Active Contours without edges for Vector-Valued Images

Shane Zamora
Saturday, March 7th, 2009
CS 216 – Winter 2009

Previous Work

T. Chan, B.Y. Sandberg, and L. Vese. *Active contours without edges for vector-valued images*. Journal of Visual Communication and Image Representation, 11:130–141, 2000.

Builds off of:

T. Chan, and L. Vese, *Active Contours Without Edges*. IEEE Trans. Image Process. [UCLACAMReport 98-53]

Most of the paper describes the scalar technique and the end describes the vector valued case.

Outline

Problem description

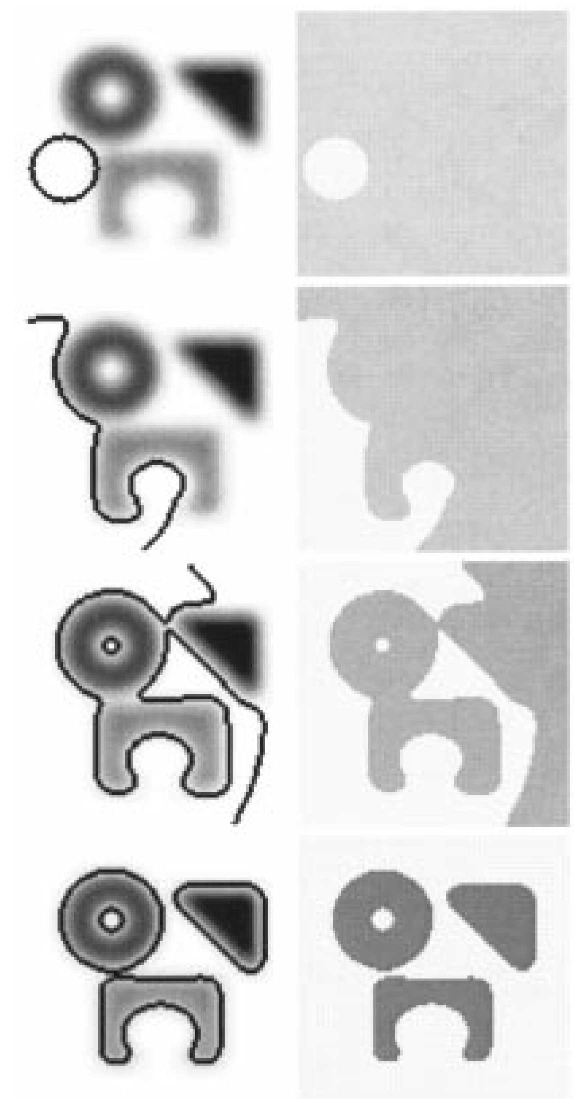
Active contours scalar case

Active contours vector-valued case

Problem Description

Premise: Evolve a curve using level sets to detect an object within a given image u_0 .

Accomplished by minimizing an energy function that attracts the contours to the object boundary.

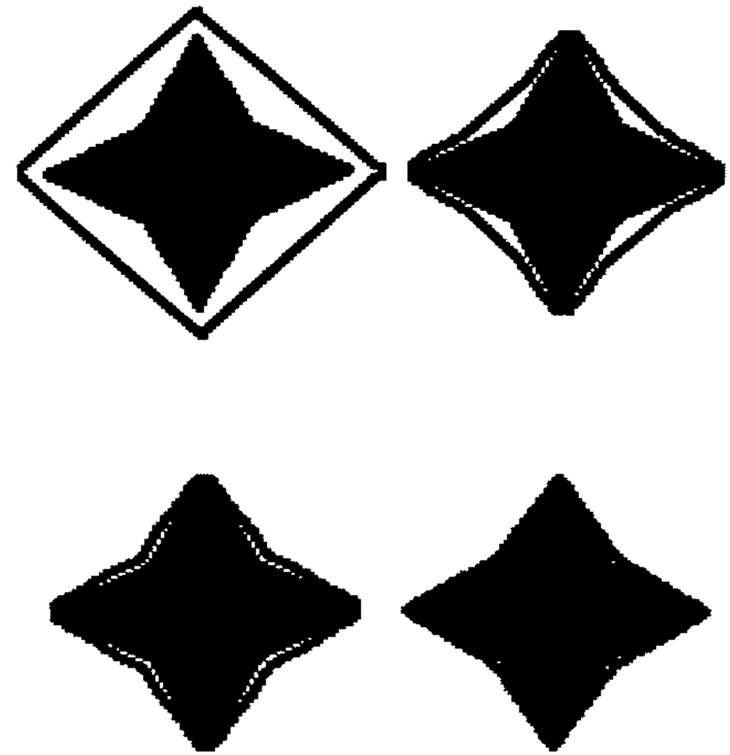


Problem Description (Continued)

Previous methods look at the gradient of the image and attempt to attract the curve to this gradient.

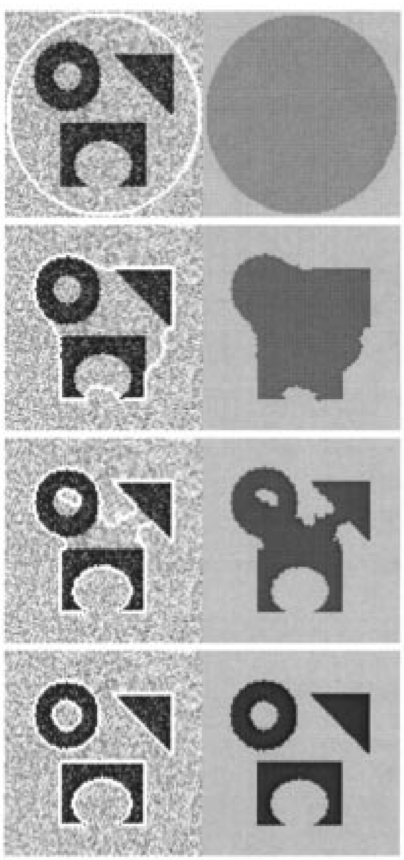
Drawback is that you require a steep gradient in order to stop the curve evolution.

Cannot handle noise or figures with blurred edges.



Problem Description (Continued)

Active Contours Without Edges:

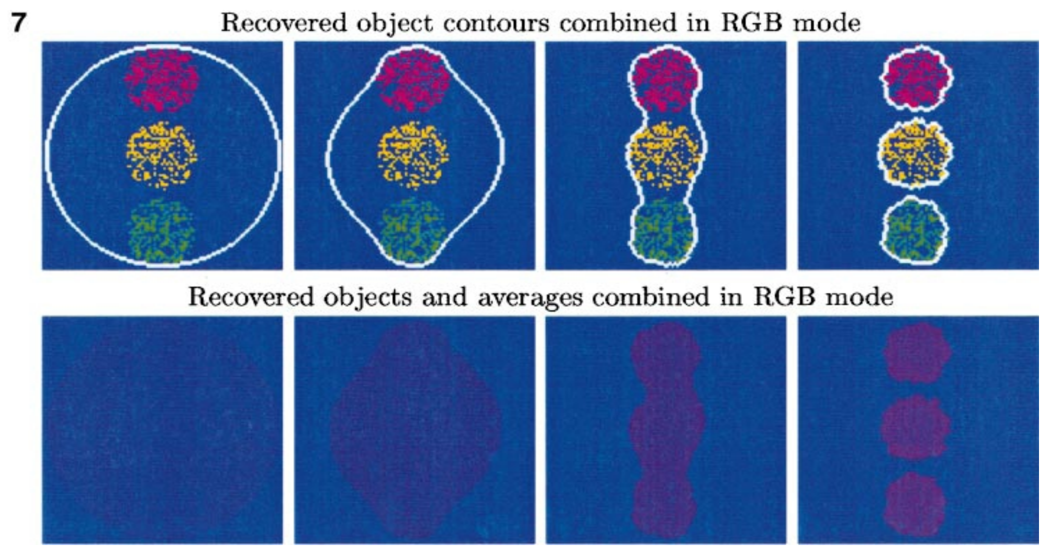


Evolve the curve by, essentially, maximizing the contrast between the inside and outside of the contour.

This method detects objects with our without hard edges, continuous boundaries, and noise. It also detects internal contours and handles arbitrary initial curves.

Problem Description (Continued)

Active Contours Without Edges for Value Vectored Images:



Modifies the previous method for images with multiple channels. Most natural example – 3 channel color images with red, green, and blue channels.

Outline

Problem description

Active contours scalar case

Active contours vector-valued case

Energy Function to Minimize

We'll want to maximize the contrast between the inside and outside of the contour.

To do this, we will want to homogenize as much as possible the subsets of the image that are inside and outside of the curve.

Energy Function to Minimize (Cont)

Need a function that describes the homogeneity of a given area that is small when the area is homogeneous and large when it is not.

(So we can use this for minimization)

Energy Function to Minimize (Cont)

Define some variables:

Ω	A subset of $\mathbb{R}^2$ containing the image u_0 being worked on.
C	The curve being evolved in Ω.
ω	The area of Ω inside the curve of which C is the boundary, i.e. $\omega \subset \Omega$ and $C = \partial\omega$

So inside(C) is the region ω and outside(C) is the region $\Omega \setminus \overline{\omega}$

Energy Function to Minimize (Cont)

Minimization function:

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

Summation of each pixel's squared difference from the average inside/outside of C.

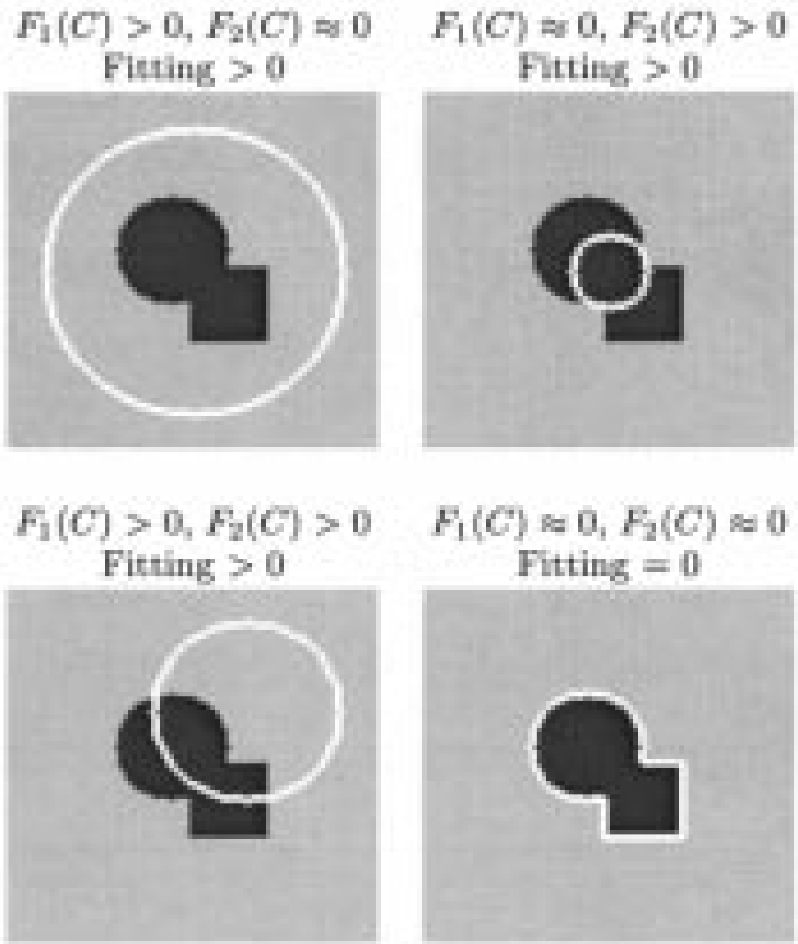
c_1 = average intensity inside C

c_2 = average intensity outside C

Energy Function to Minimize (Cont)

$$F_1(C)+F_2(C)=\int_{inside(C)} |u_0(x,y)-c_1|^2\,dx\,dy$$

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



The bottom right figure minimizes the function and finds the contour of the object!

Energy Function to Minimize (Cont)

"Regularizing" terms can be added, as well as coefficients to control the impact of each term.

$$F_1(C) + F_2(C) = \mu \cdot Length(C) + \nu \cdot Area(inside(C))$$

$$+ \lambda_1 \int_{inside(C)} |u_0(x, y) - c_1|^2 \, dx\, dy$$

$$+ \lambda_2 \int_{outside(C)} |u_0(x, y) - c_2|^2 \, dx\, dy$$

$$\mu \geq 0, \nu \geq 0, \lambda_{1,} \lambda_2 > 0$$

Most of the time, $\lambda_1 = \lambda_2 = 1, \nu = 0$ is used.

Relation to Mumford Shah

The stopping function that homes in on the final contour of C, C_0, is based on the Mumford-Shah functional:

$$F^{MS}(u, C) = \mu \cdot Length(C) + \lambda \int_{\Omega} |u_0(x, y) - u(x, y)|^2 \, dx\, dy + \int_{\Omega \setminus C} |\nabla u(x, y)|^2 \, dx\, dy$$

Relation to Mumford Shah (Cont.)

Mumford Shah aims to segment an image into regions R_i with smooth regions (a constant intensity c_i) and sharp boundaries. (like a cartoon)

Instead, we want to segment the image into two regions:

$$u = \begin{cases} average(u_0)\, inside\, C \\ average(u_0)\, outside\, C \end{cases}$$

Making it use Level Sets

So that we can integrate over the entire image, we use the Heaviside function and Dirac measure:

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

$$\delta_0(z) = H'(z)$$

Using this, we can mask the area ω and invert the mask using $1 - H(z)$.

For z , we'll use the value from the level set function ϕ .

Making it use Level Sets (Continued)

$F_1(C)+F_2(C)=$

$$\mu \int_{\Omega} \delta_0(\phi(x,y))|\nabla \phi(x,y)|\,dx\,dy$$

$$+\nu \int_{\Omega} H(\phi(x,u))\,dx\,dy$$

$$+\lambda_1 \int_{\Omega} |u_0(x,y)-c_1|^2 H(\phi(x,y))\,dx\,dy$$

$$+\lambda_2 \int_{\Omega} |u_0(x,y)-c_2|^2 (1-H(\phi(x,y)))\,dx\,dy$$

Making it use Level Sets (Continued)

$$c_1(\phi) = \frac{\int_\Omega u_0(x, y) H(\phi(x, y)) \, dx \, dy}{\int_\Omega H(\phi(x, y)) \, dx \, dy}$$

$$c_2(\phi) = \frac{\int_\Omega u_0(x, y)(1 - H(\phi(x, y))) \, dx \, dy}{\int_\Omega (1 - H(\phi(x, y))) \, dx \, dy}$$

Making it use Level Sets(Continued)

Because the Heaviside function is discontinuous, this could lead to problems. We will want to approximate (regularize) this in order to avoid these problems.

$$H_{2,\epsilon}(z) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{z}{\epsilon}\right)\right)$$

Active Contours without Edges Level Set Method

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

Active Contours without Edges Level Set Method (Discretization)

$$\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta t}=\delta_h(\phi_{i,j}^{n})\Bigg[$$

$$\frac{\mu}{\Delta x^2}(\phi_{i,j}^{n}-\phi_{i-1,j}^{n})\left(\frac{\phi_{i+1,j}^{n+1}-\phi_{i,j}^{n+1}}{\sqrt{\frac{(\phi_{i+1,j}^{n}-\phi_{i,j}^{n})^2}{\Delta x^2}+\frac{(\phi_{i,j+1}^{n}-\phi_{i,j-1}^{n})^2}{2\Delta x^2}}}\right)$$

$$+\frac{\mu}{\Delta x^2}(\phi_{i,j}^{n}-\phi_{i,j-1}^{n})\left(\frac{\phi_{i,j+1}^{n+1}-\phi_{i,j}^{n+1}}{\sqrt{\frac{(\phi_{i+1,j}^{n}-\phi_{i-1,j}^{n})^2}{2\Delta x^2}+\frac{(\phi_{i,j+1}^{n}-\phi_{i,j}^{n})^2}{\Delta x^2}}}\right)$$

$$-\nu-\lambda_1(u_{0,i,j}-c_1(\phi^n))^2+\lambda_2(u_{0,i,j}-c_2(\phi^n))^2\Bigg]$$

Active Contours without Edges Algorithm

- Initialize ϕ^0 by $\phi_{0,} n = 0.$
- Compute $c_1(\phi^n)$ and $c_1(\phi^n)$.
- Solve the PDE in ϕ to get ϕ^{n+1}
- Reinitialize ϕ. (optional)
- Check if the solution is stationary, else iterate.

Outline

Problem description

Active contours scalar case

Active contours vector-valued case

Vector-Valued Model

Defining some more things:

$u_{0,i}$ — *i*-th channel of an image u_0 on Ω with $i=1,...,N$ channels.

c_i^+ — Average intensity <u>inside</u> the curve *C* on channel *i*. (c_1 for scalar case)

$$= average(u_{0,i})\, on\, \phi \geq 0$$

c_i^- — Average intensity <u>outside</u> the curve *C* on channel *i*. (c_2 for scalar case)

$$= average(u_{0,i})\, on\, \phi < 0$$

Vector-Valued Model

$$F(\overline{c^+}, \overline{c^-}, \phi) = \mu \cdot Length(C)$$

Basically just averaging the squared difference of each pixel between all channels.

$$+ \int_{inside(C)} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^+ \left| u_{0,i}(x,y) - c_i^+ \right|^2 dx\,dy$$

$$+ \int_{outside(C)} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^- \left| u_{0,i}(x,y) - c_i^- \right|^2 dx\,dy$$

$\lambda_i^+ > 0$ and $\lambda_i^- > 0$ are parameters for each channel

Vector-Valued Model (Level Set Form)

$$F(\overline{c^+}, \overline{c^-}, \phi) =$$

$$\mu \int_\Omega \delta(\phi(x,y)) |\nabla \phi(x,y)| dx\, dy$$

$$+ \int_\Omega \frac{1}{N} \sum_{i=1}^{N} \lambda_i^+ \left| u_{0,i}(x,y) - c_i^+ \right|^2 H(\phi(x,y))\, dx\, dy$$

$$+ \int_\Omega \frac{1}{N} \sum_{i=1}^{N} \lambda_i^- \left| u_{0,i}(x,y) - c_i^- \right|^2 (1 - H(\phi(x,y)))\, dx\, dy$$

Vector-Valued Model (Euler-Lagrange)

$$\frac{\partial \phi}{\partial t} = \delta_{\epsilon}\left[\mu \cdot div\left(\frac{\nabla \phi}{|\nabla \phi|}\right) - \frac{1}{N}\sum_{i=1}^{N}\lambda_i^{+}(u_{0,i}-c_i^{+})^2 + \frac{1}{N}\sum_{i=1}^{N}\lambda_i^{-}(u_{0,i}-c_i^{-})^2\right]$$

Vector-Valued Model Discretization

Scalar Euler-Lagrange

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

Scalar Discretized Form

$$\frac{\phi_{i,j}^{n+1} - \phi_{i,j}^{n}}{\Delta t} = \delta_h(\phi_{i,j}^{n})\left[\frac{\mu}{\Delta x^2}(\phi_{i,j}^{n} - \phi_{i-1,j}^{n})\left(\frac{\phi_{i+1,j}^{n+1} - \phi_{i,j}^{n+1}}{\sqrt{\frac{(\phi_{i+1,j}^{n} - \phi_{i,j}^{n})^2}{\Delta x^2} + \frac{(\phi_{i,j+1}^{n} - \phi_{i,j-1}^{n})^2}{2\Delta x^2}}}\right)\right.$$

$$\left. + \frac{\mu}{\Delta x^2}(\phi_{i,j}^{n} - \phi_{i,j-1}^{n})\left(\frac{\phi_{i,j+1}^{n+1} - \phi_{i,j}^{n+1}}{\sqrt{\frac{(\phi_{i+1,j}^{n} - \phi_{i-1,j}^{n})^2}{2\Delta x^2} + \frac{(\phi_{i,j+1}^{n} - \phi_{i,j}^{n})^2}{\Delta x^2}}}\right) - \nu - \lambda_1 (u_{0,i,j} - c_1(\phi^n))^2 + \lambda_2 (u_{0,i,j} - c_2(\phi^n))^2\right]$$

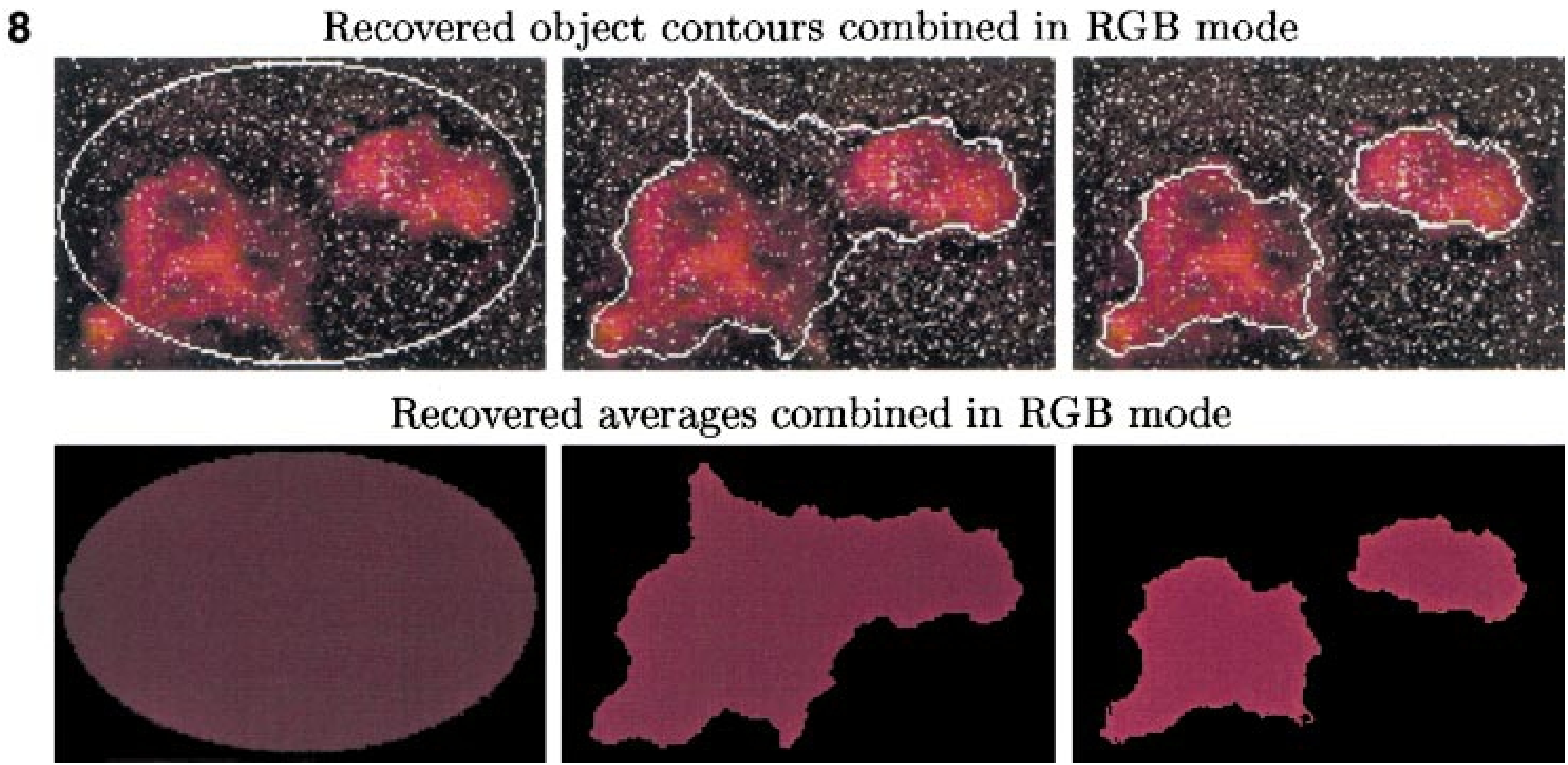
Vector-Valued Euler-Lagrange

$$\frac{\partial \phi}{\partial t} = \delta_{\epsilon}\left[\mu \cdot div\left(\frac{\nabla \phi}{|\nabla \phi|}\right) - \frac{1}{N}\sum_{i=1}^{N} \lambda_i^{+} (u_{0,i} - c_i^{+})^2 + \frac{1}{N}\sum_{i=1}^{N} \lambda_i^{-} (u_{0,i} - c_i^{-})^2\right]$$

Vector-Valued Discretized Form

Something more complex, no doubt.

Some Results



Questions?