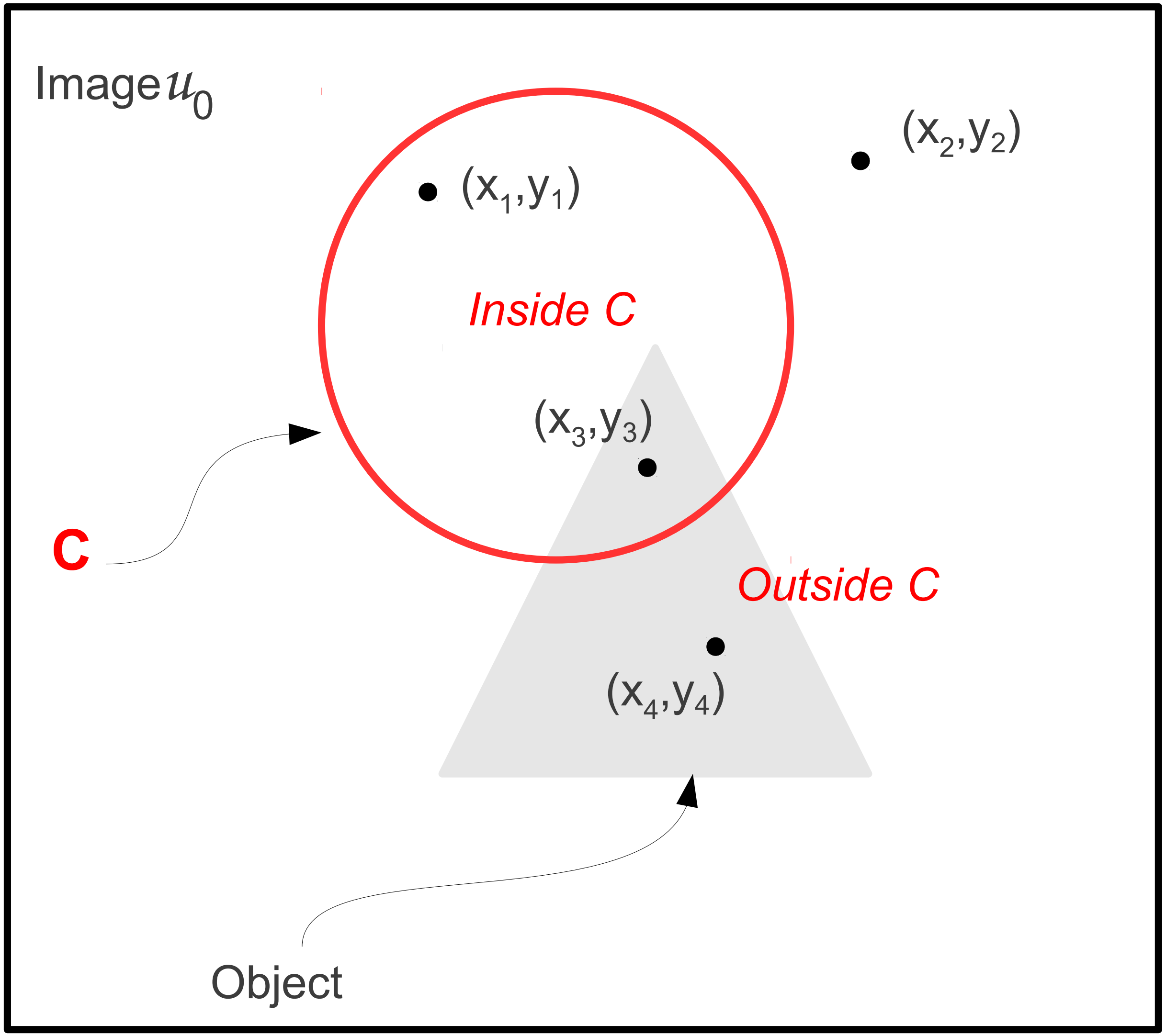
Active Contours Without Edges (Chan-Vese)

- 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



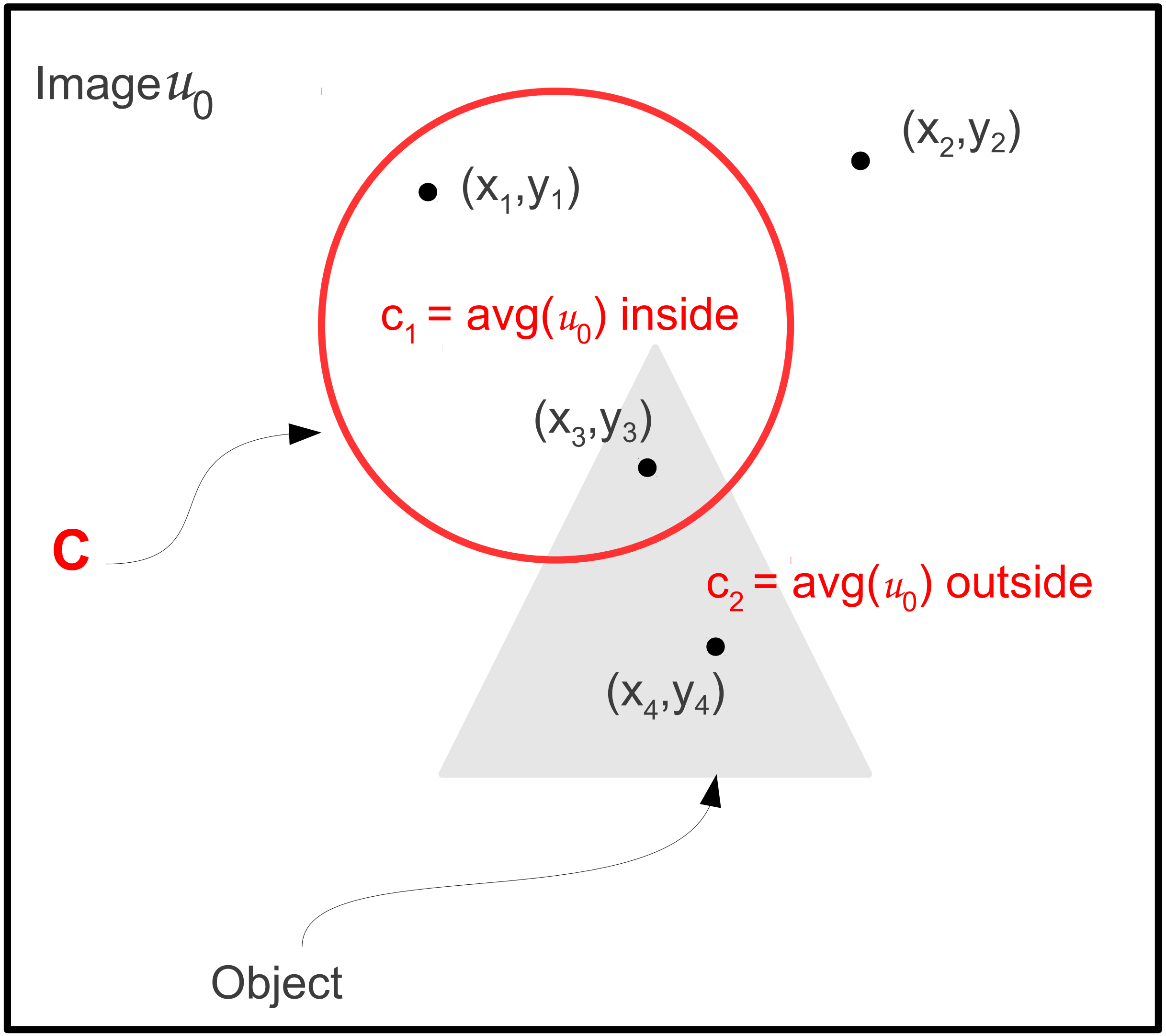
Region Based View



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

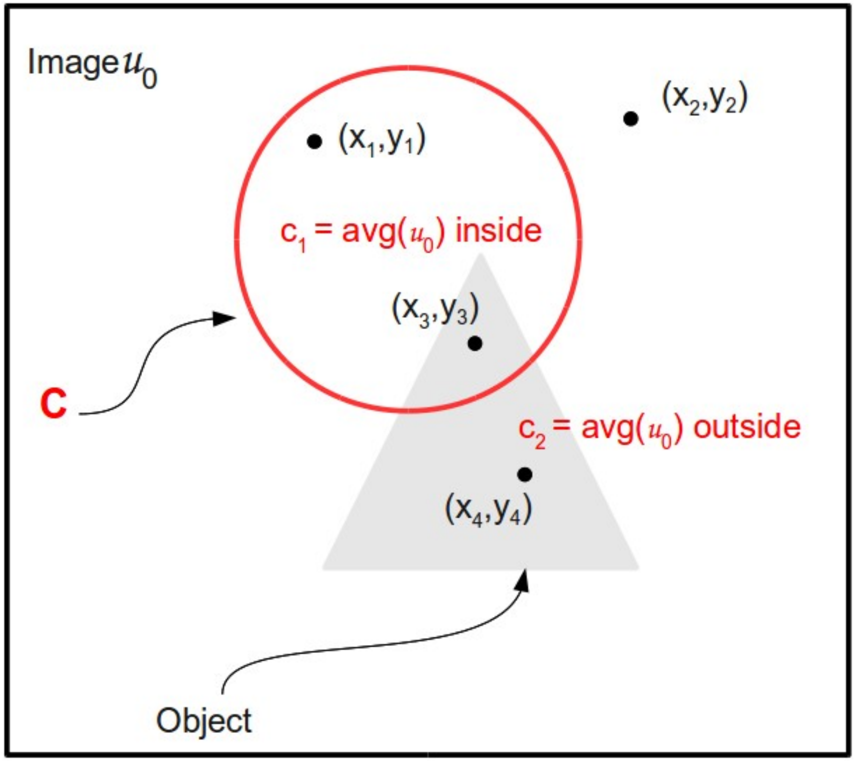
Region Based View



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:

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

Fitting outside:

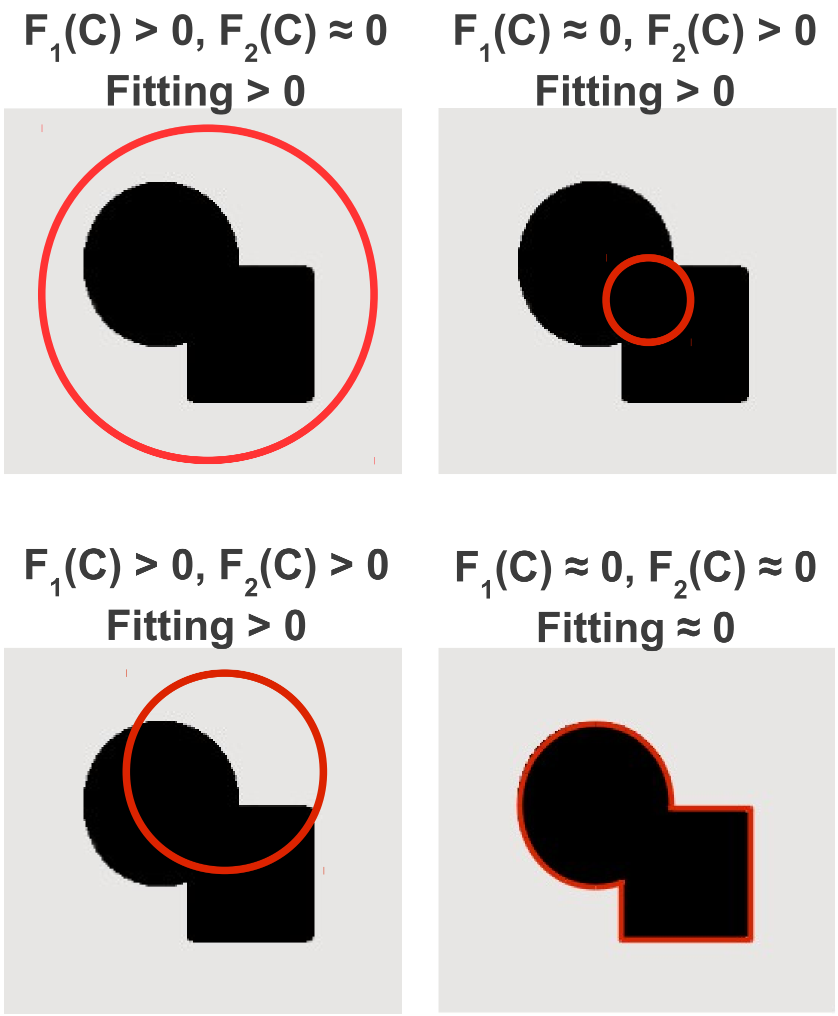
$$F_2(C) = \int_{outside(C)} |u_0(x,y) - c_2|^2 \, \mathrm{d}x \, \mathrm{d}y$$

Overall fitting of C:

$$F_1(C) + F_2(C) = \int_{inside(C)} |u_0(x,y) - c_1|^2 \, \mathrm{d}x \, \mathrm{d}y + \int_{outside(C)} |u_0(x,y) - c_2|^2 \, \mathrm{d}x \, \mathrm{d}y$$

Intuition

- C should stop evolving when C = Object
- To achieve that: Fitting ≈ 0



Energy Function

$$F(c_1, c_2, C) = \lambda_1 \int_{inside(C)} |u_0(x,y) - c_1|^2 \mathrm{d}x \, \mathrm{d}y + \lambda_2 \int_{outside(C)} |u_0(x,y) - c_2|^2 \mathrm{d}x \, \mathrm{d}y$$

Energy Function

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

Energy Function

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

- Look at minimization problem:

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

Level Set Formulation of Energy Function

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

Level Set Formulation of Energy Function

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

Level Set Formulation of Energy Function

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

Level Set Formulation of Energy Function

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

Level Set Formulation of Energy Function

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

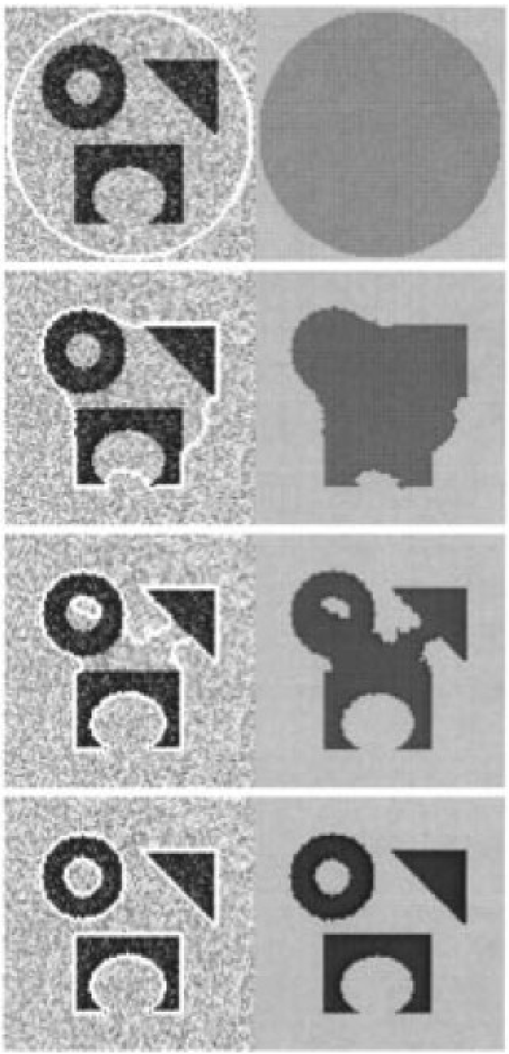
Chan-Vese Technique

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

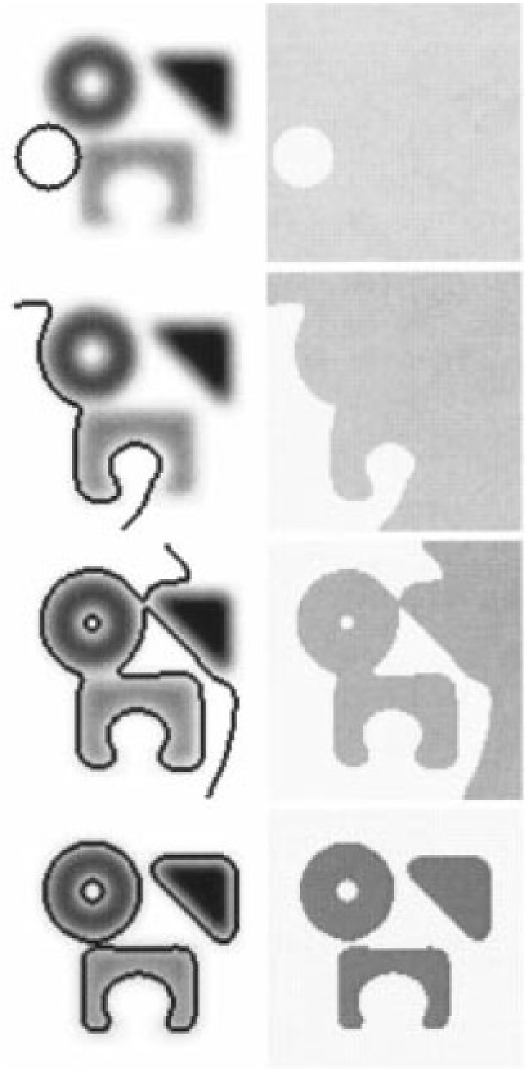
- Initialize Φ^0 by Φ_0, n = 0
- Compute c1(Φ^n) and c2(Φ^n)
- Solve the PDE to obtain Φn+1
- Reinitialize Φ locally to the signed distance function to the curve (optional)
- Check whether the solution is stationary. If not n = n + 1 and repeat

Experimental Results

Noisy Image

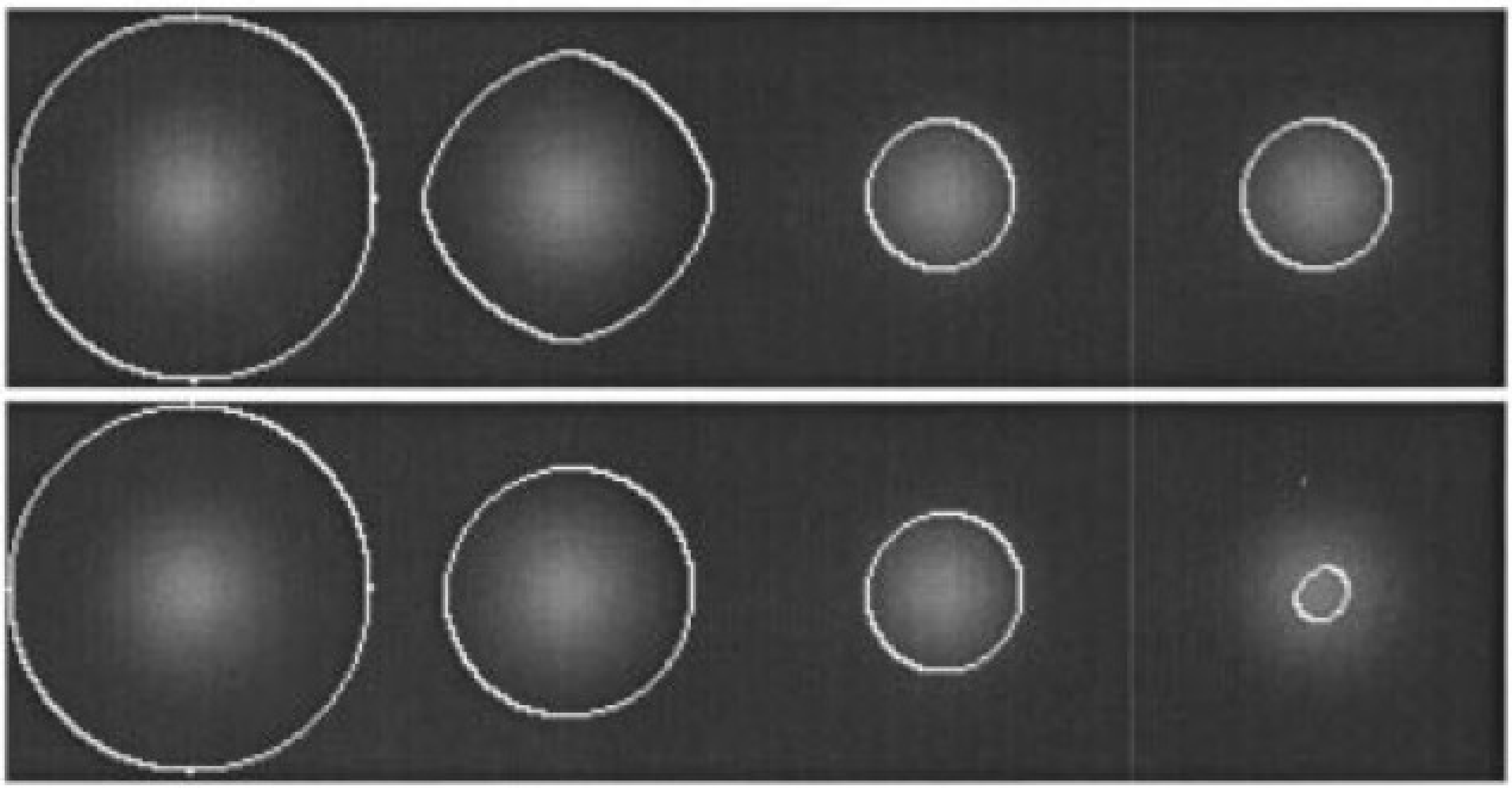


Blurry Image



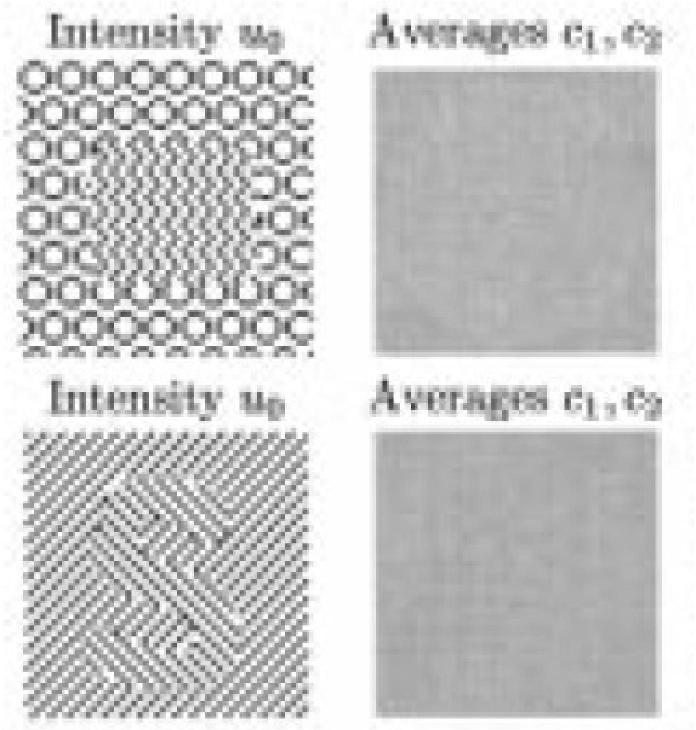
Comparing Models

Chan-Vese

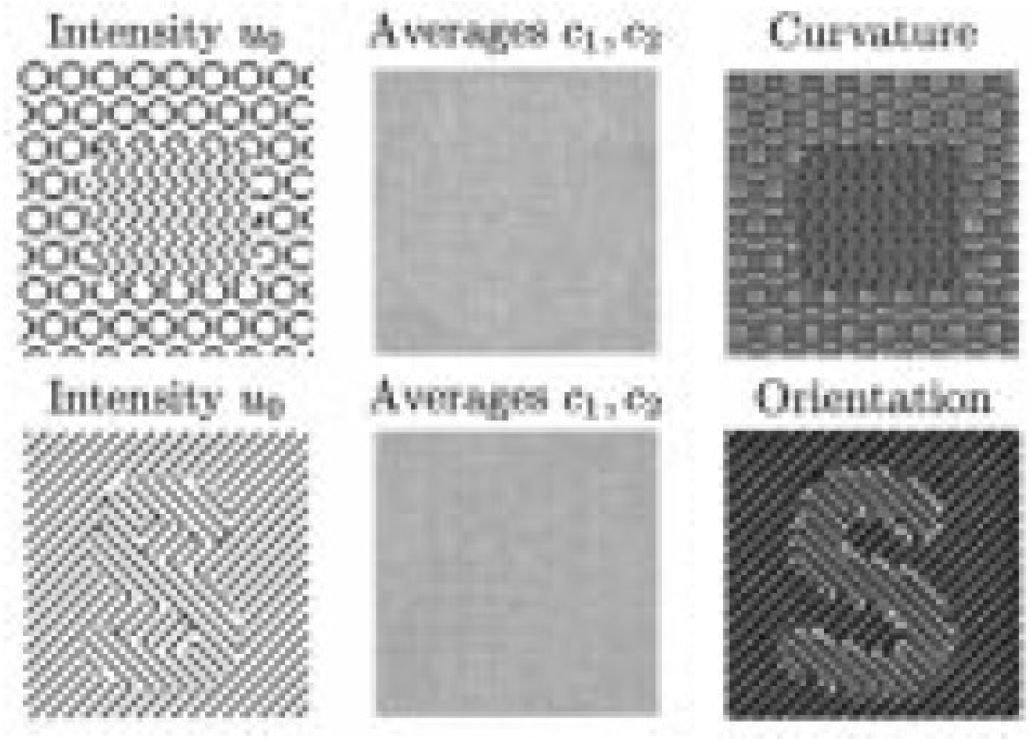


Gradient based active contour

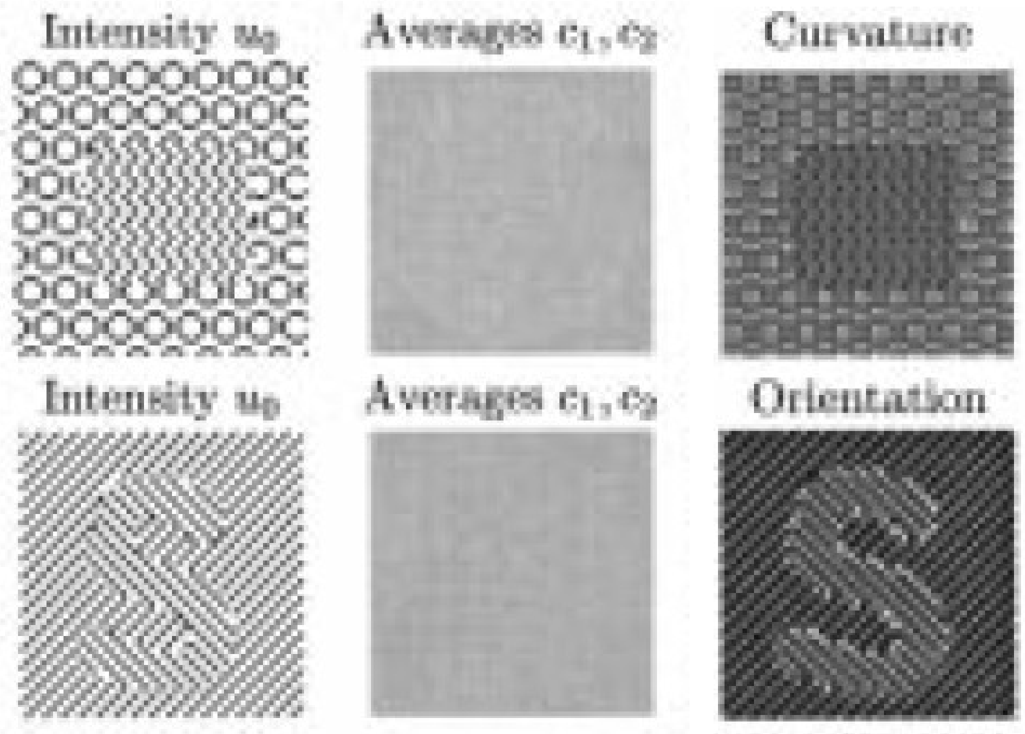
Limitations



Limitations



Limitations

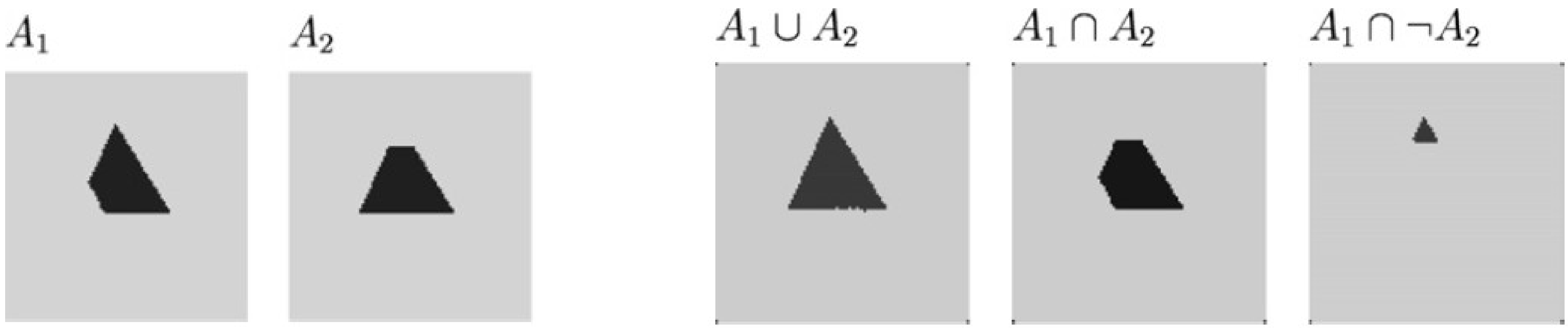


Or implement general case of Mumford-Shah function

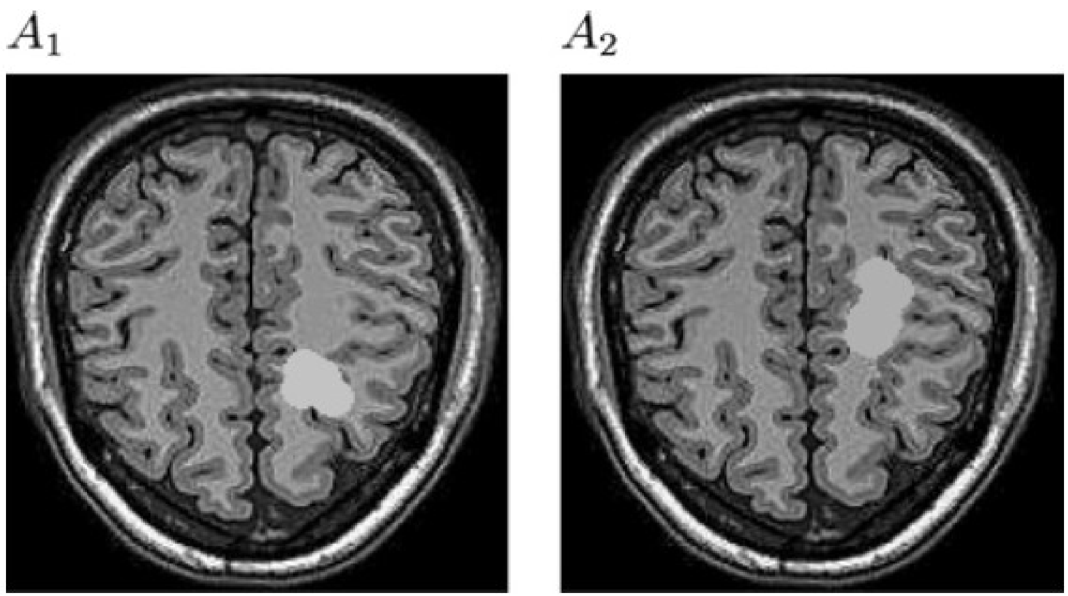
Applying the Chan-Vese Model to Multi-channel Images

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**

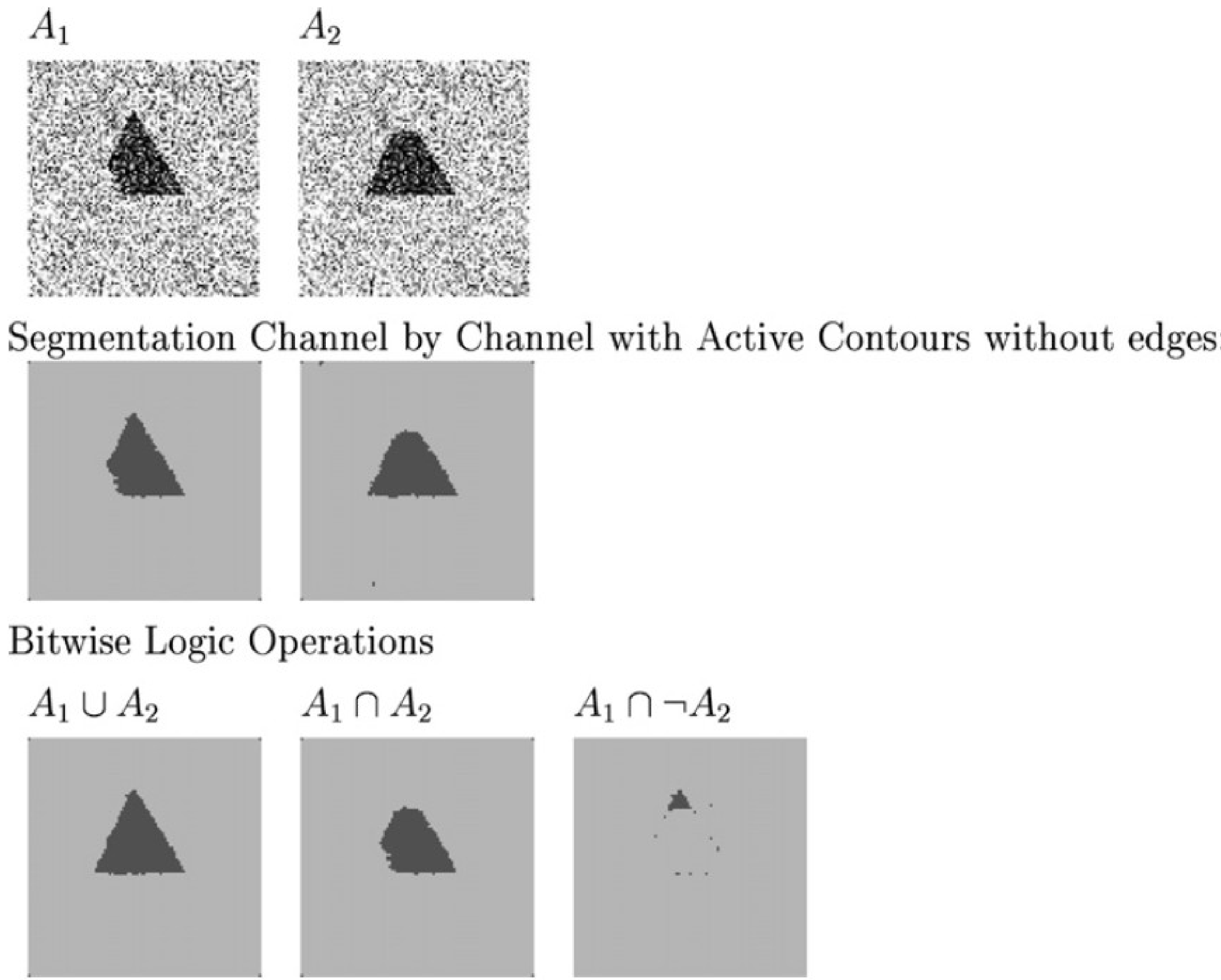
$A_1 \cap \neg A_2$ **Tumor shrinking**

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

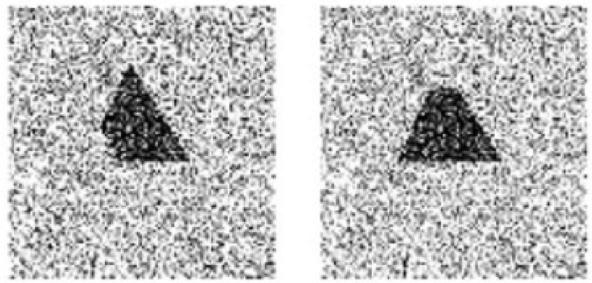
Technique 1



A_1
A_2
Segmentation Channel by Channel with Active Contours without edges:
Bitwise Logic Operations
$A_1 \cup A_2$
$A_1 \cap A_2$
$A_1 \cap \neg A_2$

Technique 1

A_1 A_2

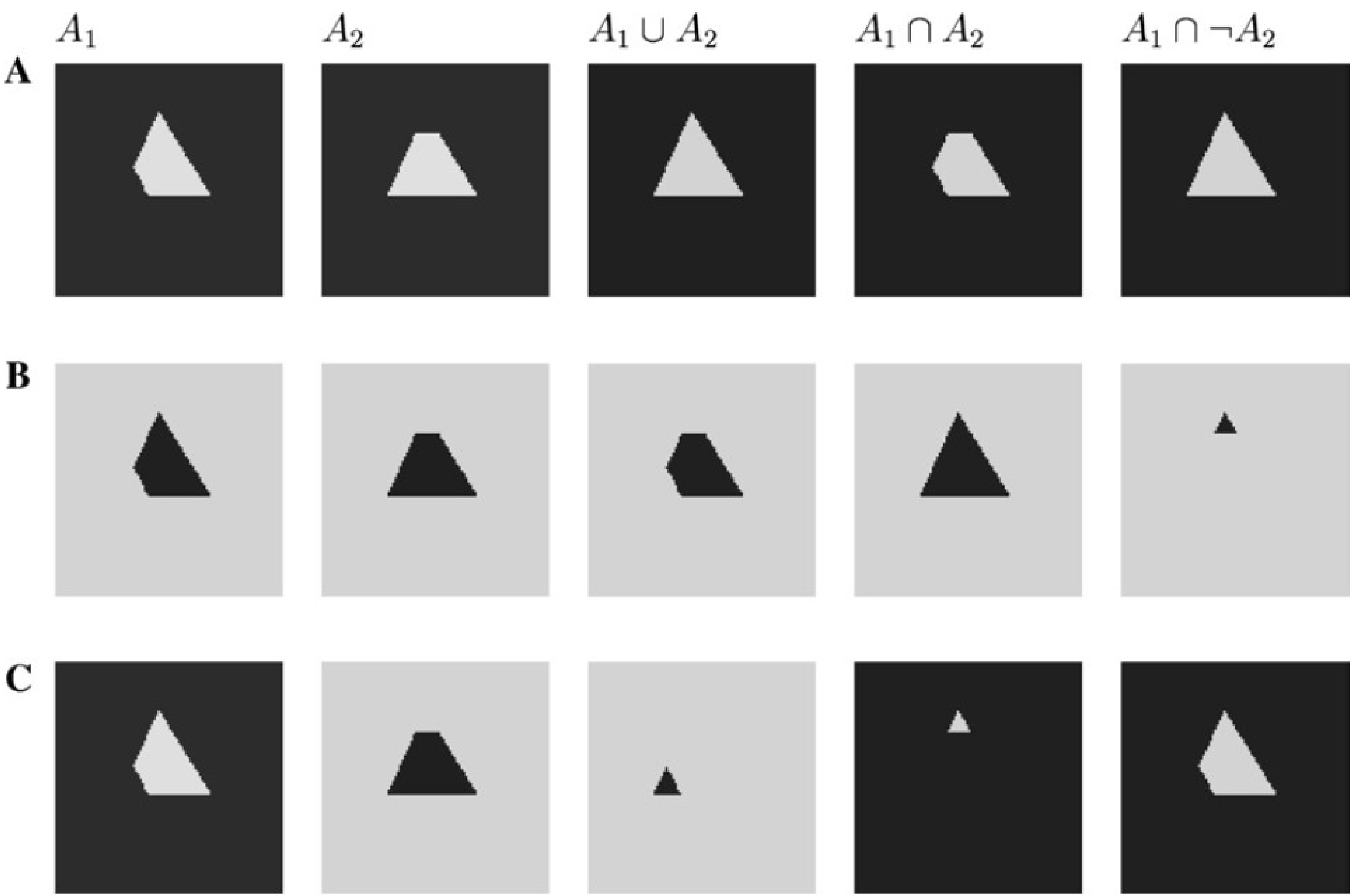


Segmentation Channel by Channel with Active Contours without edges:

Bitwise Logic Operations

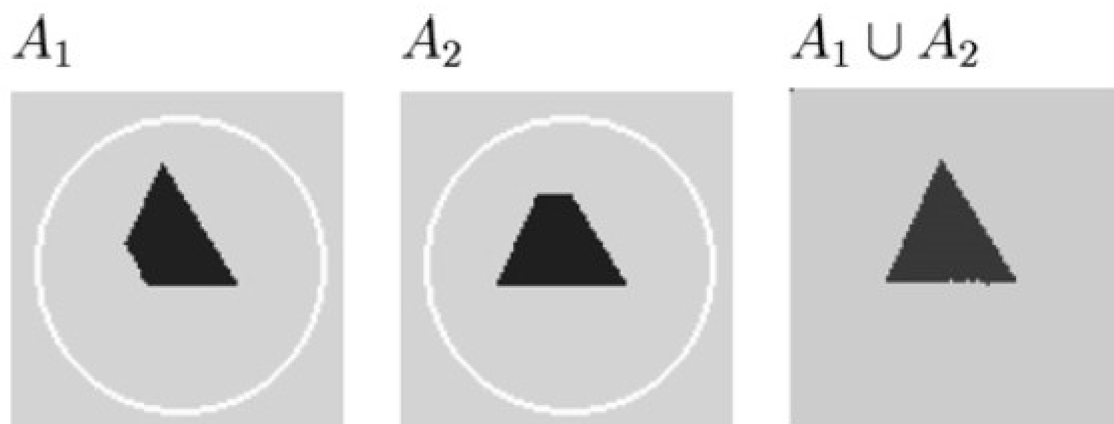
$A_1 \cup A_2$ $A_1 \cap A_2$ $A_1 \cap \neg A_2$

Technique 2

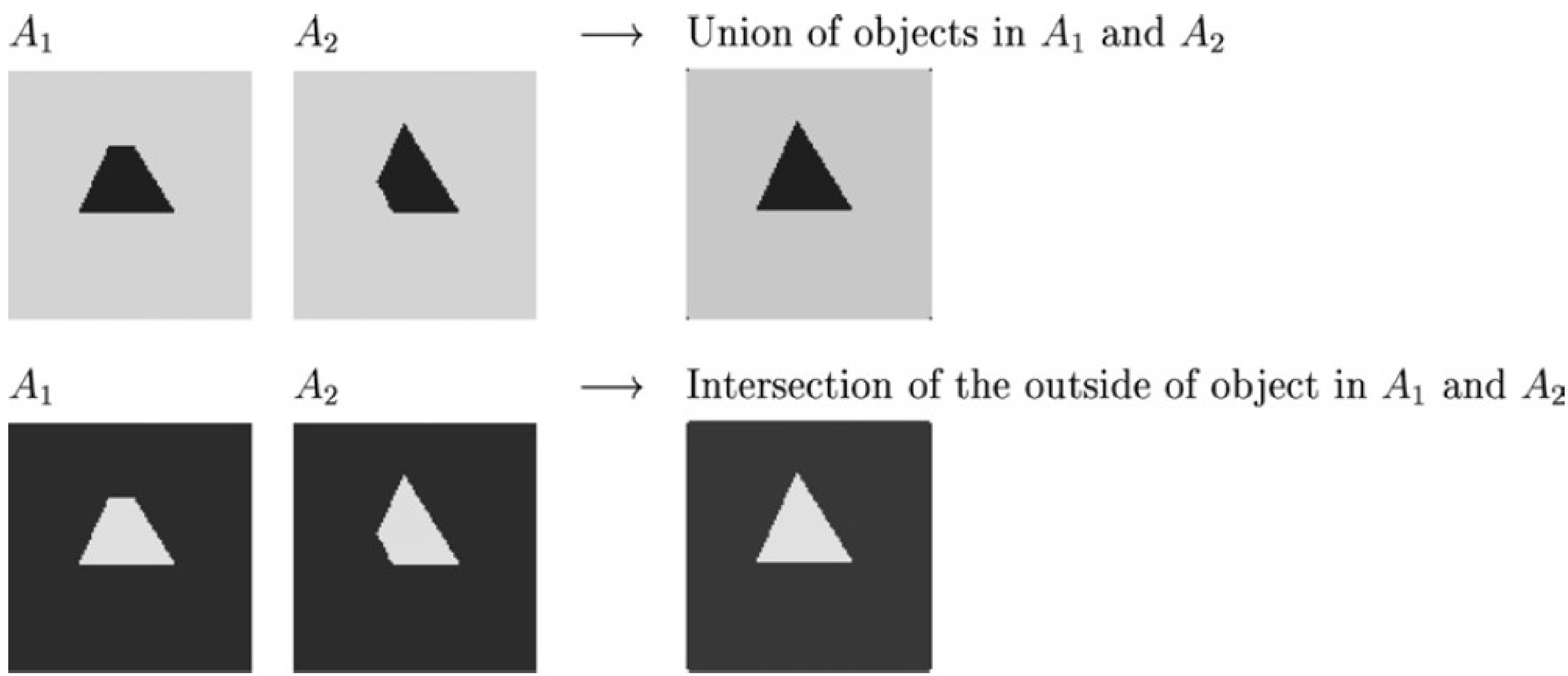


Proposed Solution

- 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

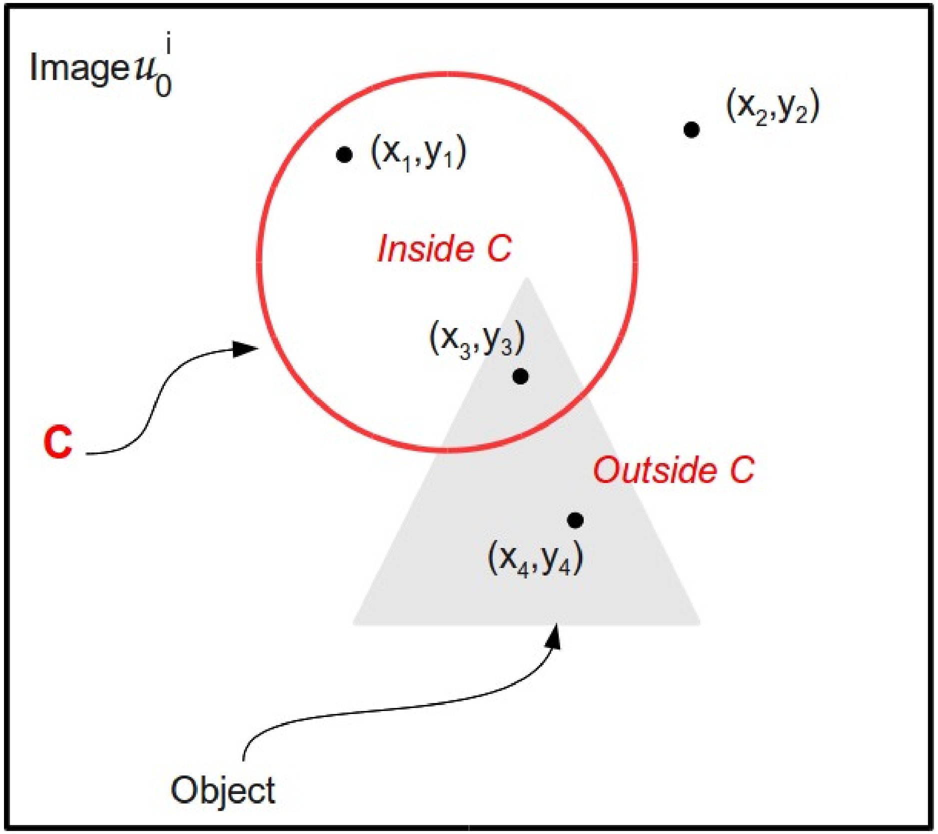


Logic Variables

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

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

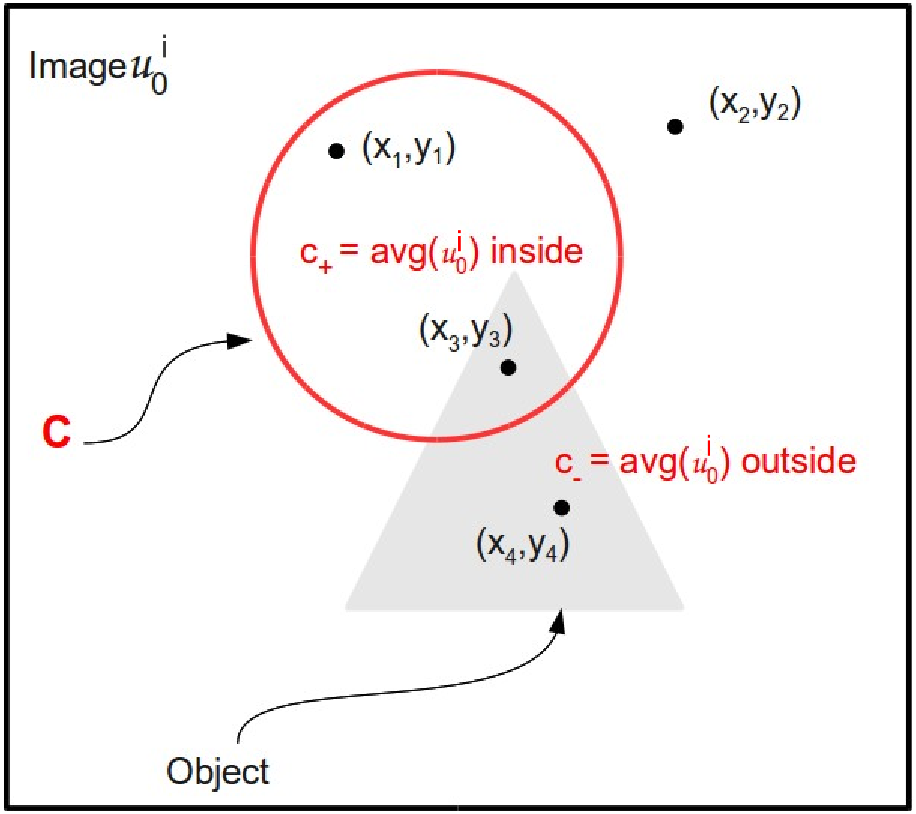
- *Note:*
  0 is true because we want to minimize the fitting term



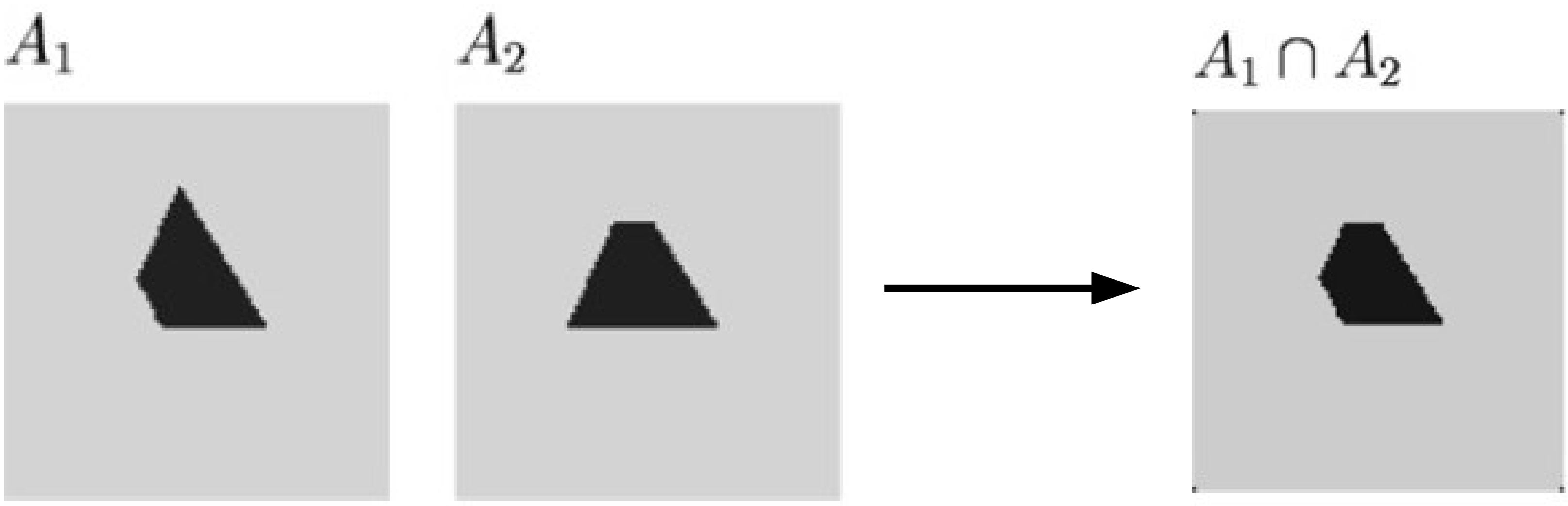
Chan-Vese Formulation

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

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



Intersection



Interpolation Functions

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

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

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

General Case

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

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

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

Level Set Formulation

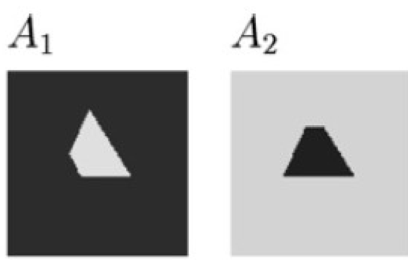
- Objective function

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

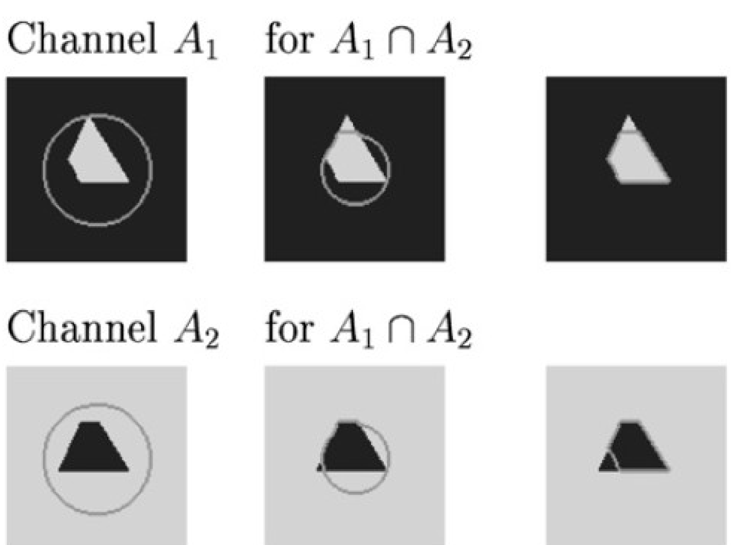
- We want to minimize F

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

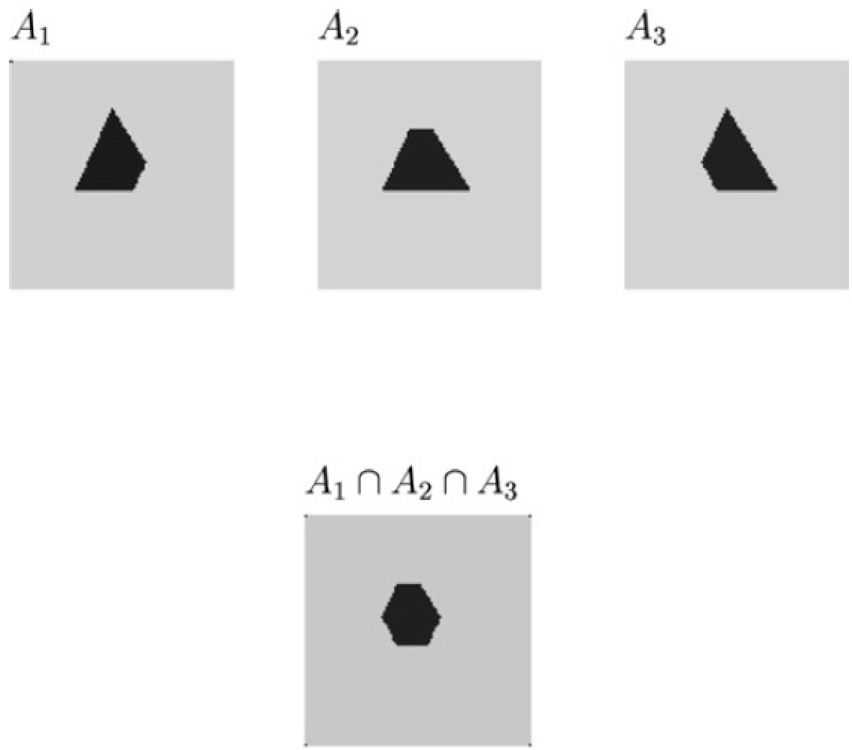
Experiments – 2 Channels



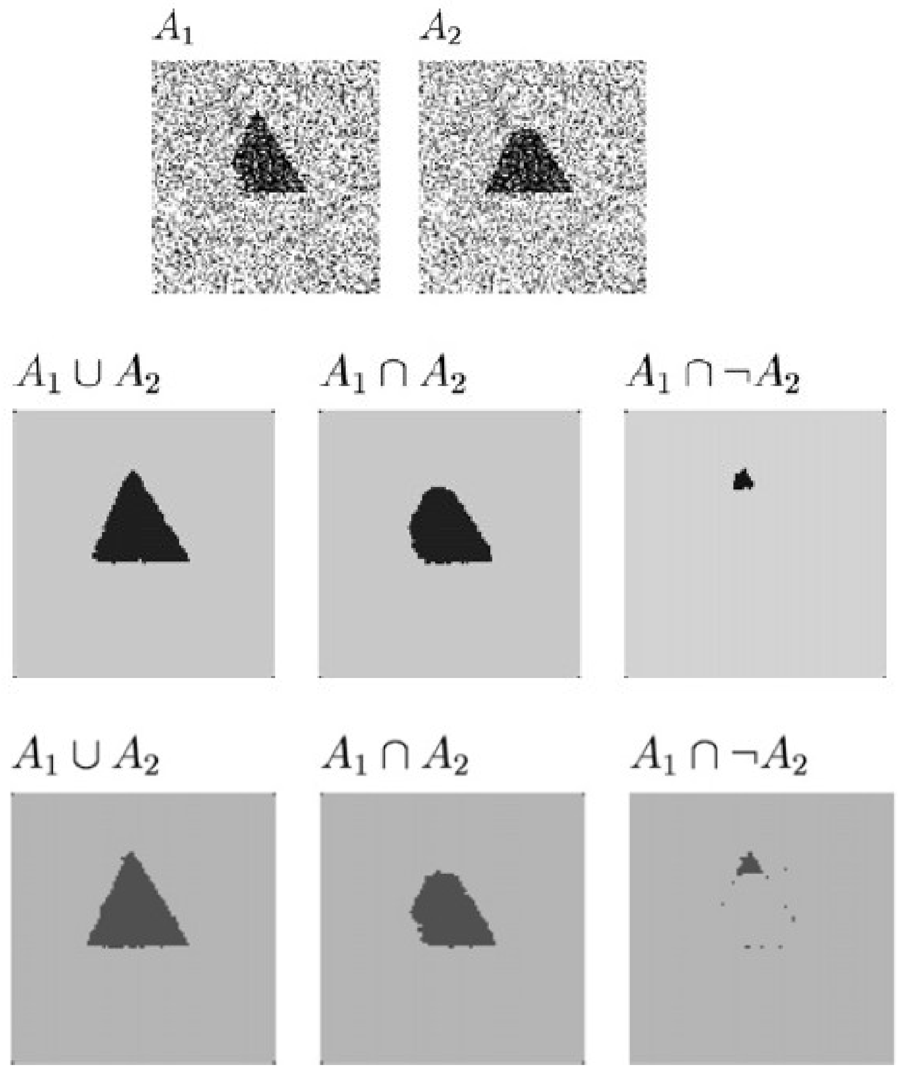
Time Evolution showing contour C evolving to desired solution in both Channels.



Experiments – 3 Channels



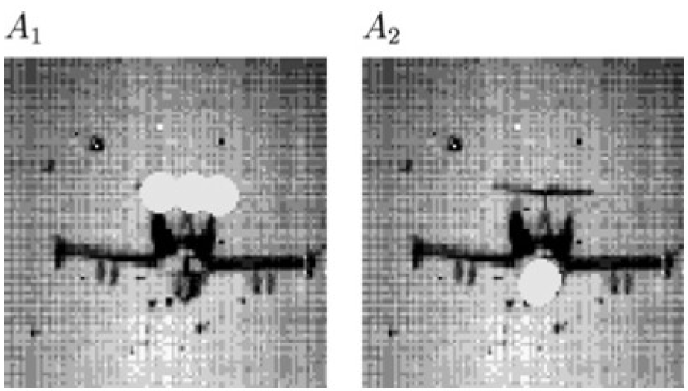
Experiments - Noise



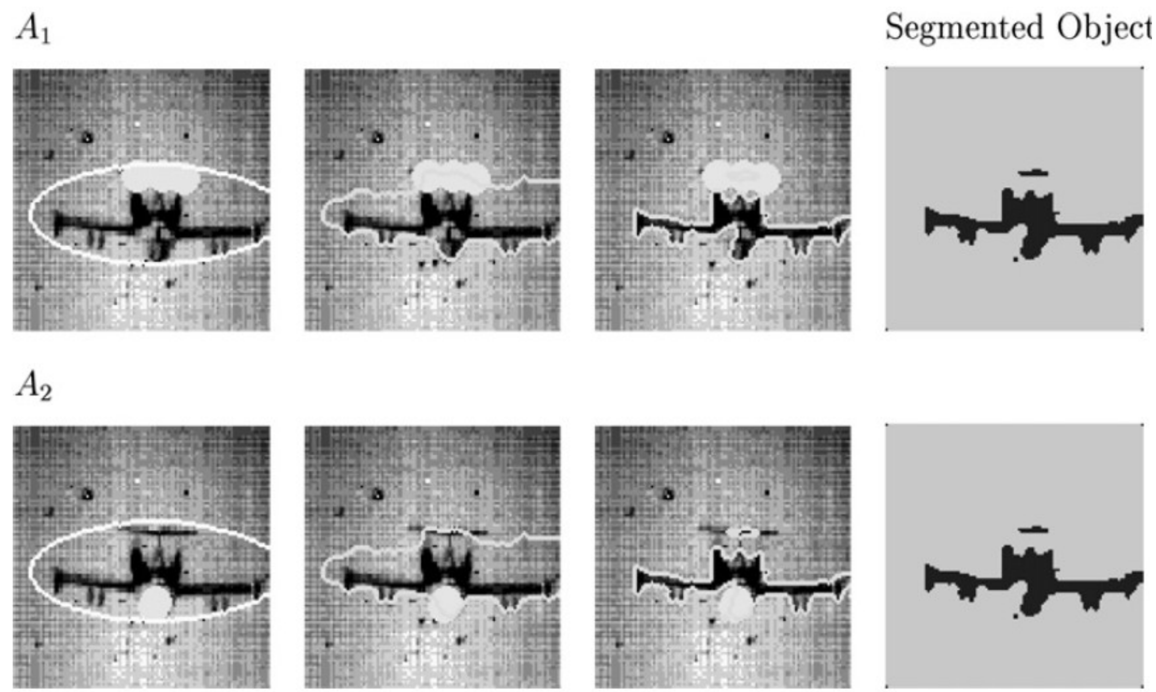
Region based logic model

Channel by channel case

Experiments - Real Images



$A_1 \cup A_2$



Conclusions

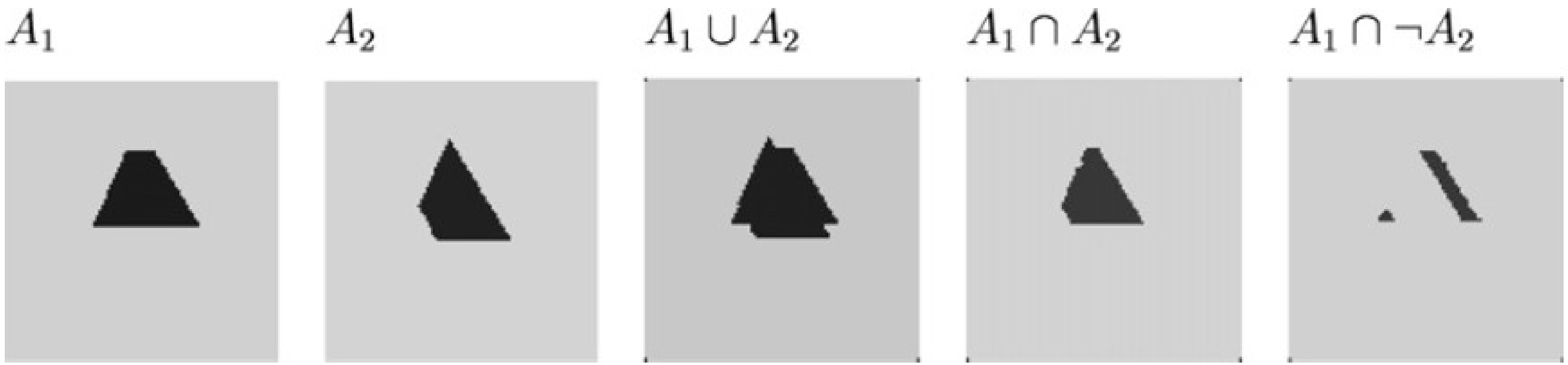
- Chan-Vese model
  - Detects objects without edge information
  - Compares image intensities inside & outside the contour
  - Can detect “holes” in objects

- Sandberg-Chan logic framework
  - Can combine multiple images using different logic operations
  - Performs segmentation & logic operation simultaneously

Thank you

???

Experiments – Unregistered Images



Experiments – Unregistered images