Image Segmentation and Tracking using a Difference of Convex Regularized Mumford-Shah Functional

Fred Park
Whittier College

UC Irvine AI/ML Seminar
November 5th, 2018

Collaborators

- ❖ Jack Xin (UCI Math)
- ❖ Yifei Lou (UT Dallas Math)

Work supported by:

- ONR (grants N00014-11-1-0602)
- NSF (grants DMS-1222507, DMS-1522383, and DMS-1522786)

Outline

① Image Segmentation and Tracking
- Motivation for Processing
- Image Denoising: TV Model
- Image Segmentation: Chan Vese Model

② Shape Modeling

③ Shape Prior Segmentation

④ Point Cloud Surface Reconstruction

⑤ Ongoing and Future Work

Outline

① Intro to Image Processing
  - Image Representation
  - Motivation for Processing
  - Image Denoising: TV Model
  - Image Segmentation: Chan Vese Model

② Shape Modeling

③ Shape Prior Segmentation

④ Point Cloud Surface Reconstruction

⑤ Ongoing and Future Work

Image Segmentation

- Problem: Detect Salient Regions & their Boundaries
- Often: P.W. Smooth (or cartoonish) regions & sharp boundaries
- Useful for object location & recognition, medical imaging (tumor detection etc.), face recognition, tracking, … and more

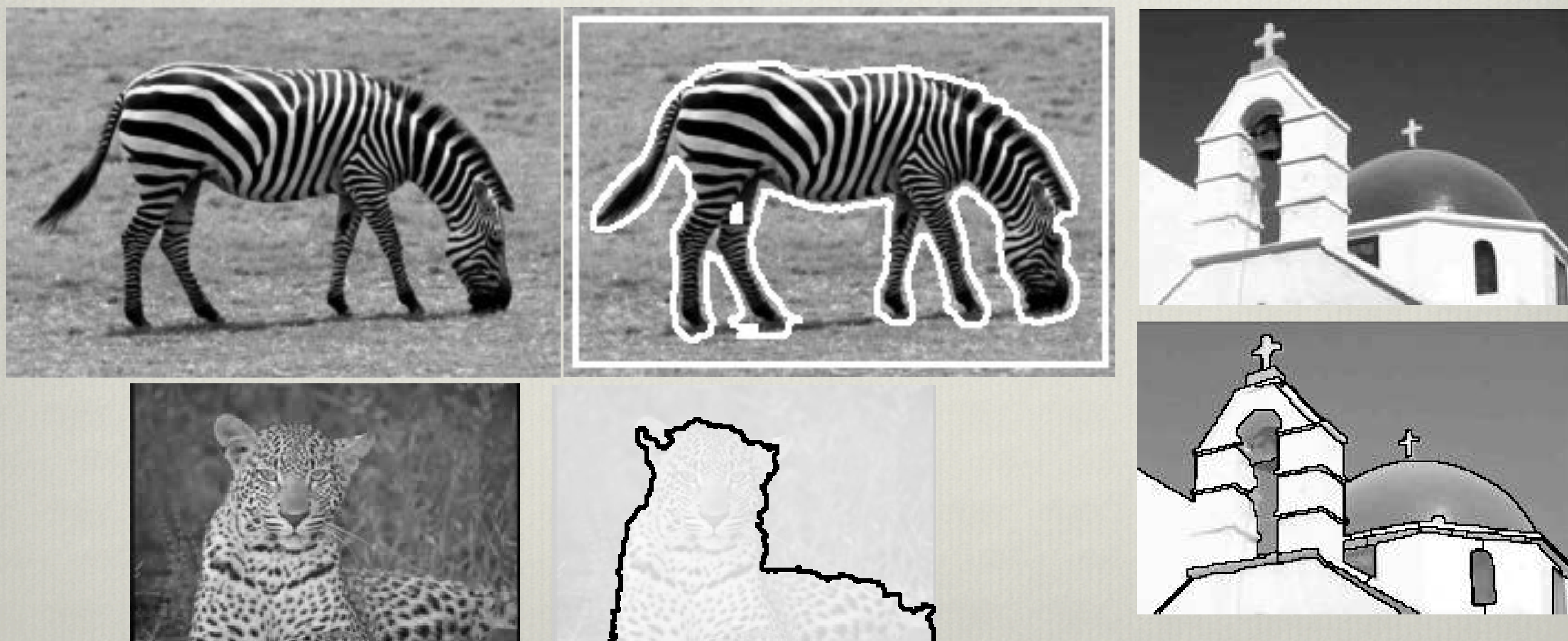


Image Segmentation: Motivation

Recall the Los Angeles Riots of 1992

Image Segmentation: Some Motivation

High Point of Riots: Reginald Denny Beaten Mercilessly on Nat'l. TV

Public Outrage!

Perpetrators at large!

Calculus Based Image Processing Used to Enhance Footage

Cognitech and UCLA Image Processing Group Help LAPD

THE WALL STREET JOURNAL.

California Company Uses Calculus to Pin The Crime on the Criminal Who Did It

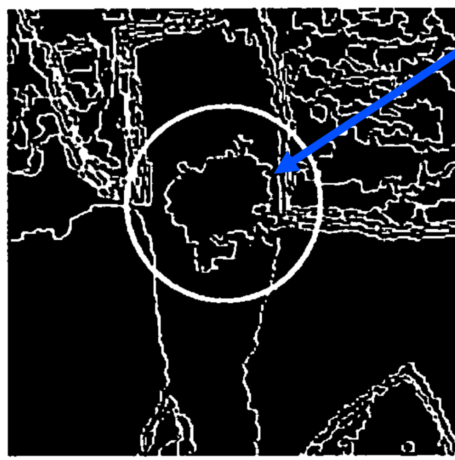
By Robert Langreth
Staff Reporter of The Wall Street Journal

The two men were on trial for murder. Convictions might have been easy: A gas station security camera had filmed the whole tussle, culminating in fatal gunshots. But the videotape was so blurry that no one could really tell who attacked whom. The two argued self-defense, and the Los Angeles County jury hung.

So local detectives turned to Cognitech Inc., a tiny company armed with a powerful new technique for enhancing fuzzy images. Cognitech's improved video clearly showed the suspects pinning the victim face down against the ground and firing into his skull. Both defendants eventually pleaded guilty.

In the past two years, analyzing crime and accident videotapes has blossomed into a full-time business for Cognitech, based in Santa Monica, Calif. It is among a handful of companies applying sophisticated mathematics to clearing up crime and accident videotapes.

Before these companies existed, police trying to enhance poor videos had to buy commercial "Photoshop" software, which generally processes one frame at a time and is limited to simple operations such as improving contrast. Or they could send their



This computer-generated image is the first step in a process that allowed Cognitech to identify a tattoo (circled) on the arm of Reginald Denny's attacker

frame and the nature of distortions caused by poor focus, atmospheric refraction, electronic noise, and

work is done on computer workstations at employees' desktops.

On a typical day, Mr. Rudin prowls the office looking over employees' shoulders as they analyze videotapes, asking questions and making suggestions. For particularly stubborn videotapes, the indefatigable Mr. Rudin often stays late into the night adapting computer programs to do that type of image better.

Cognitech isn't alone in the expanding video-enhancement field. Another small company, Trec Inc. in Huntsville, Ala. sells software for enhancing videotapes to the FBI and other law enforcement agencies. And Aerospace Corp., a nonprofit military-research agency, recently started a small unit to analyze crime videotapes. It now handles a couple of dozen cases per year.

Neither the FBI nor Trec will comment on their enhancement techniques. Aerospace, for its part, says a variety of standard mathematical methods for improving images serve it just fine. "Standard image enhancement is a whopping field," agrees Massachusetts Institute of Technology electrical engineer Alan Willsky. He adds that "the jury's still out on the overall impact" of Cognitech's method.

In any case, all sides expect their caseloads to increase as more businesses buy sophisticated video-

Los Angeles Times

The Region : SOUTHERN CALIFORNIA ENTERPRISE : Cognitech Thinks It's Got a Better Forensic Tool : The firm uses complex math in video image-enhancing technology that helps in finding suspects.

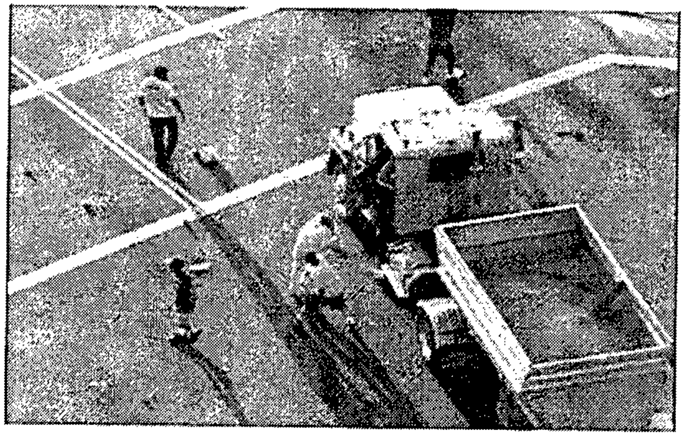
September 05, 1994 | KAREN KAPLAN | TIMES STAFF WRITER

It started out as just a speck on a photograph of a man who threw a brick at truck driver Reginald Denny at Florence and Normandie avenues in the opening hours of the 1992 Los Angeles riots.

But when Leonid Rudin subjected it to a complicated computer algorithm and a slew of complex mathematical equations, that speck--originally less than 1/6,000th the size of the total photograph--was revealed to be a rose-shaped tattoo on the arm of the man, later identified in court as Damian Monroe Williams.

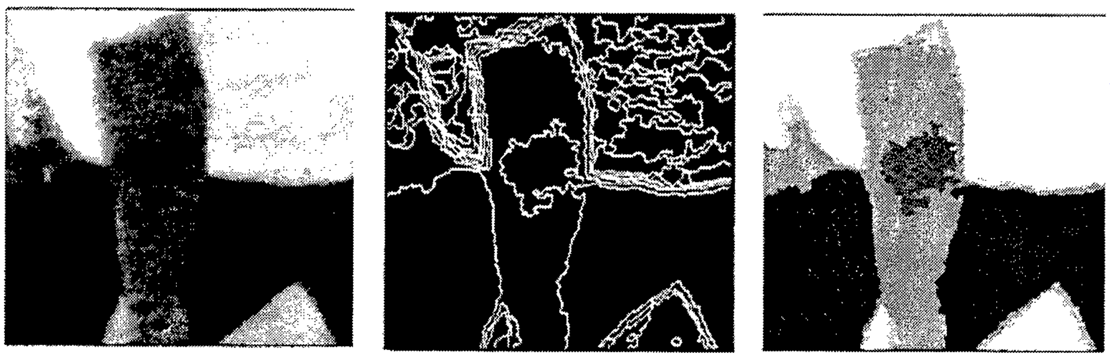
Image Processing Crime Footage

Reginald Denny Beating Investigation



Original Image

← Original Crime Scene Video



Perpetrator's tatoo superresolution

← Tattoo Superresolution

Comparison with the suspect's real tatoo

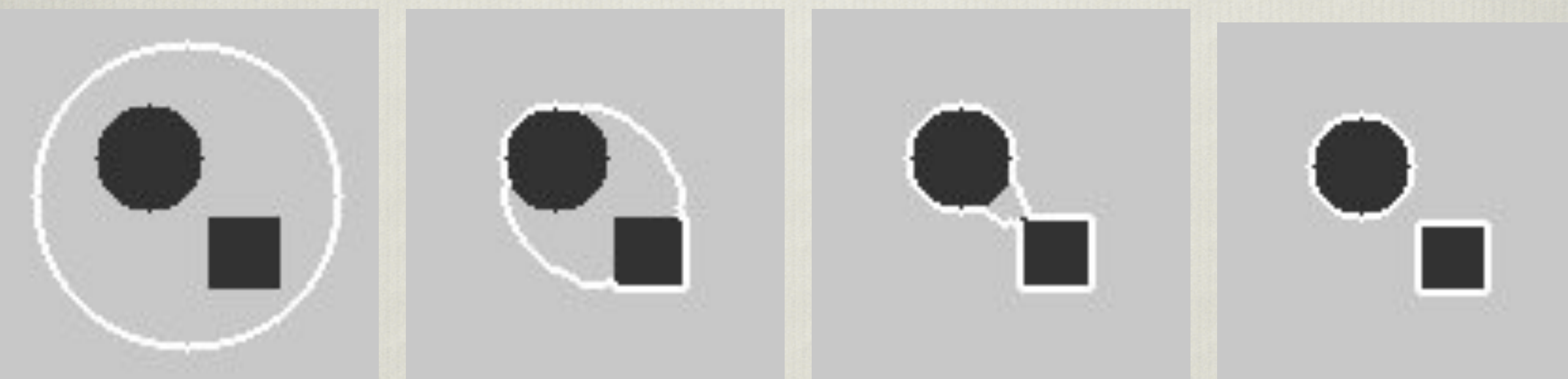
← Comparison w/ Real Tatoo

Outcome:
All 3 Criminals Convicted!

Application: "active contour"

- giving an image $f:\Omega \to \Re$
- evolve a curve C to detect objects in f
- the curve has to stop on the boundaries of the objects

Initial Curve ⟶ Evolutions ⟶ Detected



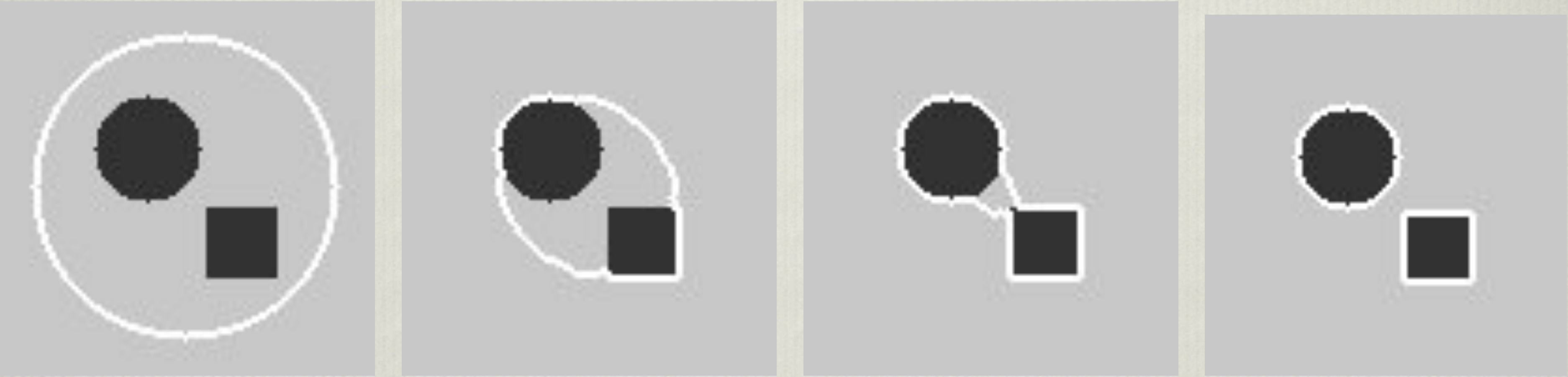
Basic idea in classical active contours

Curve evolution and deformation (internal forces):

$$\text{Min} \;\; Length(C) + Area(inside(C))$$

Boundary detection: what is it? What is stopping criteria for curve?

Initial Curve ⟶ Evolutions ⟶ Detected



Data Fidelity Term

$$\int_{inside(C)} |f - c_1|^2 dxdy + \int_{outside(C)} |f - c_2|^2 \, dx\, dy$$

where

$$c_1 = average(f) \text{ inside } C$$

$$c_2 = average(f) \text{ outside } C$$

Fit > 0 Fit > 0 Fit > 0 Fit ~ 0

Minimize: (Fitting +Regularization)

Fitting not depending on gradient detects → "contours without gradient"

Chan-Vese (CV) Model

Fitting + Regularization terms (length, area)

$$\inf_{c_1, c_2, C} F(c_1, c_2, C) = \mu \cdot |C| + \nu \cdot Area(inside(C))$$

$$+ \lambda \int_{inside(C)} |u_0 - c_1|^2 \, dxdy + \lambda \int_{outside(C)} |u_0 - c_2|^2 \, dxdy$$

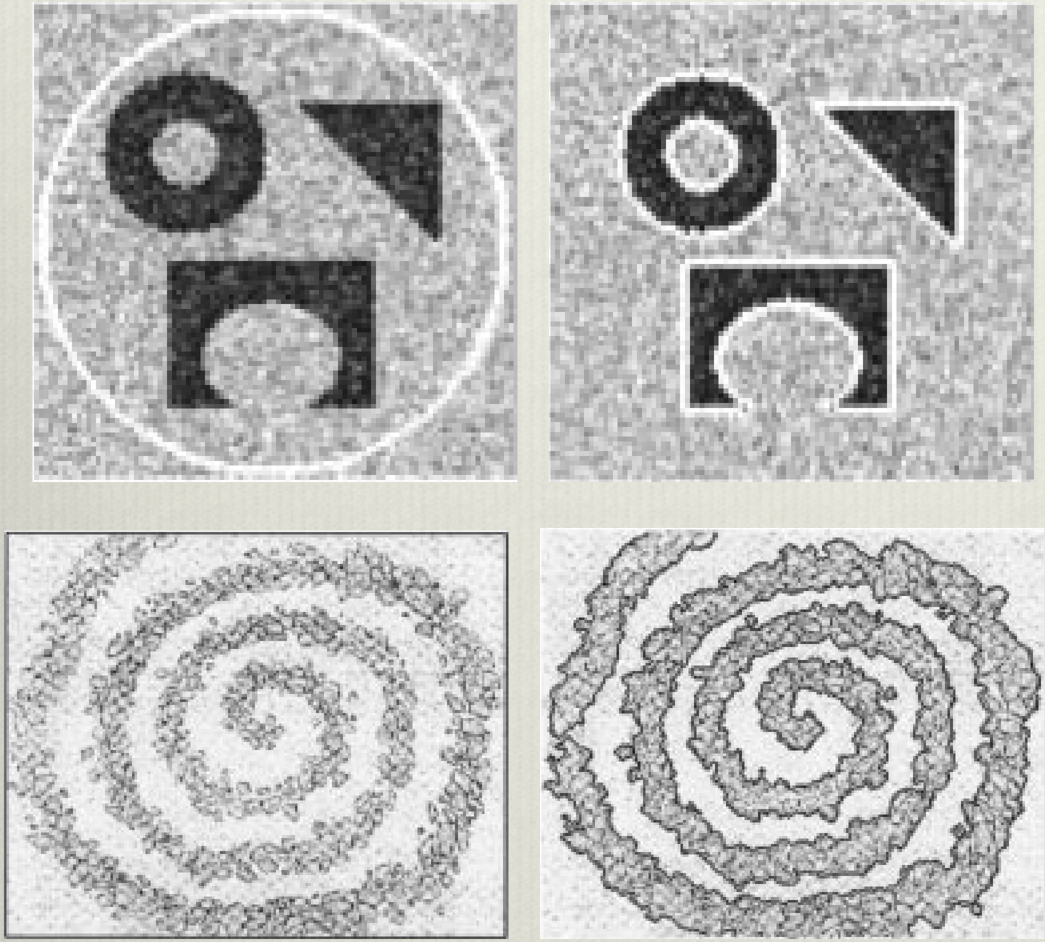
C = boundary of an open and bounded domain

$|C|$ = the length of the boundary-curve C

- P.W. Constant Version of Mumford Shah Model
- Fit constant homogeneous regions while enforcing regularity on boundary of C
- Active Contours without Edges

Experimental Results

Evolution of C



Advantages

Automatically detects interior contours!

Works very well for concave objects

Robust w.r.t. noise

Detects blurred contours

The initial curve can be placed anywhere!

Allows for automatic change of topologies

Outline

① Intro to Image Processing
- Image Representation
- Motivation for Processing
- Image Denoising: TV Model
- Image Segmentation: Chan Vese Model

② Shape Modeling

③ Shape Prior Segmentation

④ Point Cloud Surface Reconstruction

⑤ Ongoing and Future Work

Convex Relaxed CV Model

Prev. Implement's. of CV model use level set methods

$$CV(\phi, c_1, c_2) = \int_D |\nabla H(\phi)|$$
$$\lambda \int_D H(\phi)(c_1 - f(x))^2 + (1 - H(\phi))(c_2 - f(x))^2 dx$$

ϕ: level set function $\quad$ $C = \{x : \phi = 0\}$: object boundaries

$H(\phi)$ is Heaviside function

$$H(x) = \begin{cases} 0 & \text{if } x < 0; \\ 1 & \text{if } x \geq 0. \end{cases}$$

Convex Relaxed CV Model

$$CV(\phi, c_1, c_2) = \int_D |\nabla H(\phi)|$$
$$\lambda \int_D H(\phi)(c_1 - f(x))^2 + (1 - H(\phi))(c_2 - f(x))^2 dx$$

Level Set Methods:

- ❖ Depend on initialization → yields non-unique segmentations
- ❖ Needs re-distancing → slower convergence & computationally intensive
- ❖ Level set formulation is non-convex (Heaviside function → binary)
- ❖ New implement's of MS/CV models use convex relaxation techniques

Convex Relaxed CV Model

Relaxed 2 phase Mumford-Shah (Chan-Vese) model:

$$\min_{0\leq u\leq 1} \int_\Omega |\nabla u| + \lambda \int_\Omega \left[(c_1 - f)^2 - (c_2 - f)^2\right] u \, dx$$

Segmented region Σ via thresholding: $\Sigma = \{x \,|\, u(x) \geq 1/2\}$

- For fixed c_1 and c_2 model is convex
- Robust to initialization
- Fast minimization techniques

Convex Relaxed CV Model

$$\min_{0 \le u \le 1} \int_\Omega |\nabla u| + \lambda \int_\Omega \left[(c_1 - f)^2 - (c_2 - f)^2\right] u \, dx$$

$$L_2 \text{ TV of } u = \|\sqrt{|D_x u|^2 + |D_y u|^2}\|_1$$

$$L_1 \text{ TV of } u = \|D_x u\|_1 + \|D_y u\|_1$$

Caveats of CV model (L_2 TV norm):

- Isotropic TV norm (L_2 TV) regularizes in all gradient directions
- Gradient distributions not equal in all directions
- May not capture appropriate boundaries of certain objects
- Anisotropic TV norm (L_1 TV) is more directional
- Boils down to what norm has most sparsity on the gradient of u

Convex Relaxed CV Model

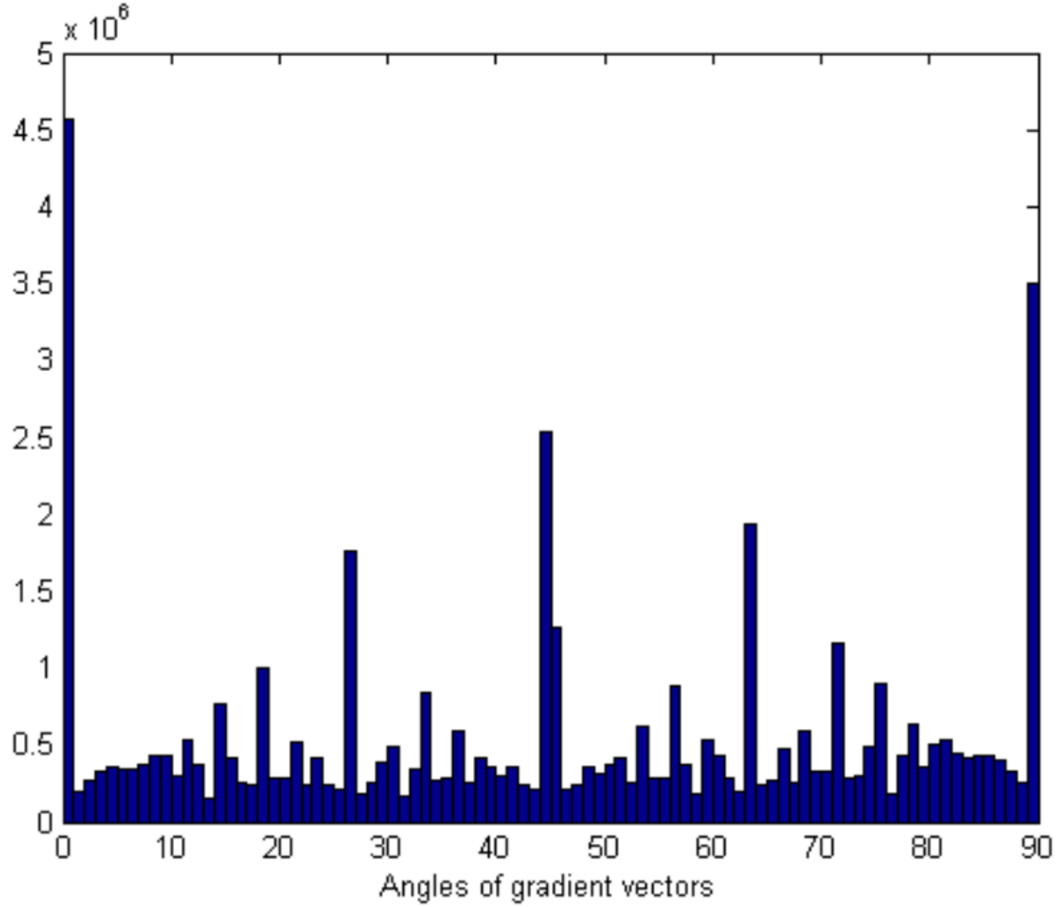
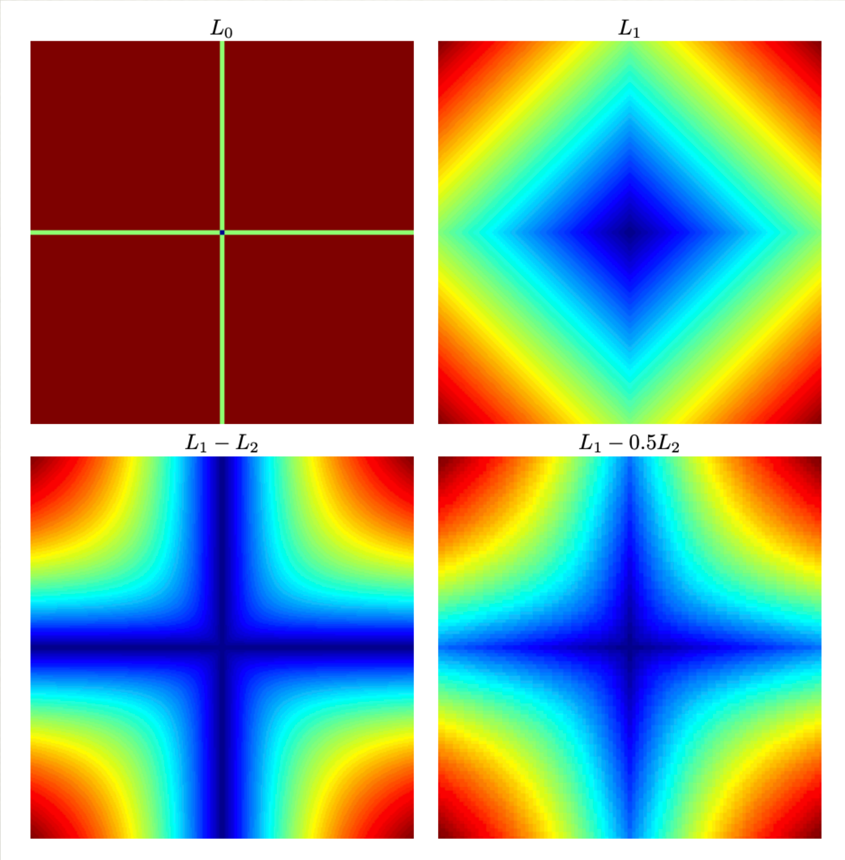


FIG. 2.1. *The histogram of gradient angles over 300 images from Berkeley segmentation dataset [27]. Two largest peaks are at 0 and 90 degrees, indicating that gradient vectors are mostly 1-sparse.*

Most gradient angles are at 0 and 90 degrees!

Convex Relaxed CV Model

- Partition boundaries in MS & Potts model rep'd by L_0 Norm: $J(u) = \|\nabla u\|_0$
- Gradient distribn's mostly vertical and horizontal in natural images.
- L_1-αL_2 TV norm is better approx'n to L_0 via level lines than L_2 TV.
- α chosen based on gradient distrib's



L_1 - $0.5L_2$ closer to L_0 than: L_1, L_2, and L_1 - L_2

Level lines plot

Proposed Model

Difference of Anisotropic and Isotropic TV CV model

$$\min_{0\le u\le 1} \int_\Omega |u_x| + |u_y| - \alpha\,|\nabla u| + \lambda \int_\Omega \underbrace{\left[(c_1 - f)^2 - (c_2 - f)^2\right]}_{r_1(c_1,c_2)} u\,dx$$

Convex Relaxed CV model is now Non-Convex!

How to minimize it now?

Proposed Model

Difference of Anisotropic and Isotropic TV CV model

$$\min_{0\leq u\leq 1} \int_\Omega |u_x| + |u_y| - \alpha\,|\nabla u| + \lambda \int_\Omega \underbrace{\left[(c_1 - f)^2 - (c_2 - f)^2\right]}_{r_1(c_1,c_2)} u\,dx$$

Can be written as a difference of convex! $F(u) = G(u) - H(u)$

$$\begin{cases} G(u) = \|u_x\|_1 + \|u_y\|_1 + c\|u\|_2^2 + \lambda\,\langle r_1,\, u\rangle \\ H(u) = \alpha\|\nabla u\|_{2,1} + c\|u\|_2^2 \end{cases}$$

(c is parameter for strong convexity)

DCA Algorithm

DCA algorithm → linearization of the non-convex terms and then iterating

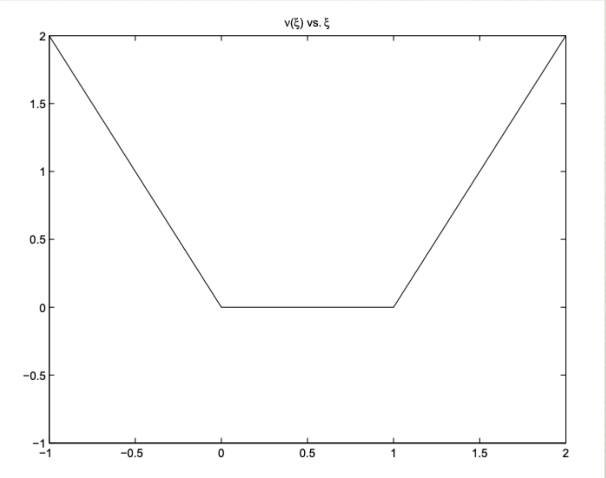
$$u^{n+1} = \arg\min_u \|u_x\|_1 + \|u_y\|_1 + c\|u\|_2^2 + \lambda\langle r_1, u\rangle - \alpha\langle q^n,\ \nabla u\rangle - 2c\langle u,\ u^n\rangle + \langle \nu(u),\ 1\rangle$$

linearized

$\nu(\xi) = \max\{0, 2|\xi - \frac{1}{2}| - 1\}$ is an exact penalty

$$q^n = \frac{\nabla u^n}{|\nabla u^n|}$$

here $q^n = 0$ if $|\nabla u^n| = 0$



Splitting

splitting

$$u^{n+1} = \arg\min_{u,v} \|u_x\|_1 + \|u_y\|_1 + c\|u\|_2^2 + \lambda\langle r_1, v\rangle$$
$$- \alpha\langle q^n, \nabla u\rangle - 2c\langle u, u^n\rangle + \beta\langle \nu(v), 1\rangle + \tfrac{1}{2\theta}\|u - v\|_2^2$$

Split data fidelity and exact penalty terms

Equivalent to minimizing sub-problems:

1. v fixed, search for u as solution of:

$$\min_u \|u_x\|_1 + \|u_y\|_1 + c\|u\|_2^2 - \alpha\langle q^n, \nabla u\rangle - 2c\langle u, u^n\rangle + \frac{1}{2\theta}\|u - v\|_2^2$$

2. u fixed, search for v a solution of:

$$\min_v \lambda\langle r_1, v\rangle + \langle \nu(v), 1\rangle + \frac{1}{2\theta}\|u - v\|_2^2$$

Solving Sub-Problems

1. v fixed, search for u as solution of:

$$\min_u \|u_x\|_1 + \|u_y\|_1 + c\|u\|_2^2 - \alpha\langle q^n, \nabla u\rangle - 2c\langle u, u^n\rangle + \frac{1}{2\theta}\|u - v\|_2^2$$

2. u fixed, search for v a solution of:

$$\min_v \lambda\langle r_1, v\rangle + \langle \nu(v), 1\rangle + \frac{1}{2\theta}\|u - v\|_2^2$$

v found by shrinkage: $v = \min\left\{ \max\left\{ u(x) - \theta\lambda r_1(x, c_1, c_2), 0 \right\}, 1 \right\}$

u found via Primal Dual Hybrid Gradient Method:

Primal Dual Hybrid Gradient

Primal problem:

$$\min_u \|u_x\|_1 + \|u_y\|_1 + c\|u\|_2^2 - \alpha\langle q^n, \nabla u\rangle - 2c\langle u, u^n\rangle + \frac{1}{2\theta}\|u-v\|_2^2$$

Becomes the primal-dual problem:

$$\min_u \max_{|p_1|\leq 1, |p_2|\leq 1} -\langle u, {p_1}_x\rangle - \langle u, {p_2}_y\rangle + c\|u\|_2^2 + \alpha\langle \operatorname{div} q^n, u\rangle$$
$$- 2c\langle u, u^n\rangle + \frac{1}{2\theta}\|u-v\|_2^2$$

How to solve the primal-dual problem?

Fast method via the Primal Dual Hybrid Gradient Method (PDHG)
Zhu and Chan '08, Chambolle and Pock '10, Chambolle et al. '18

Primal Dual Hybrid Gradient

$$\min_{u} \max_{|p_1|\leq 1,\, |p_2|\leq 1} -\langle u,\, p_{1_x}\rangle - \langle u,\, p_{2_y}\rangle + c\|u\|_2^2 + \alpha\langle \operatorname{div} q^n,\, u\rangle - 2c\langle u,\, u^n\rangle + \frac{1}{2\theta}\|u - v\|_2^2$$

Key idea:

1. Take 1 step of gradient descent in the primal variable u
2. Take 1 step of projected gradient ascent in the dual variable $\vec{p} = \langle p_1,\, p_2\rangle$

Primal Dual Hybrid Gradient

$$\min_{u} \max_{|p_1|\leq 1,\, |p_2|\leq 1} -\langle u,\, {p_1}_x\rangle - \langle u,\, {p_2}_y\rangle + c\|u\|_2^2 + \alpha\langle \operatorname{div} q^n,\, u\rangle$$
$$- 2c\langle u,\, u^n\rangle + \frac{1}{2\theta}\|u - v\|_2^2$$

Key idea:

1. Take 1 step of gradient descent in the primal variable u
2. Take 1 step of projected gradient ascent in the dual variable $\vec{p} = \langle p_1,\, p_2\rangle$

$$u^{k+1} \;=\; (1 - \theta_k - 2c\,\theta_k\theta)\,u^k + \theta_k\left(v - \theta(-\operatorname{div}\vec{p}^{\,k+1} + \alpha\operatorname{div}\vec{q}^{\,n} - 2cu^n)\right)$$

Primal Dual Hybrid Gradient

$$\min_{u} \max_{|p_1|\le 1,\, |p_2|\le 1} -\langle u,\, p_{1_x}\rangle - \langle u,\, p_{2_y}\rangle + c\|u\|_2^2 + \alpha\langle \operatorname{div} q^n,\, u\rangle$$

$$- 2c\langle u,\, u^n\rangle + \frac{1}{2\theta}\|u - v\|_2^2$$

Key idea:

1. Take 1 step of gradient descent in the primal variable u
2. Take 1 step of projected gradient ascent in the dual variable $\vec{p} = \langle p_1,\, p_2\rangle$

$$u^{k+1} \;=\; (1 - \theta_k - 2c\,\theta_k\theta)\, u^k + \theta_k \left(v - \theta(-\operatorname{div}\vec{p}^{\,k+1} + \alpha \operatorname{div}\vec{q}^{\,n} - 2cu^n) \right)$$

$$\vec{p}^{\,k+1} = P_{\tilde{\mathbf{X}}}(\vec{p}^{\,k} + \tau_k \nabla u^k)$$

$$P_{\tilde{\mathbf{X}}}(\vec{p}) = \left\langle \frac{p_1}{\max\{|p_1|, 1\}}, \frac{p_2}{\max\{|p_2|, 1\}} \right\rangle.$$

Primal Dual Hybrid Gradient

Final Algorithm

Algorithm 1 DCA to minimize unconstrained AICV

Initialization: Pick u^0 MaxDCA, MaxPDHG, step size τ_k and θ_k, and set $u = q = 0$.

for 1 **to** MaxDCA **do**

Set $v^0 = \vec{p}^{\,0} = 0$ and $k \leftarrow 0$

for 1 **to** MaxPDHG **do**

$$v^{k+1} = \min\left\{ \max\left\{u^k - \theta\lambda r_1(x, c_1, c_2),\, 0\right\},\, 1\right\}$$

$$\vec{p}^{\,k+1} = P_{\tilde{\mathbf{X}}}(\vec{p}^{\,k} + \tau_k \nabla u^k)$$

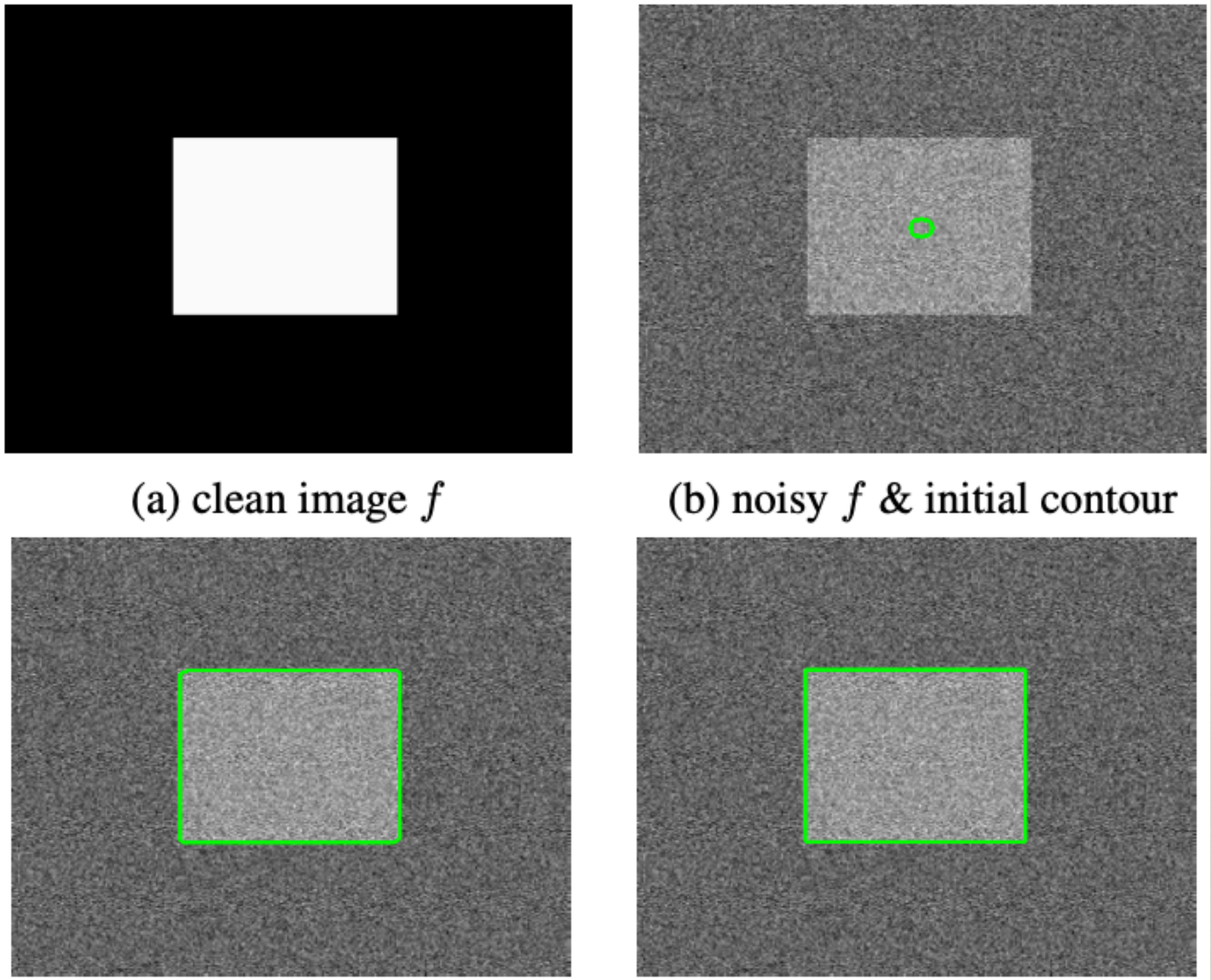
$$u^{k+1} = (1 - \theta_k - 2c\,\theta_k\theta)u^k + \theta_k(v^{k+1} - \theta(-\operatorname{div}\vec{p}^{\,k+1} + \alpha \operatorname{div}\vec{q} - 2cu))$$

end for

$u = u^{k+1}$, $\vec{q} = \nabla u^{k+1} / |\nabla u^{k+1}|$

end for

Numerical Results: Proposed

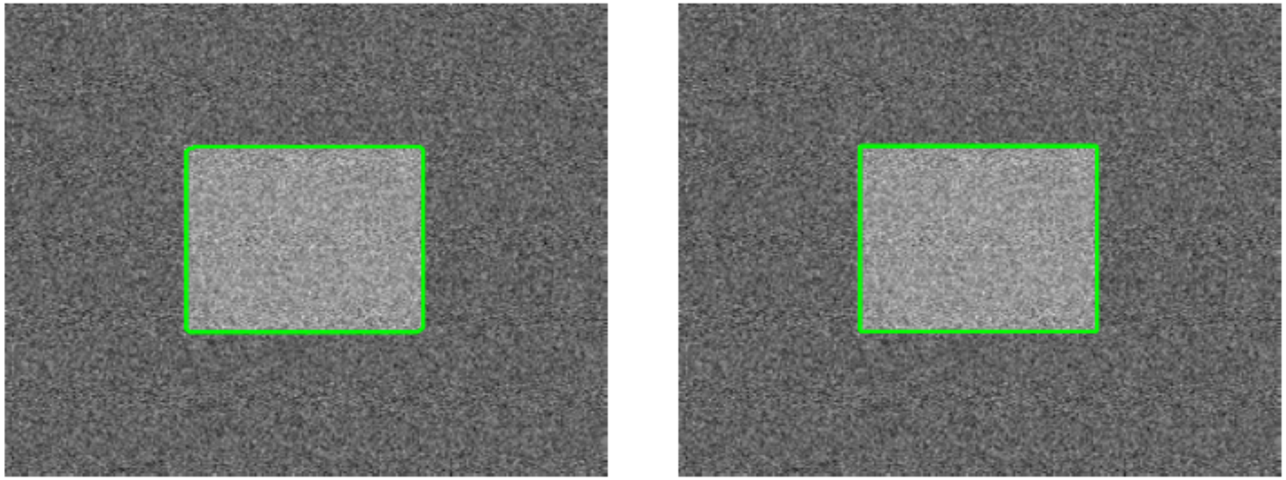


(a) clean image f (b) noisy f & initial contour

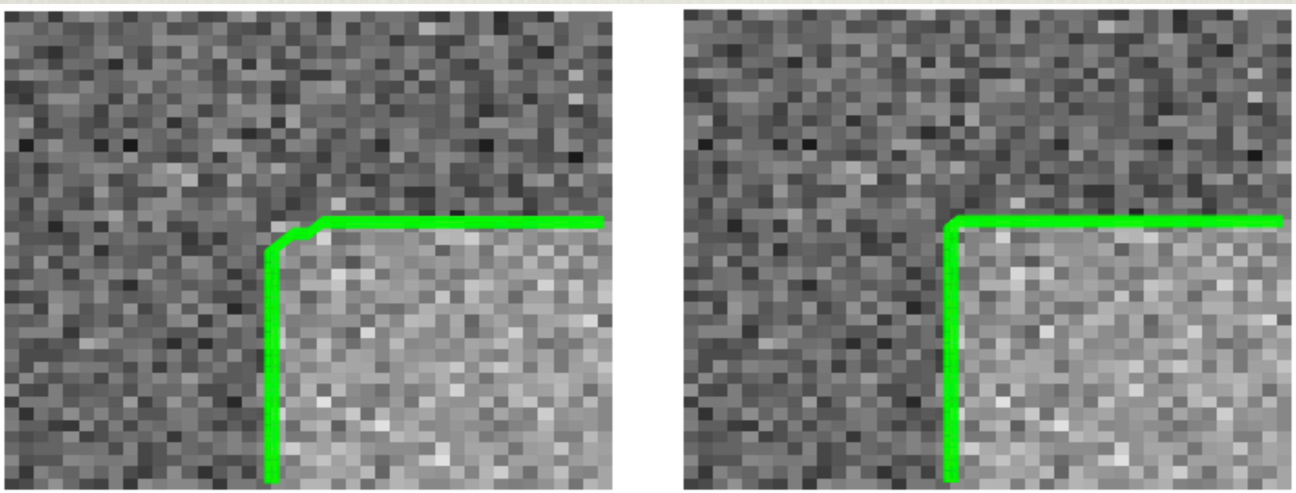
(c) TV ($\sigma = 0.5$) (d) $L_1, L_1 - \alpha L_2$ ($\sigma = 0.5$)

Proposed more accurately captures the boundary of the square

Numerical Results: Zoomed



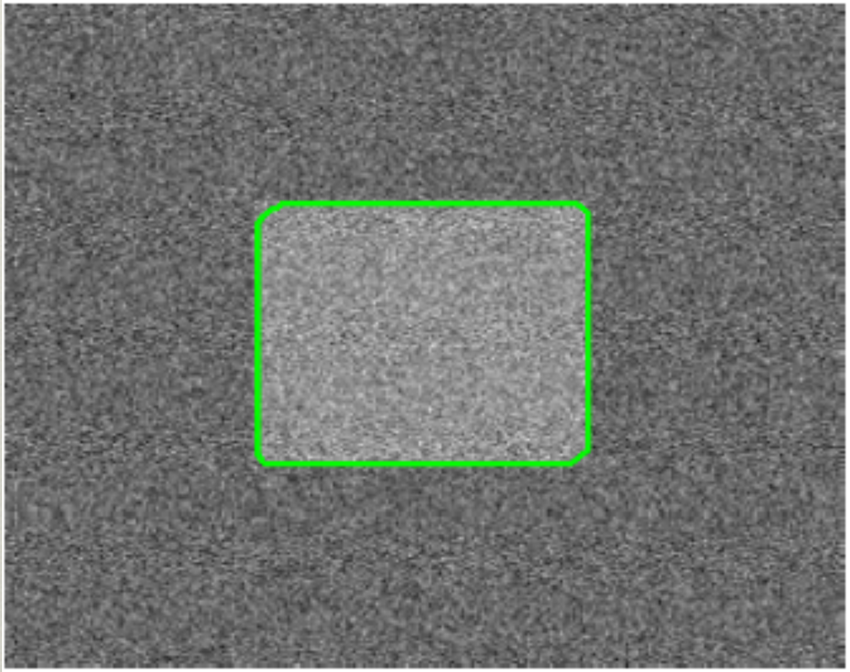
(c) TV ($\sigma = 0.5$) (d) $L_1, L_1 - \alpha L_2$ ($\sigma = 0.5$)



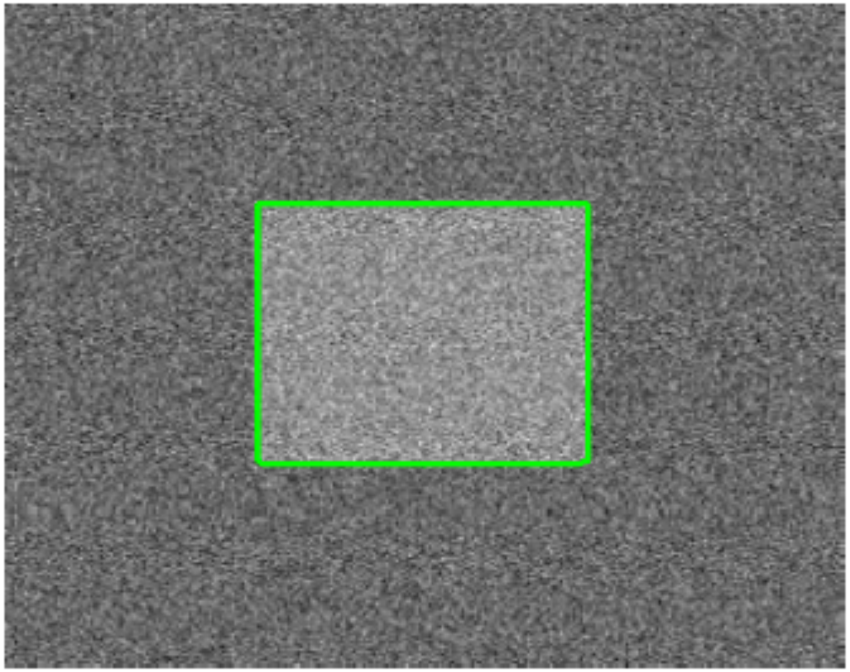
(e) TV ($\sigma = 0.5$) (f) $L_1, L_1 - \alpha L_2$ ($\sigma = 0.5$)

Proposed more accurately captures the boundary of the square

Numerical Results: More Noise



(g) TV ($\sigma = 0.65$)

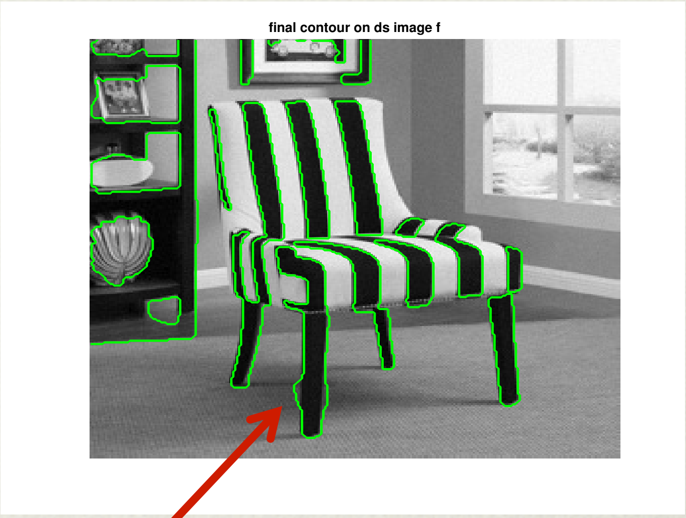


(h) $L_1 - 0.5L_2$ ($\sigma = 0.65$)

Proposed more accurately captures the boundary of the square

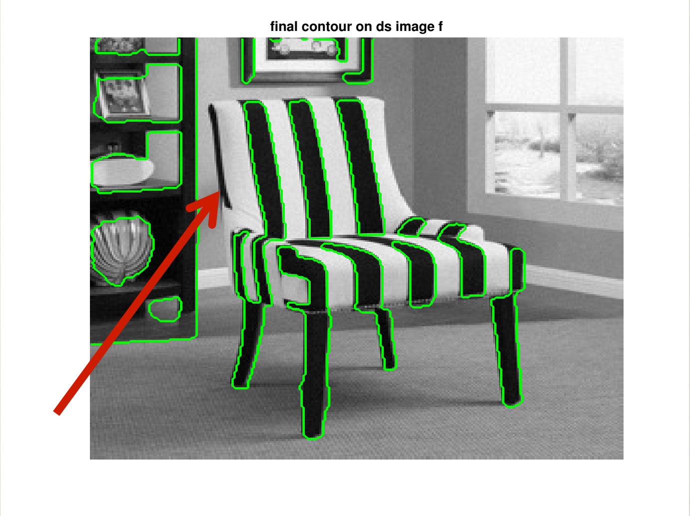
Numerical Results L_2 TV

Chair Image



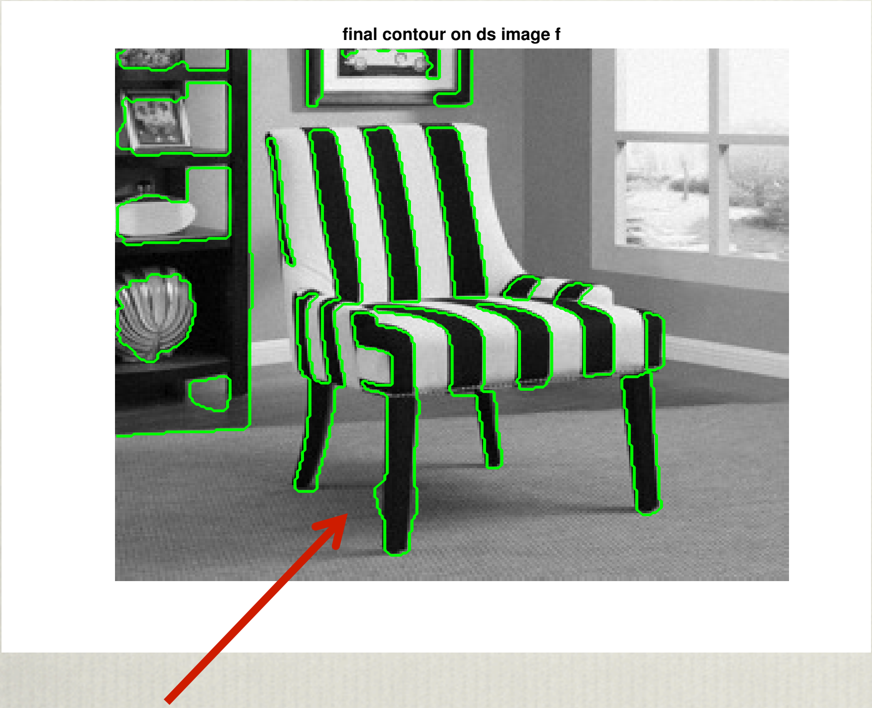
Regularize gradients in all directions

- Chair leg captured incorrectly.
- Increasing regularization loses features.



Increasing Regularization Results in Loss of Features

Numerical Results: Proposed

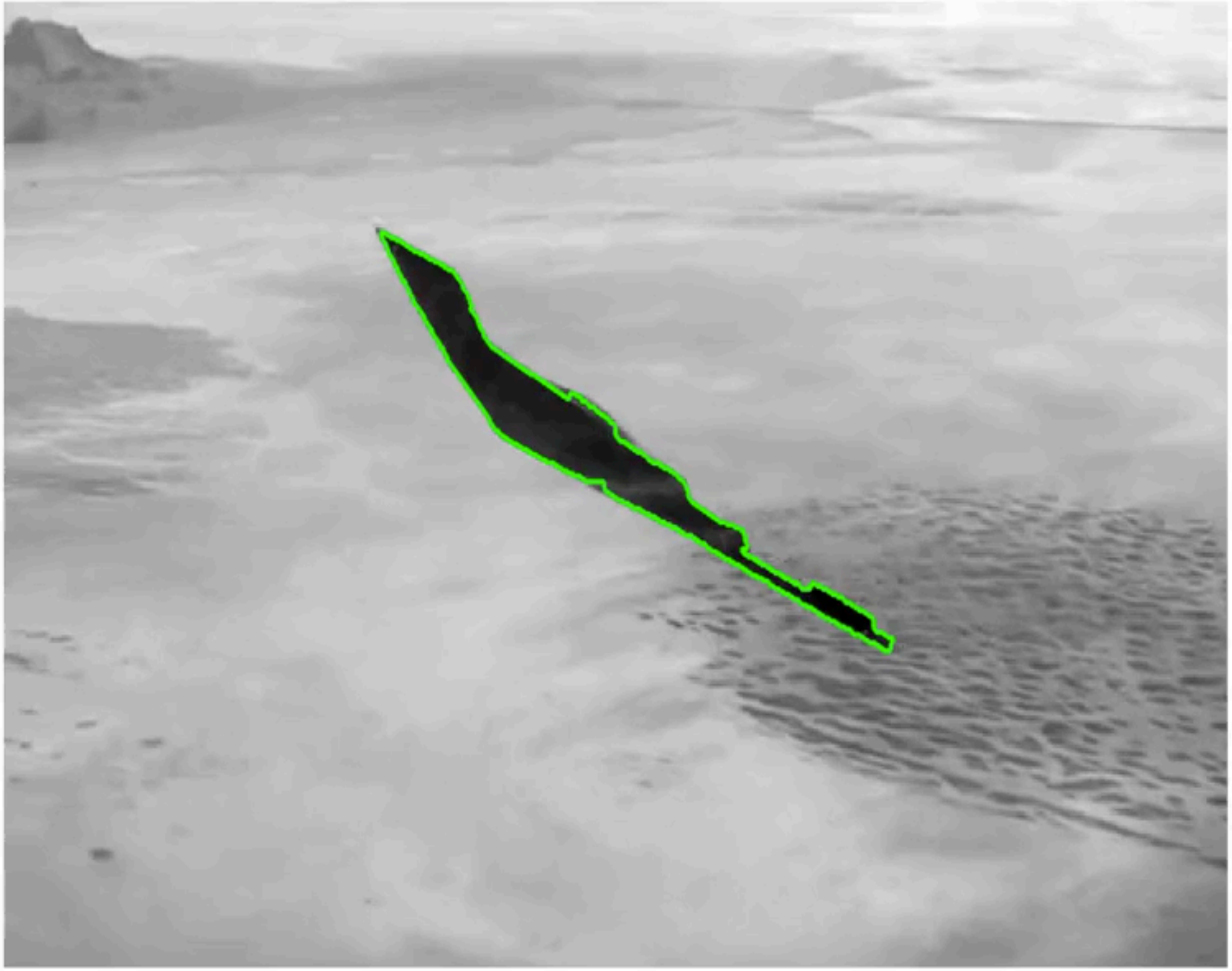


L_2 TV

L_1-αL_2 TV

Proposed accurately captures chair leg without loss of features

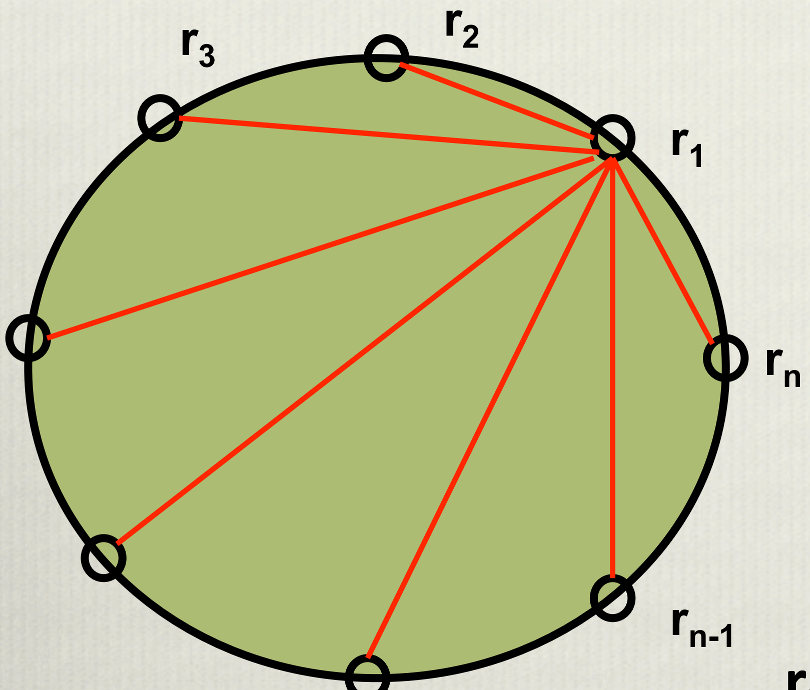
Proposed: Tracking Example



Ongoing and Future Work

- Convergence proof of the DCA algorithm for proposed model
- Other ways of achieving/exploiting directional sparsity
- Shape Prior Segmentation: Modeling both occluders and shapes
- Neural Network → semantic segmentation. Interplay between a trained model and a mathematical one. Not a 2 step approach but a synergistic one.
- Using CNN's for applications to spatially varying blind deconvolution.
- Primal Dual methods for Neural Network Optimization
- Convex relaxation techniques for NN's. Some work done on quantizing weights. Xi et al. '18

Cliques Invariant Signature



Motivatation:
Bending Invariant Signatures
Elad and Kimmel 03'

Intervertex Distances:

$$\sum_{i,j} (\| p_i - p_j \|^2 - \| r_i - r_j \|)^2$$

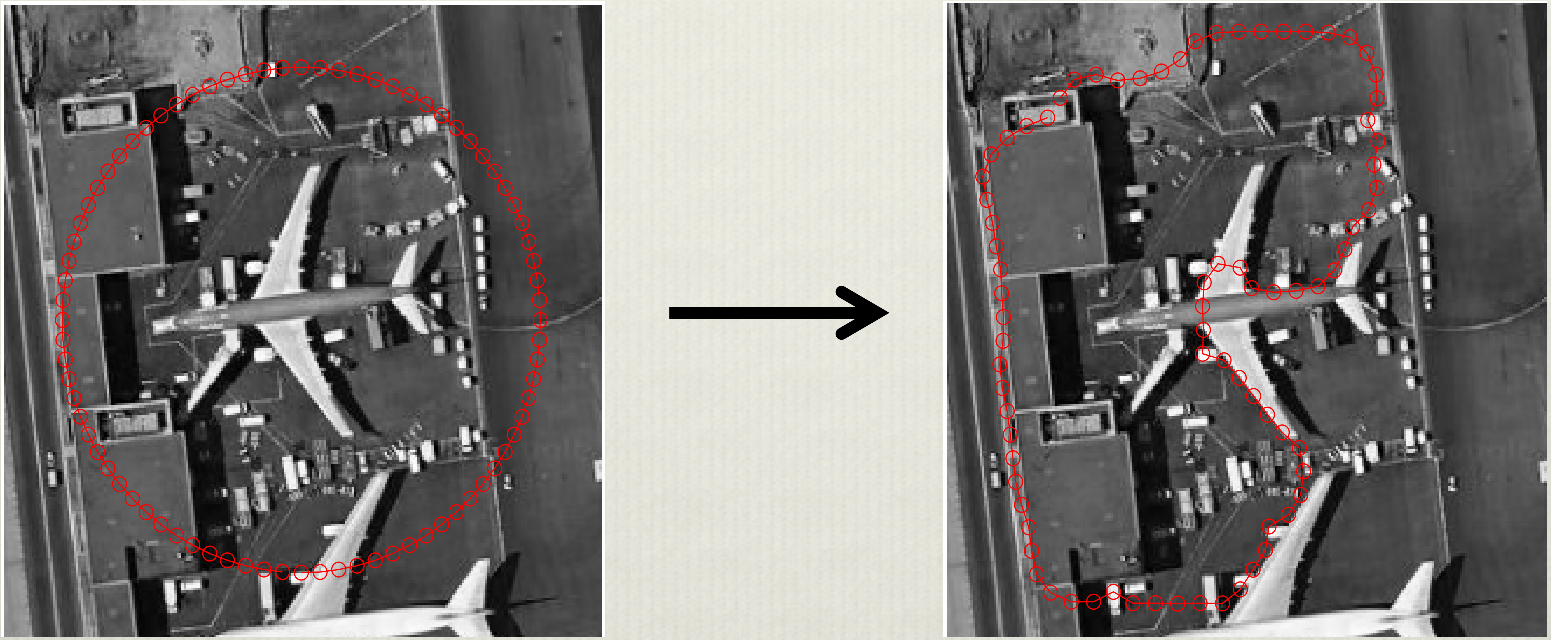
r_i: pts. lying on reference shape

p_i: pts. lying on some evolving contour

Incorporation into:
- Geodesic Active Contours (Snakes)
- Polygonal Implementation of the P.W. Constant MS Model

Shape Prior Segmentation

Why are Shape Priors Needed?

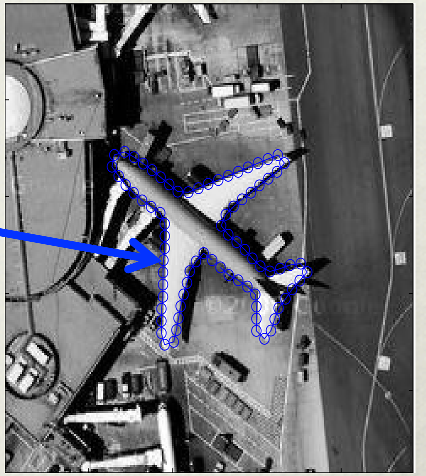


Example of MS Segmentation Without Shape!

- Difficult Cases: Clutter, Regions w/ non-uniform intensities, Occluded Objects
- Prior Must be compatible with Segmentation Models i.e. both can be minimized

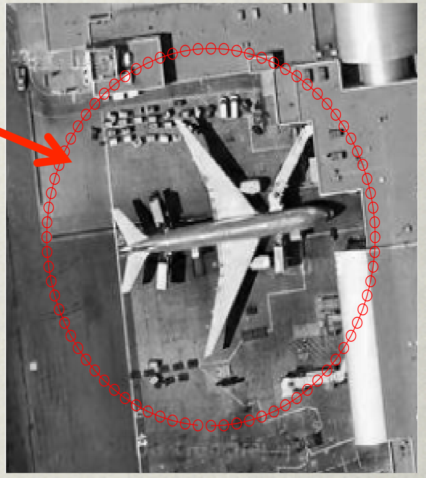
Cliques Shape Matching Energy

S Polygonal Rep. of a Reference Shape: $S = \{\vec{r}_i\}_{i=1}^{N}$



$[d_{ij}]$: Symmetric Matrix of Intervertex Distances $d_{ij} = |\vec{r}_i - \vec{r}_j|$

Σ an Evolving Polygonal Contour: $\Sigma = \{\vec{p}_j\}_{j=1}^{N}$



Shape Matching Energy:

$$\inf_{\Sigma,\, s} \left\{ E_c(\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_N, s) = \sum_{i,j=1}^{N} \left(|\vec{p}_i - \vec{p}_j|^2 - s d_{i,j}^2 \right)^2 \right\}$$

- ‘s’: scale parameter to be min’d over as well
- Invariance to Rigid Motion
- Scale Invariance

Proposed Model

$$E(\Sigma, c_1, c_2, s) = E_{MS} + E_C$$

$$= \min \left\{ \mathrm{Per}(\Sigma) + \lambda \int_{\mathrm{In}(\Sigma)} (c_1 - f)^2 \mathrm{dxdy} + \lambda \int_{\mathrm{Out}(\Sigma)} (c_2 - f)^2 \mathrm{dxdy} + \alpha \sum_{i,j=1}^{N} \left(|\vec{p}_i - \vec{p}_j|^2 - s d_{ij}^2 \right)^2 \right\}$$

α: shape strength

Σ: Evolving Polygonal Curve

Polygonal CV Model + Shape!

Best approx of ‘f’ in L^2 sense taking 2 values c_1 and c_2

While Enforcing Σ matches reference shape

Variations of the Perimeter Term

Perimeter Term:

$$\mathrm{Per}(\Sigma) = \sum_{j=1}^{N} \sqrt{(D^+x_j)^2 + (D^+y_j)^2} \qquad D^+\xi_j = \xi_{j+1} - \xi_j.$$

Variations of Perim. Term:

$$\frac{d}{dt}\mathrm{Per}(\Sigma) = -\sum_{j=1}^{N} \vec{v}_j^1 \cdot (\dot{x}_j, \dot{y}_j)\sqrt{(D^+x_j)^2 + (D^+y_j)^2}$$

$$\vec{v}_j^1 = \frac{1}{\sqrt{(D^+x_j)^2 + (D^+y_j)^2}}\left(D^-\left(\frac{D^+x_j}{\sqrt{(D^+x_j)^2 + (D^+y_j)^2}}\right), D^-\left(\frac{D^+y_j}{\sqrt{(D^+x_j)^2 + (D^+y_j)^2}}\right)\right)$$

ODE system describing time evolution of vertices:

$$\begin{aligned}\frac{d}{dt}(x_j(t), y_j(t)) &= \left(D^-\left(\frac{D^+x_j}{\sqrt{(D^+x_j)^2 + (D^+y_j)^2}}\right), D^-\left(\frac{D^+y_j}{\sqrt{(D^+x_j)^2 + (D^+y_j)^2}}\right)\right) \\ &= \vec{v}_j^{per}\end{aligned}$$

Variations of Data Fidelity Term

In gradient descent, the fidelity term contributes the velocity:

$$\vec{v}_j^2 = \lambda \left[(c_2 - f)^2 - (c_1 - f)^2 \right] \vec{n}_j$$

$\vec{n}_j$ the unit normal vector at j-th vertex:

$$\vec{n}_j = \frac{1}{\sqrt{(D^+ x_j)^2 + (D^+ y_j)^2}} \left(-D^+ y_j, D^+ x_j \right)$$

ODE system describing time evolution of vertices:

$$\begin{aligned} \frac{d}{dt}(x_j(t), y_j(t)) &= \lambda \left[(c_2 - f)^2 - (c_1 - f)^2 \right] \frac{1}{\sqrt{(D^+ x_j)^2 + (D^+ y_j)^2}} \left(-D^+ y_j, D^+ x_j \right) \\ &= \vec{v}_j^{fid} \end{aligned}$$

Variations of shape prior and scale parameter

Variations wrt Shape Prior Term:

$$\frac{\partial}{\partial \vec{p}_k}\left(\alpha \sum_{i,j=1}^{N} \left(|\vec{p}_i - \vec{p}_j|^2 \; - \; s\, d_{i,j}^2\right)^2\right) \;=\; 8\alpha \sum_{j=1}^{N} \left(|\vec{p}_k - \vec{p}_j|^2 \; - \; s\, d_{kj}^2\right)\left(\vec{p}_k - \vec{p}_j\right) \cdot \dot{\vec{p}}_k$$

In gradient descent, shape term contributes the velocity:

$$\vec{v}_k^{\,shape} \;=\; \dot{\vec{p}}_k \;=\; -\alpha \sum_{j=1}^{N} \left(|\vec{p}_k - \vec{p}_j|^2 \; - \; s d_{kj}^2\right)\left(\vec{p}_k - \vec{p}_j\right).$$

Optimization wrt scale parameter ‘s’ yields simple cond’n:

$$s \;=\; \frac{\sum_{i,j} |\vec{p}_i - \vec{p}_j|^2 d_{ij}^2}{\sum_{i,j} d_{ij}^4}$$

Update of constants c_1 and c_2

Variations wrt c_1 and c_2 in CV NRG yield:

$$c_1 = \frac{\int_\Sigma f\,dxdy}{|\Sigma|} \quad \text{and} \quad c_2 = \frac{\int_{\Sigma^c} f\,dxdy}{|\Sigma^c|}$$

Vector Form of Green's Theorem:

$$\int_\Sigma f\,dxdy = \int_{\partial\Sigma} F\,n_x d\sigma \quad \text{where} \quad \frac{\partial F}{\partial x} = f.$$

Procedure to get integral quant's:

- Integrate f in x-direction using Trap. rule to get primitive F so $\frac{\partial f}{\partial x} = F$
- Approximate area integral as follows:

$$\int_\Sigma f\,dxdy = \int_{\partial\Sigma} F\,n_x d\sigma \approx \sum_{j=1}^{N} \frac{1}{2}\left[F(x_j, y_j) + F(x_{j+1}, y_{j+1})\right] \frac{D^+ y_j}{\sqrt{(D^+ x_j)^2 + (D^+ y_j)^2}}$$

$|\Sigma|$ can also be found simply by using formula with $f \equiv 1$

Point is: Can obtain area integrals via boundary integrals of polygonal curve accurately!

Final Proposed Algorithm

Complete Gradient Descent Curve Evolution ODE:

$$\dot{\vec{p}}_k = \left[\left(\vec{v}_k^{per} + \vec{v}_k^{fid} + \vec{v}_k^{shape}\right) \cdot \vec{n}_k\right] \vec{n}_k$$

Explicit Gradient Descent Time Marching Scheme:

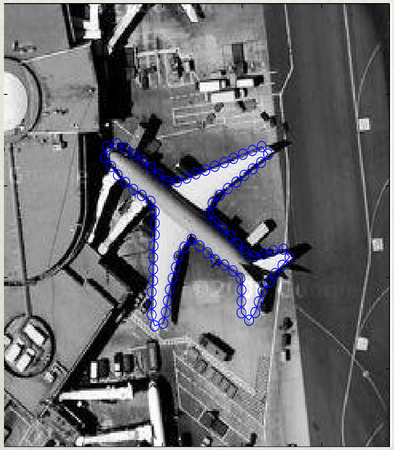
$$\vec{p}_k^{n+1} = \vec{p}_k^{n} + dt \left[\left(\vec{v}_k^{per} + \vec{v}_k^{fid} + \vec{v}_k^{shape}\right) \cdot \vec{n}_k\right] \vec{n}_k$$

Algorithm:

1. Calculate Shape Signature for Ref. shape
2. Initialize Polygonal Curve
3. Update constants c_1 and c_2 via explicit formulas
4. Calculate Scale Match by explicit formula
5. Evolve curve by above time marching scheme

Repeat: Steps 3 to 5 until reach fixed contour (steady state)

Shape Prior Segmentation Example



Learned Ref Shape

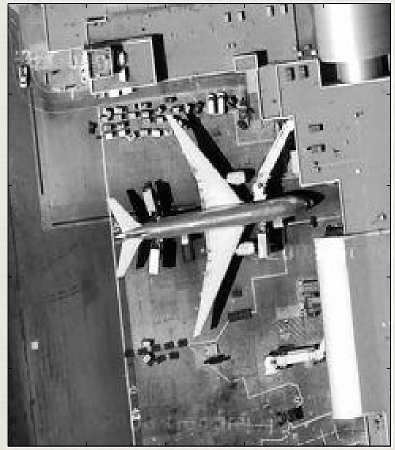
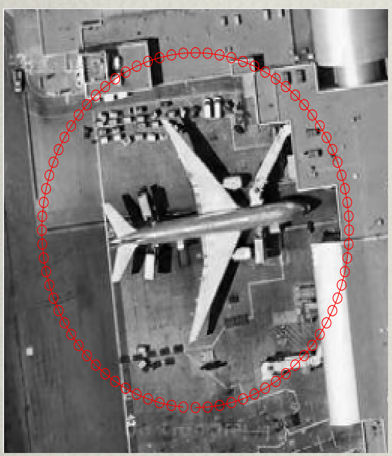
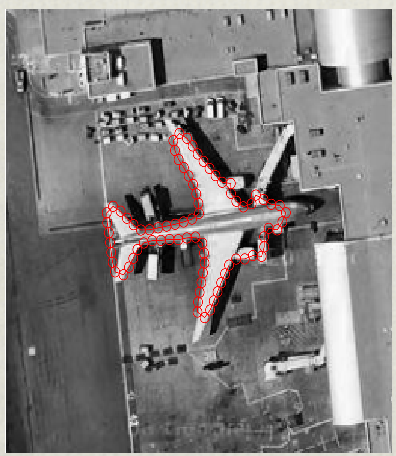
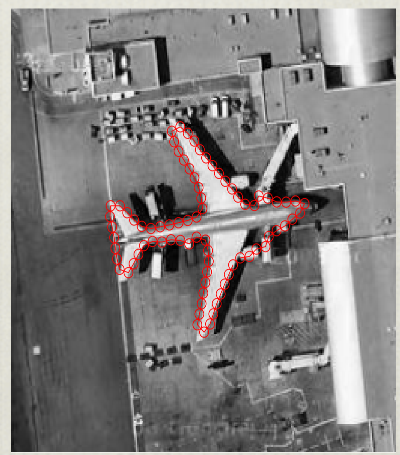


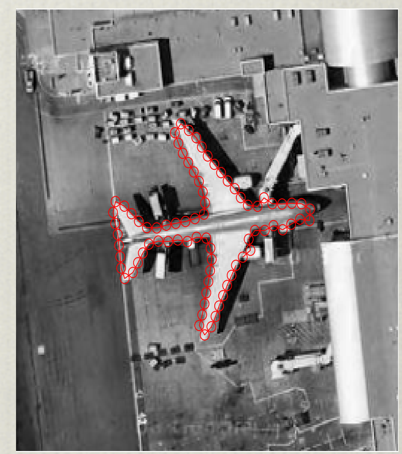
Image to be Segmented

Prior Juxt'd on Image









Final Seg'd. Image!

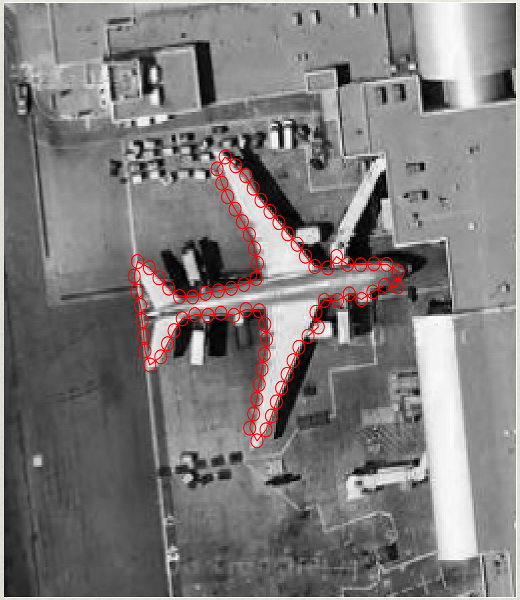
Increasing shape strength

$\alpha = 0.1$

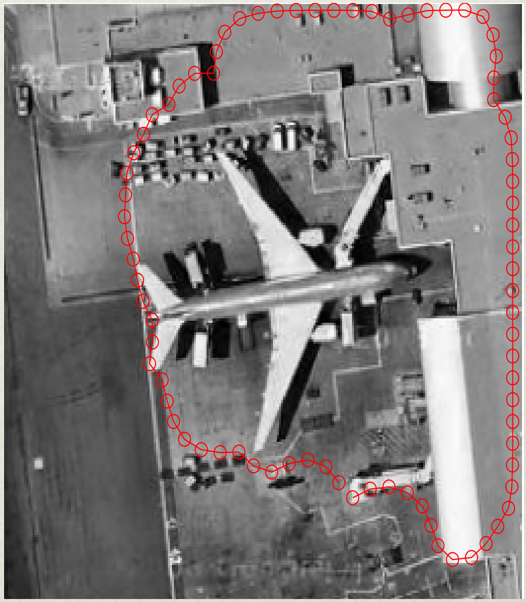
$\alpha = 0.5$

$\alpha = 1.0$

Shape Prior Segmentation



With Shape



Without Shape

Disocclusion

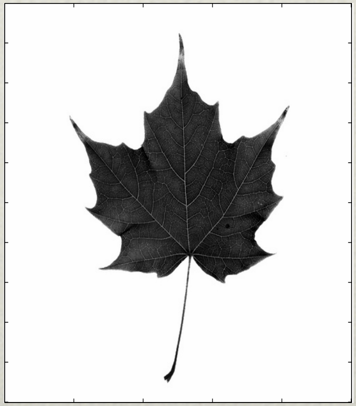
$$E(\Sigma, c_1, c_2, s) = E_{MS} + E_C$$

$$= \min \left\{ \mathrm{Per}(\Sigma) + \lambda_R \int_{\mathrm{In}(\Sigma)} (c_1 - f)^2 \mathrm{dxdy} + \lambda_R \int_{\mathrm{Out}(\Sigma)} (c_2 - f)^2 \mathrm{dxdy} + \alpha \sum_{i,j=1}^{N} \left(|\vec{p}_i - \vec{p}_j|^2 - s d_{ij}^2 \right)^2 \right\}$$

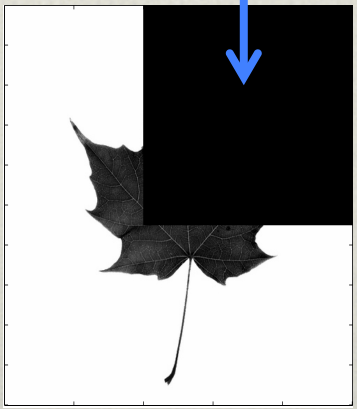
$$\lambda_R = \begin{cases} 0 & \text{if } \vec{x} \in R \\ \lambda & \text{otherwise.} \end{cases}$$

R: Occluded Region

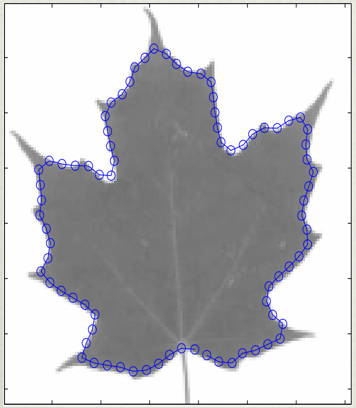
Don't fit data in R



Occluded Region: R

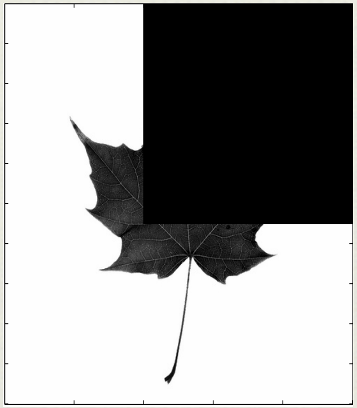


Disocclusion Example



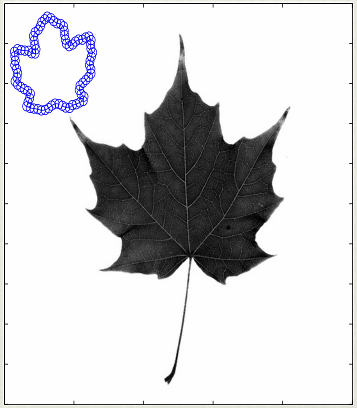
144x144

Learned reference shape

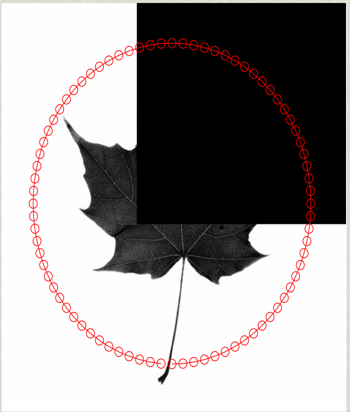


500x500

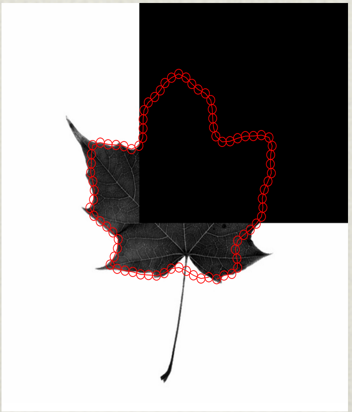
Occluded Image



Scale difference



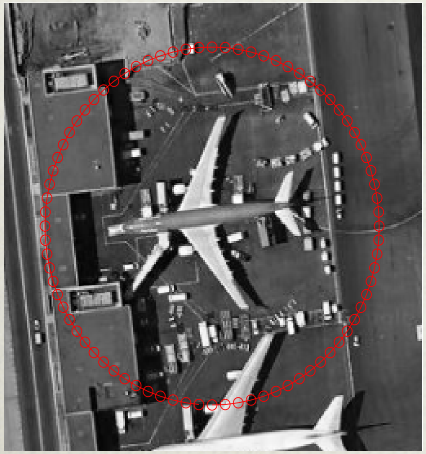
Initial contour



Final evolved contour

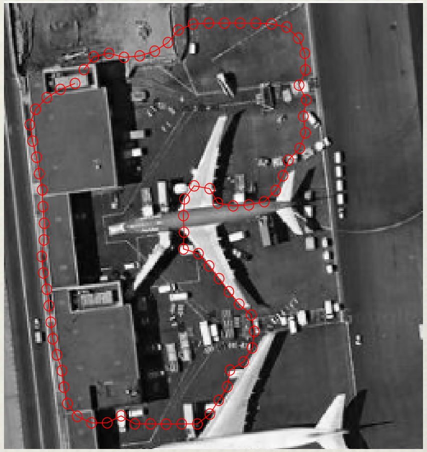
Disoccluded contour w/ true image

Shape Prior Segmentation: Very Difficult Case



Initial curve

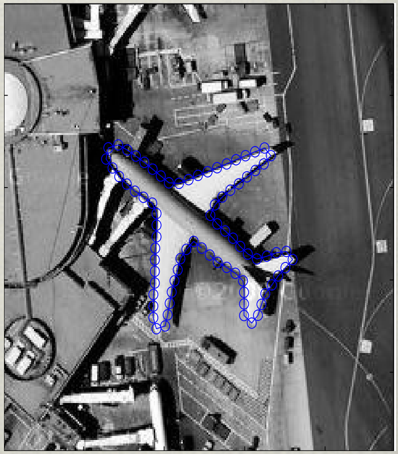
With Prior



No Prior!

- Fuselage matches tarmac.
- 2 completely different intensities in plane

Very difficult segmentation!



prior

Ongoing and Future Work

- Convergence proof of the DCA algorithm for proposed model
- Other ways of achieving/exploiting directional sparsity
- Shape Prior Segmentation: Modeling both occluders and shapes
- Neural Network → semantic segmentation. Interplay between a trained model and a mathematical one. Not a 2 step approach but a synergistic one.
- Using CNN's for applications to spatially varying blind deconvolution.
- Primal Dual methods for Neural Network Optimization
- Convex relaxation techniques for NN's. Some work done on quantizing weights. Xi et al. '18

Thank You for Your Attention!

Thank You for Your Attention!

Basic idea in classical active contours

Curve evolution and deformation (internal forces):

$$\text{Min } \mathit{Length(C)} + \mathit{Area(inside(C))}$$

Boundary detection: stopping edge-function (external forces)

$$g \geq 0, \quad g \downarrow, \quad \lim_{t \to \infty} g(t) = 0$$

Example:

$$g(|\nabla u_0|) = \frac{1}{1 + |\nabla G_\sigma * u_0|^p}$$

Snake model (Kass, Witkin, Terzopoulos '88)

$$\inf_C F(C) = \int_0^1 |C'(s)|^2 ds + \lambda \int_0^1 g(|\nabla I(C(s))|) ds$$

Geodesic model (Caselles, Kimmel, Sapiro '95)

$$\inf_C F(C) = 2\int_0^1 |C'(s)| g(|\nabla I(C(s))|) ds$$

Limitations

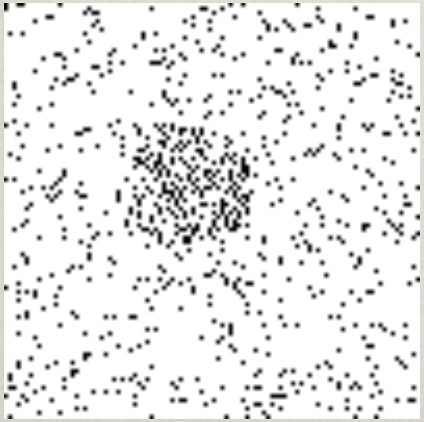
- detects only objects with sharp edges defined by gradients

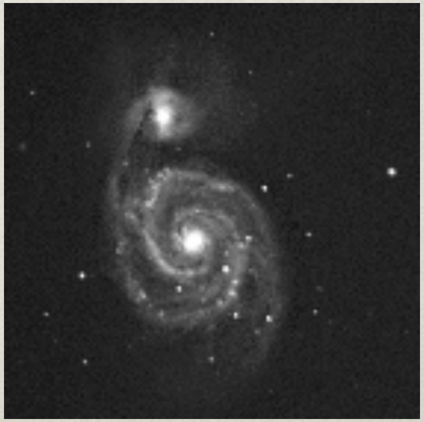
- the curve can pass through the edge

- smoothing may miss edges in presence of noise

- not all can handle automatic change of topology

Examples





Mumford-Shah Image Segmentation ('89)

$$\min_{u,\,K\subseteq\Omega}\left\{MS(u,K) := \int_{\Omega\setminus K}|\nabla u|^2\,dx + \mu Length(K) + \lambda\int_{\Omega}(u-f)^2dx\right\}$$

- K: finite union of curves (set of edges)
- u: -piecewise smooth approx. to f
  -smooth in each connected compt. of $\Omega\setminus K$
  -jumps allowed across curves in K

❖ Early variational model for image segmentation

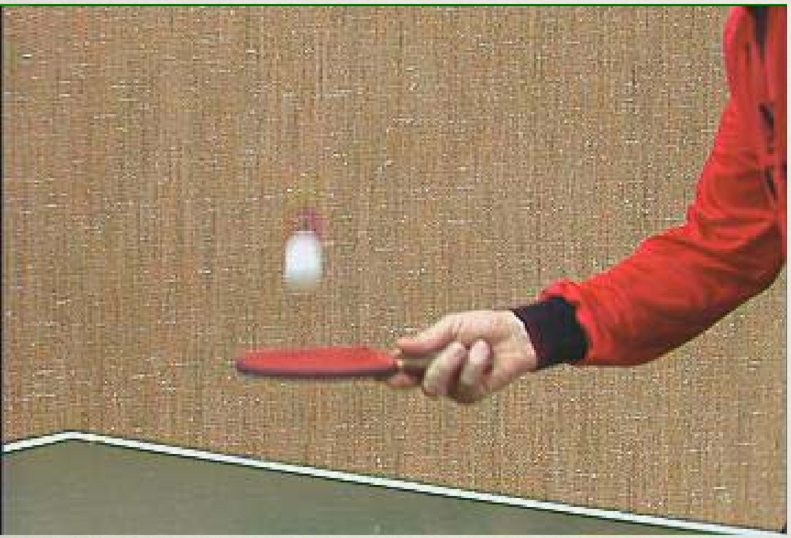
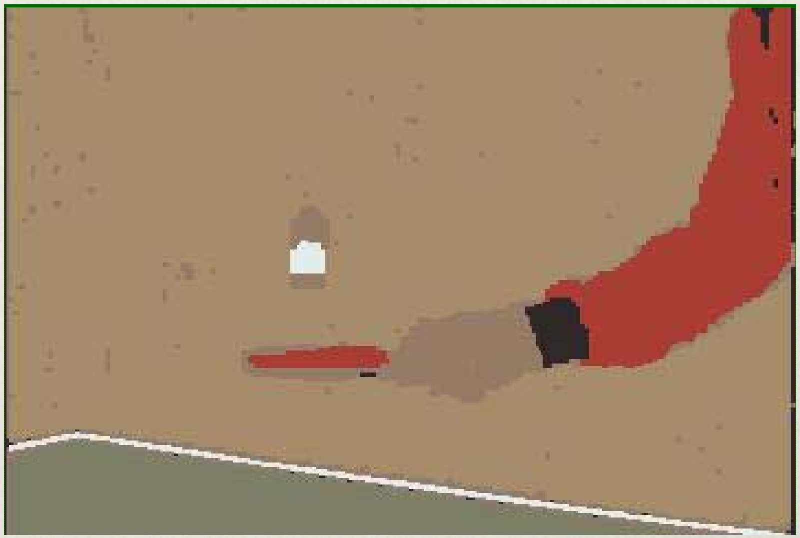
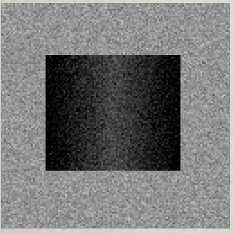
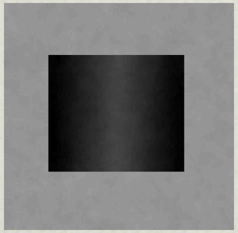
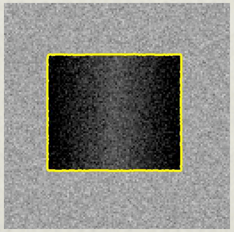
❖ u fits given image f by p.wise smooth regions w/ sharp edge boundaries K

❖ Non-Convex Model!

❖ Still many approaches to minimize the energy by approximation cf. level set methods, global convexification methods, etc.

❖ Full Model is "Overkill" for most applications

Mumford-Shah Segmentation Examples

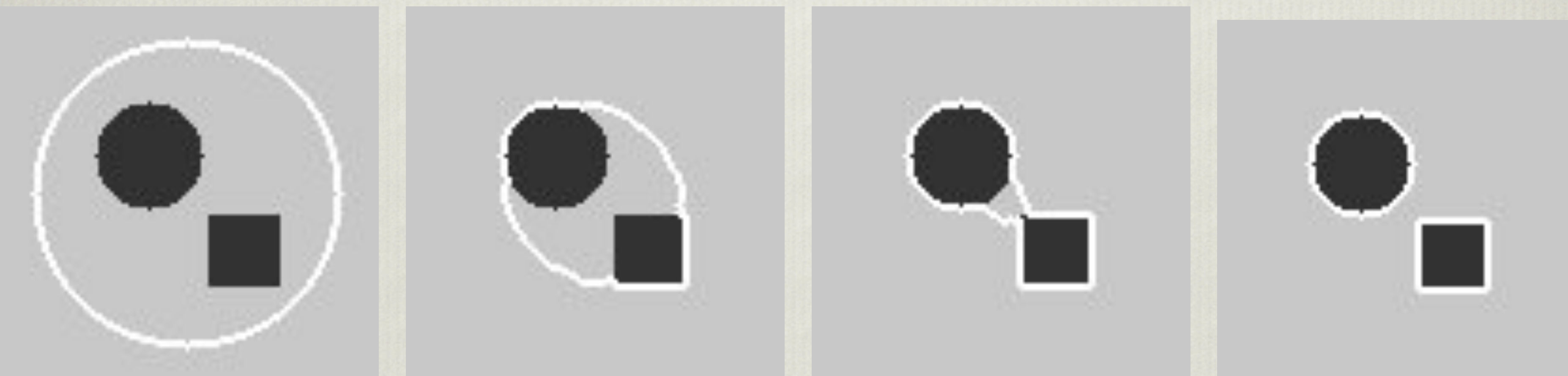

Application: Active Contours

– given an image $f:\Omega \rightarrow \Re$

– evolve a curve C to detect objects in f

– the curve has to stop on the boundaries of the objects

Initial Curve ⟶ Evolutions ⟶ Detected



Basic idea in classical active contours

Curve evolution and deformation (internal forces):

$$\text{Min } \; Length(C) + Area(inside(C))$$

Boundary detection: stopping edge-function (external forces)

$$g \geq 0, \quad g \downarrow, \quad \lim_{t \to \infty} g(t) = 0$$

Example: $$g(|\nabla u_0|) = \frac{1}{1 + |\nabla G_\sigma * u_0|^p}$$

Snake model (Kass, Witkin, Terzopoulos '88)

$$\inf_C F(C) = \int_0^1 |C'(s)|^2 ds + \lambda \int_0^1 g(|\nabla I(C(s))|) ds$$

Geodesic model (Caselles, Kimmel, Sapiro '95)

$$\inf_C F(C) = 2\int_0^1 |C'(s)| \, g(|\nabla I(C(s))|) ds$$