The active geometric shape model: A new robust deformable shape model and its applications

Computer Vision and Image Understanding, December 2012

Quan Wang, Kim L. Boyer

Signal Analysis and Machine Perception Laboratory

Department of Electrical, Computer, and Systems Engineering

Rensselaer Polytechnic Institute

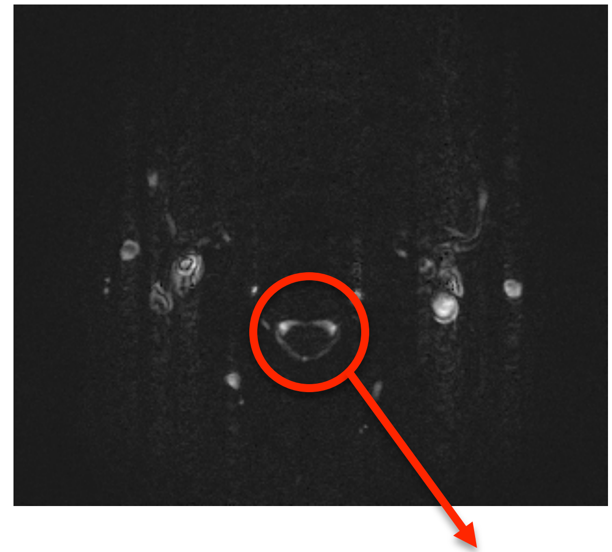
Motivation

- How do we detect and segment the "donut-shaped" cerebrospinal fluid (CSF) from MR images?

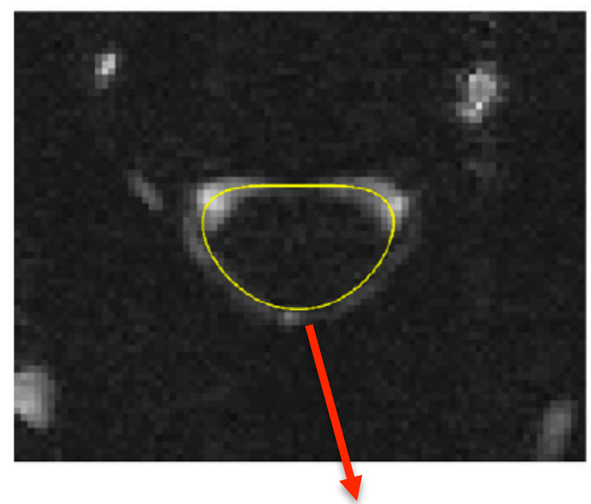
- Challenges:
  - Inhomogeneous intensity distribution
  - Limited training data

- Constraints:
  - The shape of the CSF is always a distorted ellipse (degree of freedom = 5)

$$\begin{cases} x = x_c + a\cos\theta \\ y = y_c + b\left(1 - (1 - \sin\theta)^p\right) \end{cases}$$



CSF



distorted ellipse

Option 1: Model-based object detection?

- Active Contour Model (Snakes)
  - The only assumption is **smooth closed** contour
  - No shape priors – can result in unacceptable shapes
- Active Shape Model (ASM)
  - Use statistics of point distribution
  - Need accurate annotation of landmark points
  - Need a large training dataset – not always available
- Active Appearance Model (AAM)
  - Statistics of point distribution + appearance

Option 2: Geometric shape fitting?

- Total least squares (TLS) and variants
  - Difficult to solve for complicated shapes
  - For sets of points, not suited for images

- Hough transform (HT) / generalized HT
  - Brute-force search on a high dimensional parameter space – cost increases exponentially when the number of parameters increases

Method	Outlier Rejection	Working for Images	Complicated Shapes		
			Difficulty to Design	Time	Space
TLS	No	No	Very Difficult	—	—
TLS RANSAC	Yes	No	Very Difficult	—	—
HT	Yes	Yes	Easy	$O(m^{p+1})$	$O(m^p)$
AGSM	Yes	Yes	Medium	$O(m^2p)$	$O(m)$

Image size: $m \times m$, number of parameters: p

Our solution: AGSM

- WHAT: a method for fitting a geometric shape in images
- WHY: detect an object, which can be represented by **parametric equations**
- HOW: we iteratively adjust shape parameters according to the force field → find optimal parameters
- APPLICATIONS:
  - line/circle/ellipse fitting, *etc.*
  - detect cerebrospinal fluid (CSF) in phase-contrast magnetic resonance (PC-MR) image sequences

Inspiration

- How do we find a good fit of circle in image?
  - For simplicity, assume we know the radius
- Throw a ring into flowing water, it will stabilize when all forces are balanced
  - Image gradients → flowing water
  - Stationary status → best fit
  - Balanced forces → the basic idea of our method

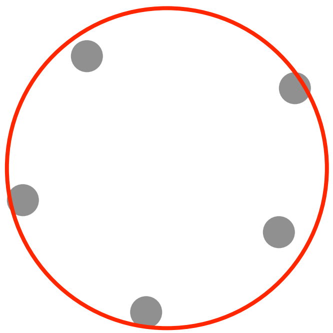
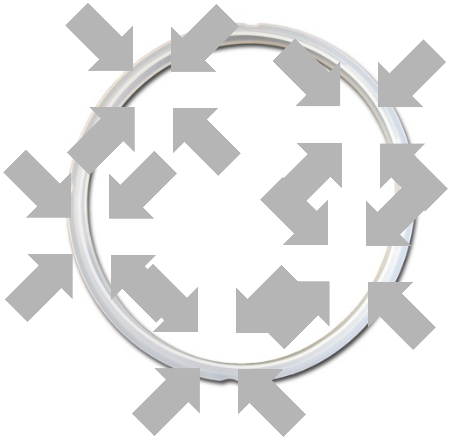


Image and circle



Flowing water and ring

Important concept: Force field

- To fit a deformable model (*e.g.* snakes, ASM), model points move along the *force field* in each iteration

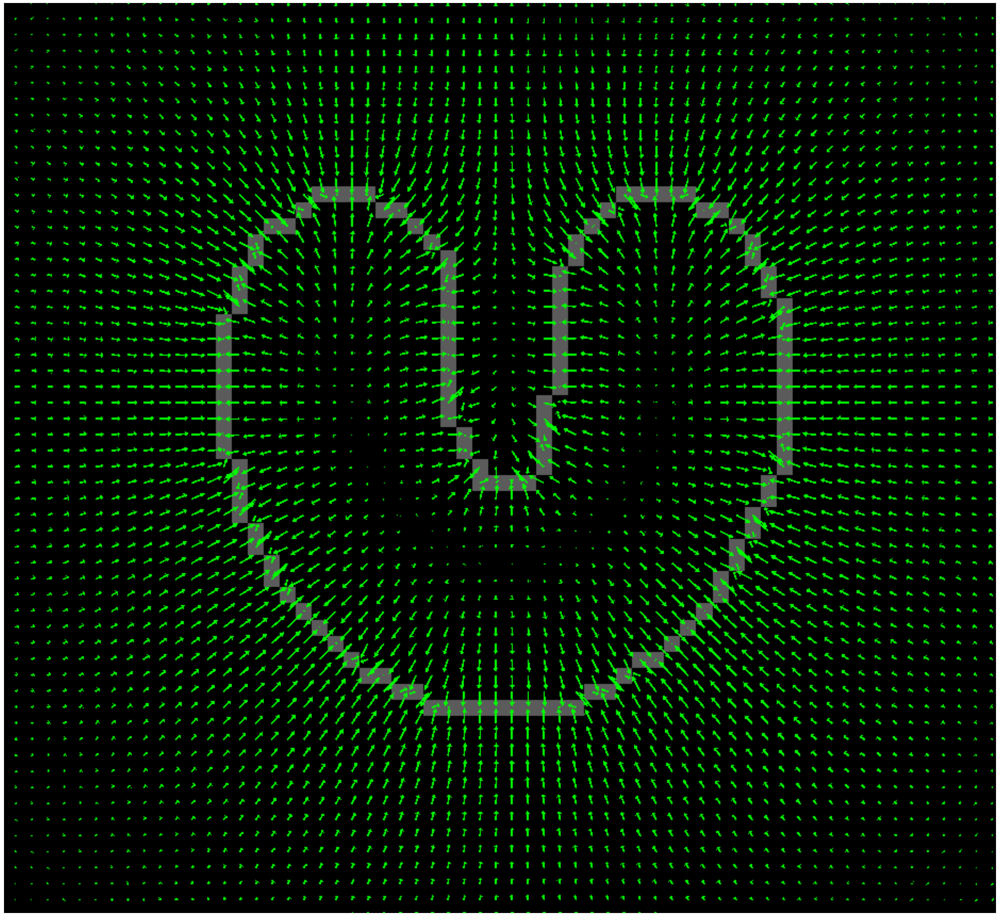
- A good force field needs to:
  1. Respect the image gradients
  2. Be smooth and have a large capture range

- Gradient vector flow (GVF) is most widely used:
  - GVF $\mathbf{v}(x,y) = [u(x,y), v(x,y)]$ minimizes an energy functional (f is the smoothed image)

$$\mathcal{E} = \iint \left(\mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \|\nabla f\|^2 \|\mathbf{v} - \nabla f\|^2 \right) \mathrm{d}x\mathrm{d}y$$

Deformable models and force field

- Biggest advantage of gradient vector flow (GVF) – large capture range



Basic idea of AGSM

1. We associate each parameter with a force or torque
   - *Force* for position/size/shape parameters
   - *Torque* for orientation parameters

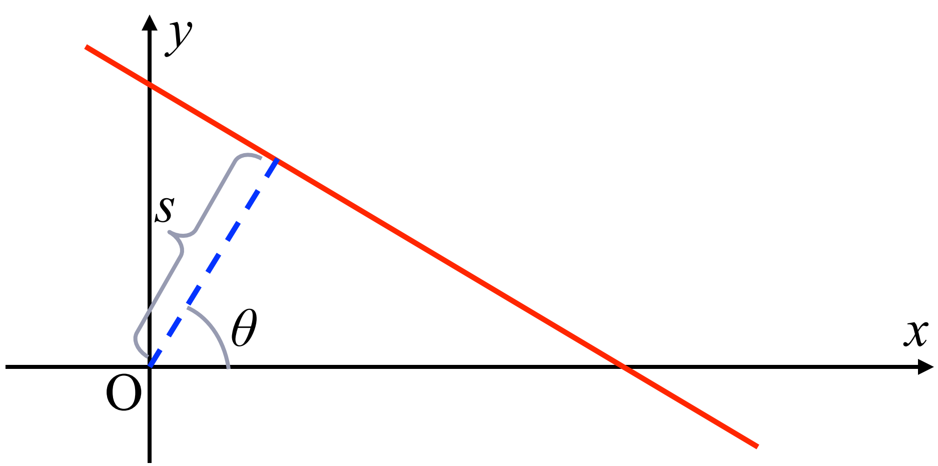
2. We adjust the parameter according to this force or torque

Example: Line-fitting

- Parametric equation for a line:

$$x \cos\theta + y \sin\theta - s = 0$$

- Two parameters: s and θ
- Geometric understanding:
  - s: the distance from the origin to the line
  - θ: the orientation
- Let the force field be $\mathbf{F}(x,y) = [F_x(x,y), F_y(x,y)]$

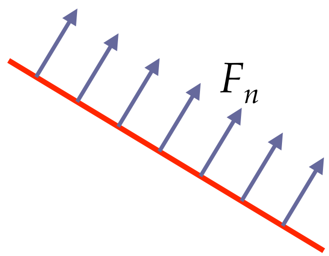


Example: Line-fitting (define the force)

- The normal force for parameter *s*:

The dot product indicates whether the force is pushing the line or pulling the line

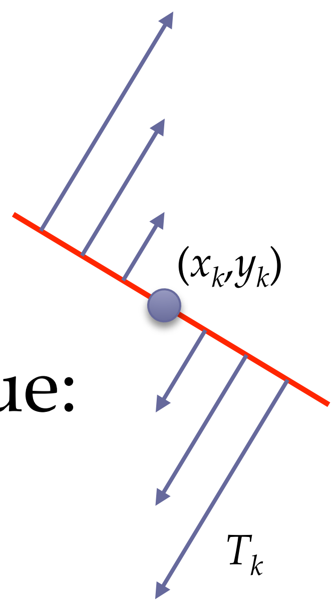
$$F_n = \frac{1}{N}\sum_{i=1}^{N} \mathbf{F}(x_i, y_i) \cdot \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$



- The torque around pivot point (x_k, y_k):

$$T_k = \frac{1}{N^2}\sum_{i=1}^{N} \mathrm{sgn}(k-i)\, d_{ik} \mathbf{F}(x_i, y_i) \cdot \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

$$d_{ik} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$$



- The k is selected to maximize the torque:

$$\tilde{k} = \arg\max_{k} |T_k|$$

Example: Line-fitting (update parameters)

- Parameters are updated according to the force/torque:

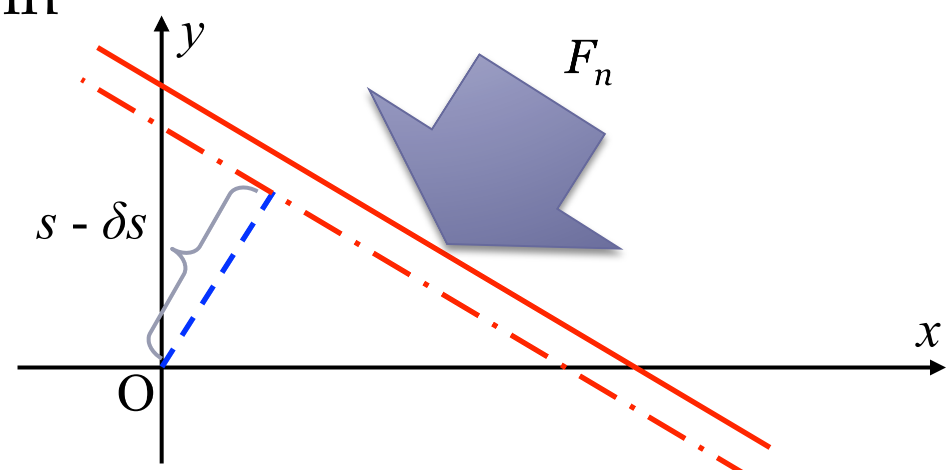
$$\begin{cases} s_{\text{new}} = s + \delta s & \text{if } F_n > t_s \\ s_{\text{new}} = s - \delta s & \text{if } F_n < -t_s \end{cases}$$

step size

threshold

$$\begin{cases} \theta_{\text{new}} = \theta - \delta\theta & \text{if } T > t_\theta \\ \theta_{\text{new}} = \theta + \delta\theta & \text{if } T < -t_\theta \end{cases}$$

- Explanation: if the force pushes the line towards the origin, then we change the parameters to move it closer to the origin



Generalization from the line example

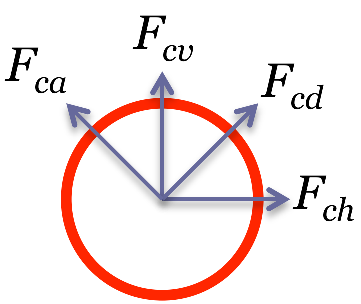
1. For each parameter, we define a force/torque for it according to its **geometric meaning**
   - This force/torque tends to directly change the value of this parameter
   - Number of parameters = degree of freedom
2. We adjust the parameter according to the **sign** of the force/torque
3. All parameters are adjusted in arbitrary order (order does not matter) in one iteration
4. After many iterations we get a good fit to the image

Fitting a circle

- Parametric equations:

$$\begin{cases} x = x_c + r\cos\theta \\ y = y_c + r\sin\theta \end{cases}$$

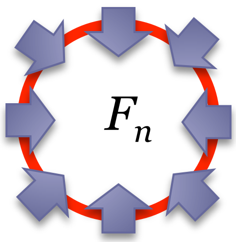
- For the center (x_c, y_c), we define horizontal (ch), vertical (cv), diagonal (cd), and anti-diagonal (ca) forces:



$$F_{ch} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{F}(x_i, y_i) \cdot [1, 0]^{\mathsf{T}}, \quad F_{cd} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{F}(x_i, y_i) \cdot \left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right]^{\mathsf{T}},$$

$$F_{cv} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{F}(x_i, y_i) \cdot [0, 1]^{\mathsf{T}}, \quad F_{ca} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{F}(x_i, y_i) \cdot \left[-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right]^{\mathsf{T}}.$$

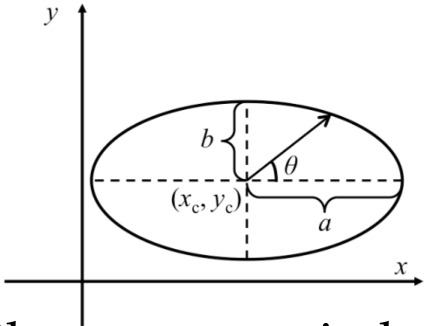
- For the radius *r*, we define the normal force:



$$F_n = \frac{1}{N}\sum_{i=1}^{N} \mathbf{F}(x_i, y_i) \cdot \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix}$$

The dot product indicates whether the force makes the circle expand or shrink

Fitting an ellipse in standard orientation



- Parametric equations:

$$\begin{cases} x = x_c + a\cos\theta \\ y = y_c + b\sin\theta \end{cases}$$

- The center (x_c, y_c) can be fitted in a similar way to a circle
- The force for the shape parameters a and b are defined on part of the ellipse:

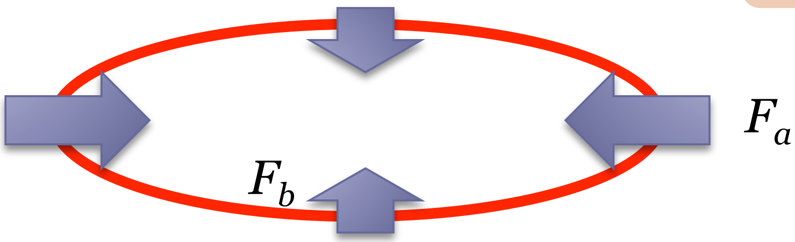
$$F_a = \frac{1}{N_a}\left(\sum_{\frac{3\pi}{4}<\theta_i<\frac{5\pi}{4}} \mathbf{F}(x_i, y_i)\cdot[1,0]^{\mathrm{T}} + \sum_{\theta_i<\frac{\pi}{4}\text{ or }\theta_i>\frac{7\pi}{4}} \mathbf{F}(x_i, y_i)\cdot[-1,0]^{\mathrm{T}}\right)$$

$$F_b = \frac{1}{N_b}\left(\sum_{\frac{5\pi}{4}<\theta_i<\frac{7\pi}{4}} \mathbf{F}(x_i, y_i)\cdot[0,1]^{\mathrm{T}} + \sum_{\frac{\pi}{4}<\theta_i<\frac{3\pi}{4}} \mathbf{F}(x_i, y_i)\cdot[0,-1]^{\mathrm{T}}\right)$$

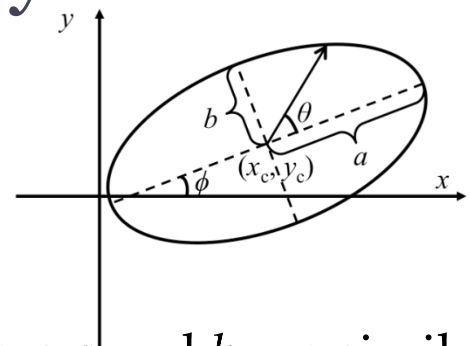
$$N_a = \sum_{\frac{3\pi}{4}<\theta_i<\frac{5\pi}{4}} 1 + \sum_{\theta_i<\frac{\pi}{4}\,\mathrm{or}\,\theta_i>\frac{7\pi}{4}} 1,$$

$$N_b = \sum_{\frac{5\pi}{4}<\theta_i<\frac{7\pi}{4}} 1 + \sum_{\frac{\pi}{4}<\theta_i<\frac{3\pi}{4}} 1.$$

Normalization numbers



Fitting an ellipse in arbitrary orientation



- Parametric equations:

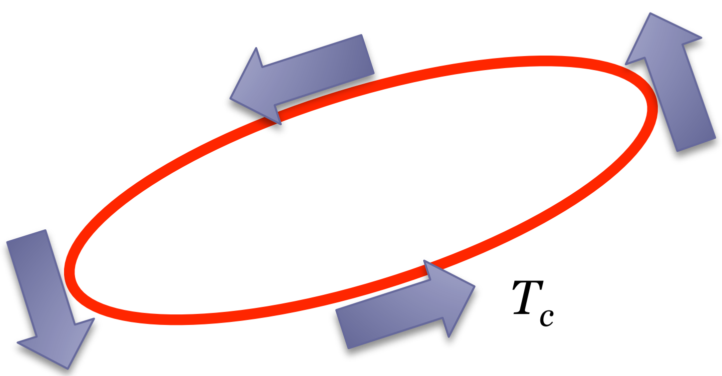
$$\begin{cases} x = x_c + a\cos\theta\cos\phi - b\sin\theta\sin\phi \\ y = y_c + a\cos\theta\sin\phi + b\sin\theta\cos\phi \end{cases}$$

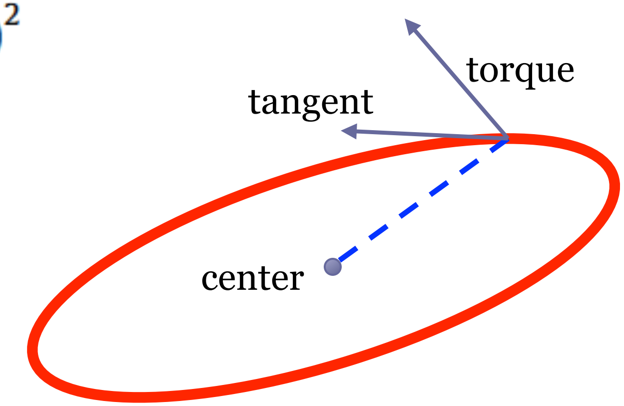
- The center (x_c, y_c) and the shape parameters a and b are similar to a standard ellipse
- The torque for the shape orientation ϕ:

$$T_c = \frac{1}{N^2}\sum_{i=1}^{N} d_i\,\mathbf{F}(x_i, y_i) \cdot \begin{bmatrix} -\sin(\theta+\phi) \\ \cos(\theta+\phi) \end{bmatrix}$$

$$d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$$

The dot product can be thought of something similar to a shear stress, but not necessarily in a tangent direction!





Fitting a distorted ellipse

- Parametric equations ($p > 1$):

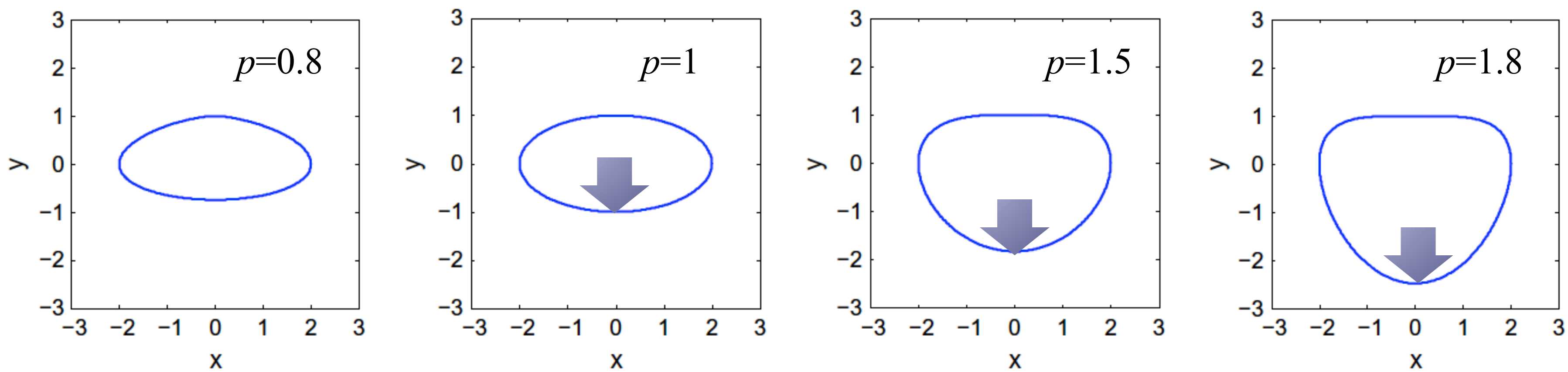
$$\begin{cases} x = x_c + a\cos\theta \\ y = y_c + b\left(1 - (1 - \sin\theta)^p\right) \end{cases}$$

This is the problem that motivated this work

- The force for the distortion parameter p:

$$F_p = \frac{1}{N_p} \sum_{\frac{11\pi}{8} < \theta_i < \frac{13\pi}{8}} \mathbf{F}(x_i, y_i) \cdot [0,\ 1]^{\mathrm{T}}$$

Defined on the lower part (the most protruding part) of the shape



Fitting a cubic spline contour

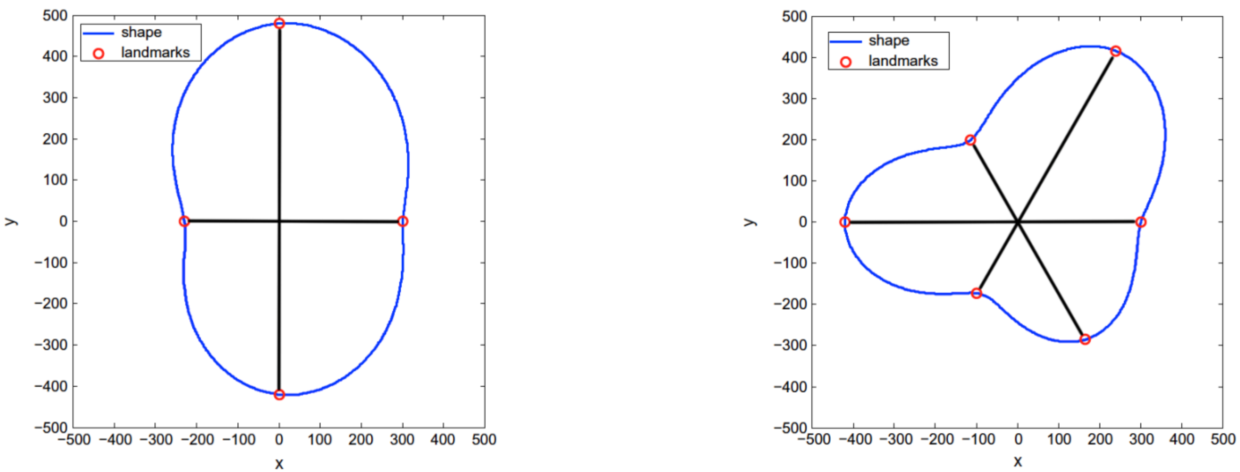
- Shape is obtained by cubic spline interpolation using N_{lm} landmark points:

$$\begin{cases} x_{P_k} = x_c + D_k \cos \Theta_k \\ y_{P_k} = y_c + D_k \sin \Theta_k \end{cases} \qquad \Theta_k = (k-1)\frac{2\pi}{N_{lm}}$$

- Parameters: (x_c, y_c) and $D = (D_1, D_2, \ldots, D_{Nlm})$
- Force for D_k:

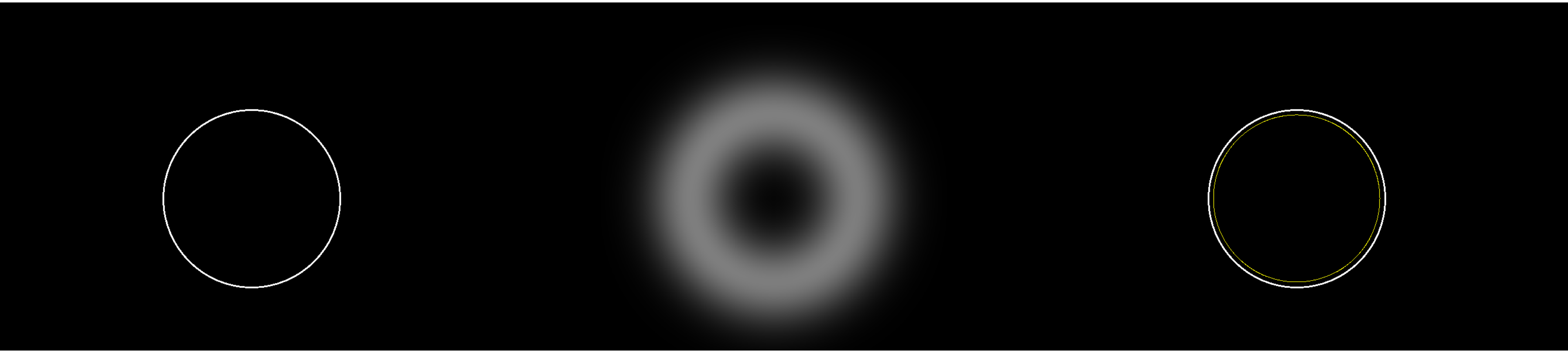
Dot product defined on local arc: expand or shrink

$$F_{D_k} = \frac{1}{N_{D_k}} \sum_{\Theta_k - \frac{\pi}{N_{lm}} < \theta_i < \Theta_k + \frac{\pi}{N_{lm}}} \mathbf{F}(x_i, y_i) \cdot [\cos\theta_i, \sin\theta_i]^{\mathrm{T}}$$



Correction of curvature

- To increase the capture range of the force field, the gradient is computed on the **smoothed** version of the image (standard practice)
- This smoothing operation dislocates the local maxima (where the model converges to) from original positions

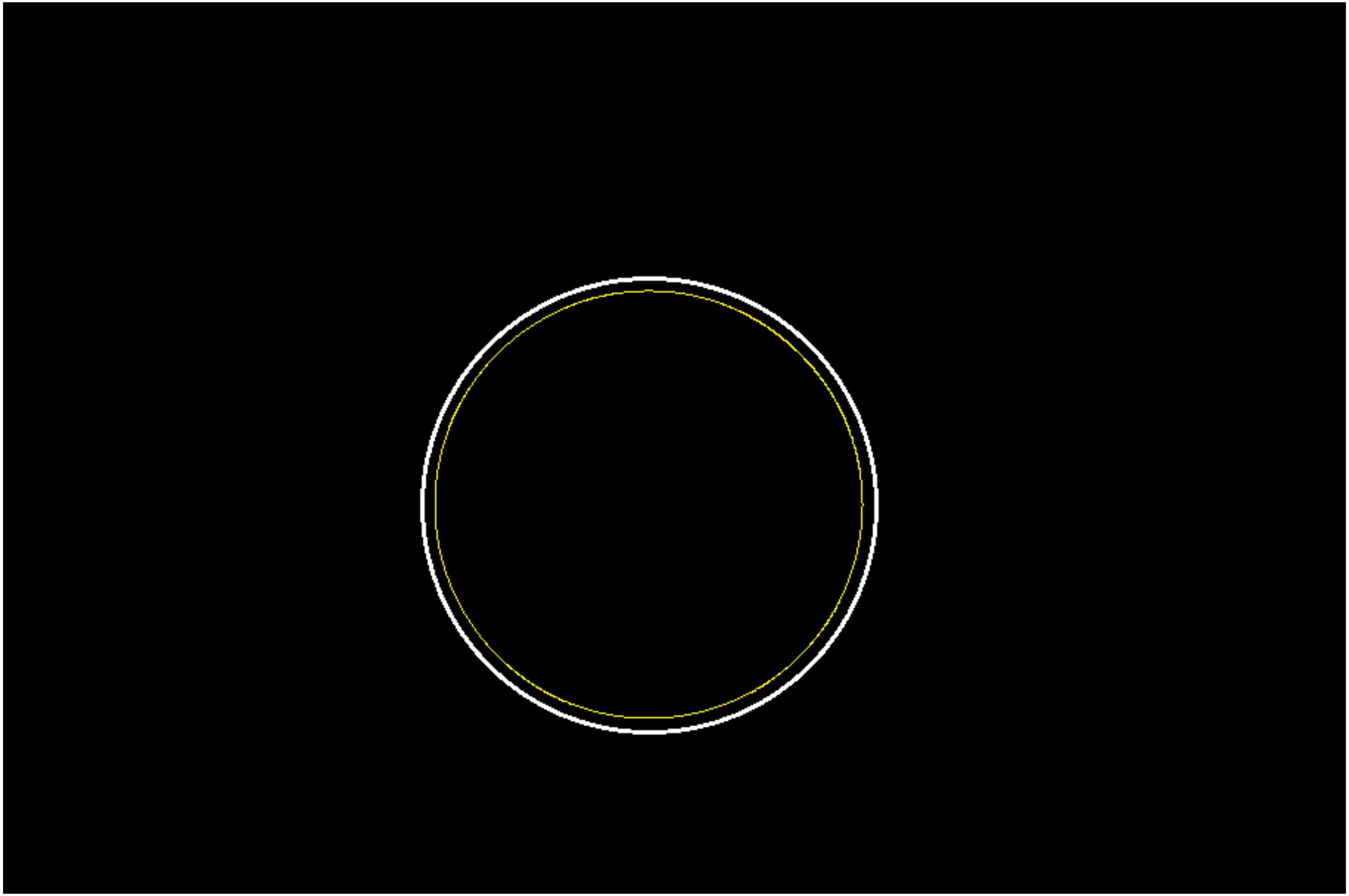


A circle

The Gaussian smoothed circle (enhanced for visualization)

The local maxima of the smoothed circle are on a smaller circle (yellow)

Correction of curvature



Correction for a circle

- In the polar coordinate system (ρ,θ), we define a disk with radius R as $M(\rho,\theta) = U(R - \rho)$, where $U(\bullet)$ is the unit step

convolution

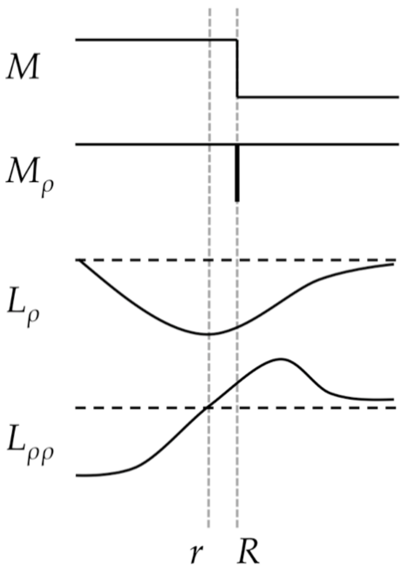
- The convolution with Gaussian kernel $G_\sigma(\rho,\theta)$ is $L(\rho,\theta) = G_\sigma * M$

standard deviation

- The derivative of M in the radial direction is $M_\rho = - \delta(R - \rho)$
- Based on the work of Bouma *et al.* (PAMI 2005), we can compute the first order and second order derivatives of $L(\rho,\theta)$:

$$L_\rho(\rho,\theta) = G_\sigma * M_\rho = -\frac{R}{\sigma^2} e^{-\frac{R^2+\rho^2}{2\sigma^2}} I_1\left(\frac{\rho R}{\sigma^2}\right)$$

$$L_{\rho\rho}(\rho,\theta) = e^{-\frac{R^2+\rho^2}{2\sigma^2}} \left(-\frac{R^2}{\sigma^4} I_0\left(\frac{\rho R}{\sigma^2}\right) + \left(\frac{\rho R}{\sigma^4} + \frac{R}{\rho\sigma^2}\right) I_1\left(\frac{\rho R}{\sigma^2}\right)\right)$$



- $I_n(\bullet)$ is the modified Bessel function of the first kind

Correction for a circle

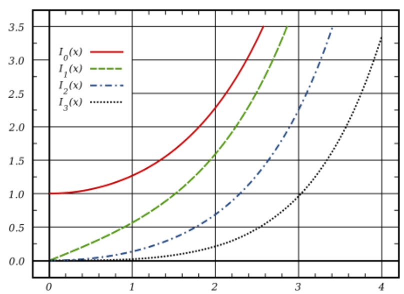
- If $L_{\rho\rho}(r,\theta) = 0$, then r is the dislocated radius of the disk $M(\rho,\theta) = U(R - \rho)$ whose true radius is R

M: disk
M_ρ: circle
$M_{\rho\rho}$: derivative of circle
L: smoothed disk
L_ρ: smoothed circle
$L_{\rho\rho}$: derivative of smoothed circle

- The equation $L_{\rho\rho}(r,\theta) = 0$ can be rewritten as:

$$\frac{R}{\sigma^2} I_0\left(\frac{rR}{\sigma^2}\right) = \left(\frac{r}{\sigma^2} + \frac{1}{r}\right) I_1\left(\frac{rR}{\sigma^2}\right)$$

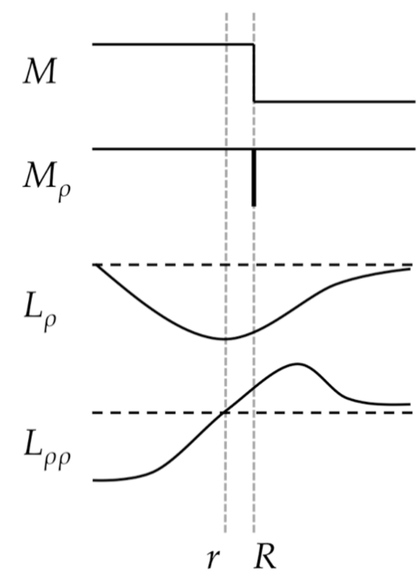
- We solve for $R = \Omega(r,\sigma)$ using numeric iterations:



$$R^{(k+1)} = \left(r + \frac{\sigma^2}{r}\right) \frac{I_1\left(\frac{rR^{(k)}}{\sigma^2}\right)}{I_0\left(\frac{rR^{(k)}}{\sigma^2}\right)}$$

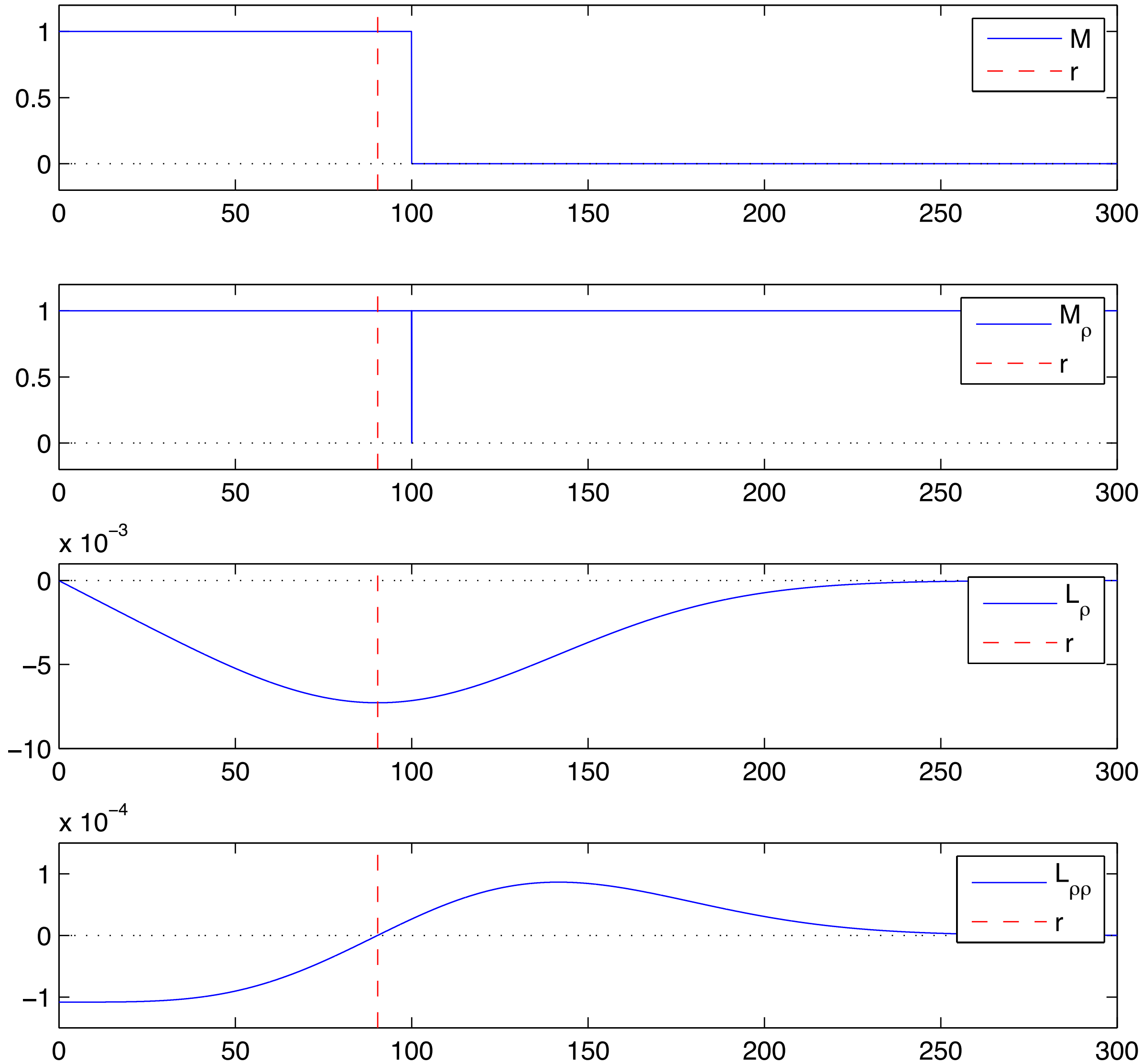
- When x is large, we make use of the fact:

$$\frac{I_1(x)}{I_0(x)} \approx \frac{128x^2 - 48x - 15}{128x^2 + 16x + 9}$$



Correction for a circle

- Example:
  - $R = 100$
  - $\sigma = 50$
  - $r = 90.42$



Correction for other shapes

- If the shape is not a circle, it is difficult to analytically determine the dislocation using equations of mathematical physics

- Thus we approximately make corrections according to <u>local curvature</u>

- Example – approximate correction for an ellipse
  - For an ellipse, we correct a and b for the curvature at $\theta = k\pi/2$
  - Let the solution of the equation for a circle be $R = \Omega(r,\sigma)$

$$\frac{b'^2}{a'} = R_1 = \Omega\left(\frac{b^2}{a}, \sigma\right)$$

$$\frac{a'^2}{b'} = R_2 = \Omega\left(\frac{a^2}{b}, \sigma\right)$$

$$a' = \sqrt[3]{R_2^2 R_1}$$

$$b' = \sqrt[3]{R_1^2 R_2}$$

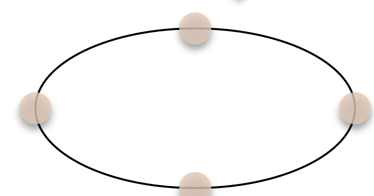
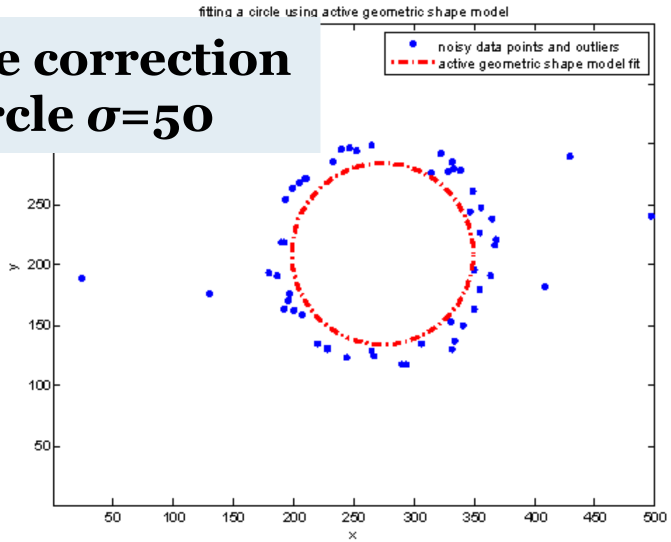
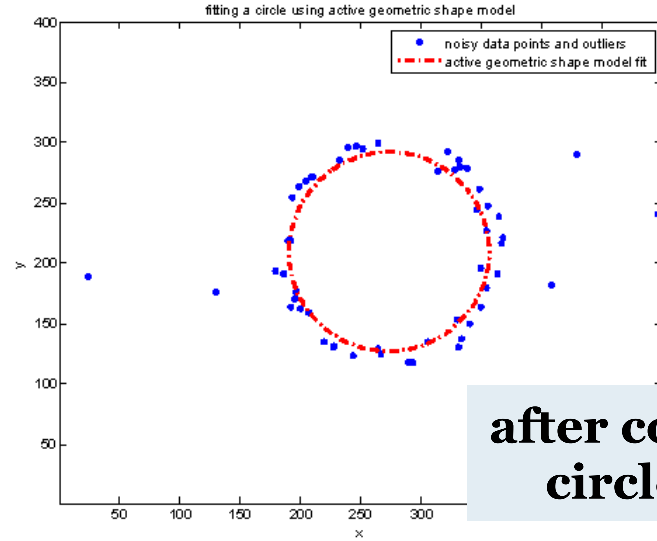


Fig. The 4 positions to be corrected.

Before and after correction of curvature

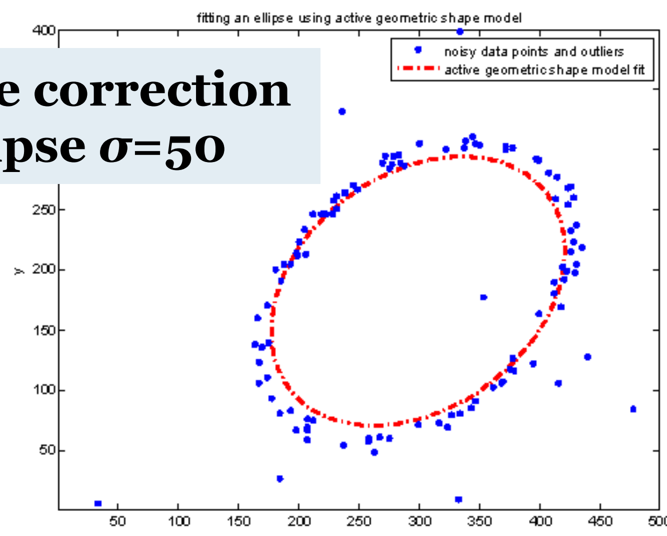
before correction
circle σ=50

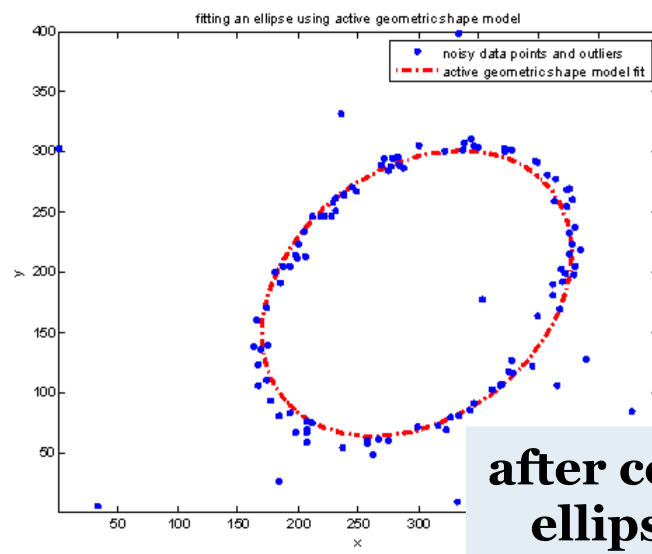




after correction
circle σ=50

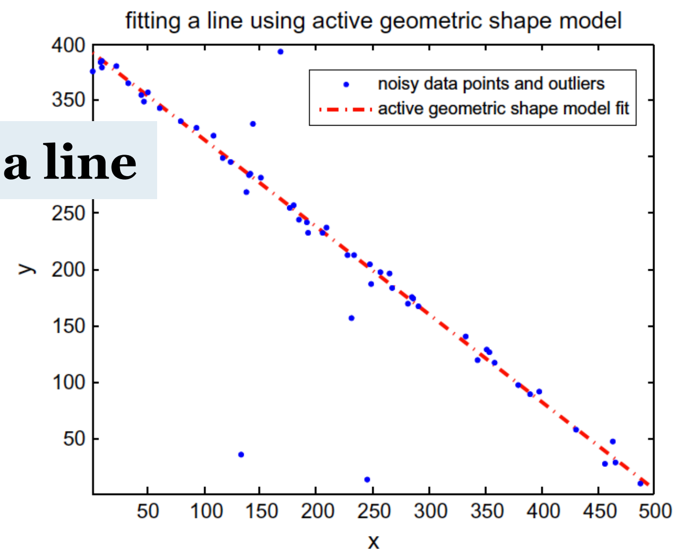
before correction
ellipse σ=50



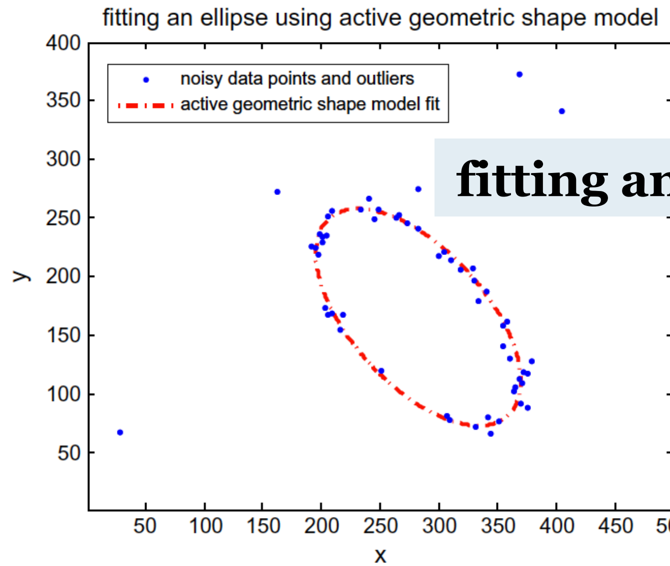


after correction
ellipse σ=50

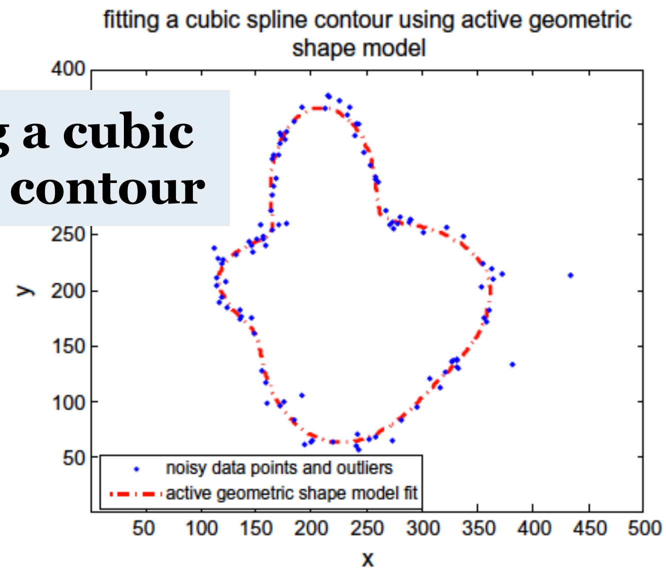
Experiments on synthetic data



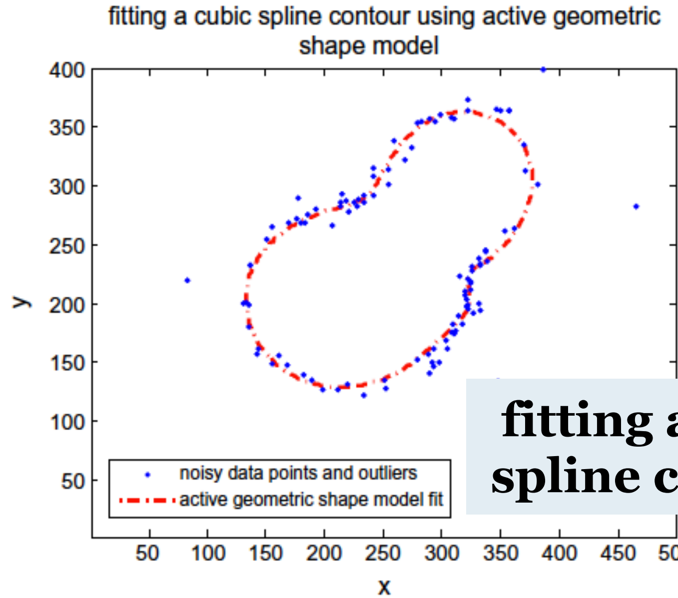
fitting a line



fitting an ellipse



fitting a cubic spline contour



fitting a cubic spline contour

Line fitting results (Quantitative)

- AGSM *vs.* TLS RANSAC *vs.* Hough
  - Similar precision performance
- Without RANSAC, TLS fails when outliers exist

Num of Noisy Data Points	Num of Outliers	Ground Truth		TLS		TLS RANSAC		Hough		AGSM	
		θ	s	$\overline{\lvert e_\theta \rvert}$	$\overline{\lvert e_s \rvert}$	$\overline{\lvert e_\theta \rvert}$	$\overline{\lvert e_s \rvert}$	$\overline{\lvert e_\theta \rvert}$	$\overline{\lvert e_s \rvert}$	$\overline{\lvert e_\theta \rvert}$	$\overline{\lvert e_s \rvert}$
50	0	4.8994	-149.74	0.0034	1.15	0.0043	1.48	0.0208	4.10	0.0050	1.36
50	0	2.4844	-72.04	0.0035	1.23	0.0046	1.40	0.0158	3.15	0.0098	2.96
50	5	0.9005	307.38	0.0162	3.61	0.0066	1.94	0.0127	2.19	0.0086	1.33
50	5	0.6776	311.15	0.0278	2.66	0.0134	1.90	0.0133	2.23	0.0141	1.28
50	10	4.8649	-163.15	0.0381	10.87	0.0155	4.27	0.0196	3.45	0.0080	2.63
50	10	2.6357	-125.99	0.0605	16.85	0.0198	5.42	0.0150	3.15	0.0186	5.27
100	0	1.4015	238.48	0.0029	0.61	0.0034	0.70	0.0140	1.84	0.0054	1.20
100	0	1.8678	111.63	0.0016	0.65	0.0023	0.82	0.0107	2.43	0.0082	2.58
100	5	4.1033	-305.15	0.0062	2.53	0.0042	1.88	0.0096	2.05	0.0093	1.31
100	5	0.8169	319.97	0.0102	2.31	0.0082	1.99	0.0125	2.17	0.0035	0.58
100	10	1.8975	106.32	0.0150	5.08	0.0094	3.59	0.0134	2.25	0.0118	3.63
100	10	3.0898	-242.20	0.0407	10.36	0.0231	5.63	0.0099	2.24	0.0126	2.75
100	20	3.3150	-271.14	0.0409	7.85	0.0169	3.01	0.0094	1.37	0.0270	2.92
100	20	0.3059	301.38	0.0605	6.95	0.0249	2.33	0.0099	1.41	0.0132	1.23

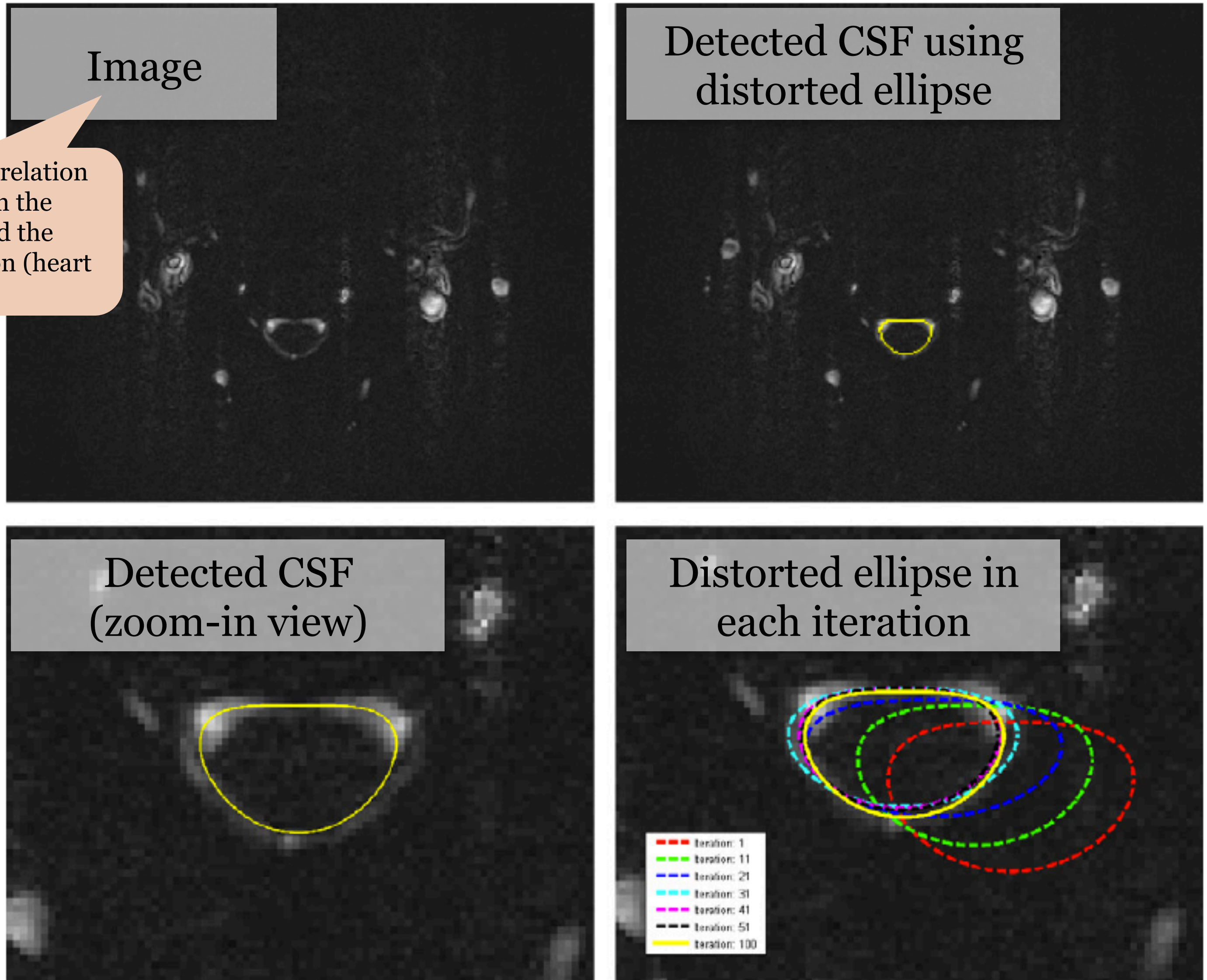
Circle fitting results (Quantitative)

- AGSM *vs.* Hough transform
  - When using **coarse** Hough space, AGSM has better precision
  - When using **fine** Hough space, similar precisions, but Hough transform is 80+ times slower
- For more complex shapes, TLS and Hough become impractical

Num of Noisy Data Points	Num of Outliers	Ground Truth			Hough			AGSM		
		x_c	y_c	r_t	$\overline{\lvert e_{x_c}\rvert}$	$\overline{\lvert e_{y_c}\rvert}$	$\overline{\lvert e_r\rvert}$	$\overline{\lvert e_{x_c}\rvert}$	$\overline{\lvert e_{y_c}\rvert}$	$\overline{\lvert e_r\rvert}$
50	0	259.0423	215.8965	71.1720	2.2500	2.2103	1.1828	1.6510	1.3337	0.6274
50	0	254.9968	193.0642	84.6669	2.0491	1.9064	1.4333	1.3233	0.9551	0.6203
50	5	249.1938	202.2778	88.9143	1.8194	2.6778	1.3671	1.1810	1.4203	0.8455
50	5	263.1093	183.7038	94.8626	1.7609	1.9889	1.5000	0.7396	1.1504	0.6053
50	10	242.0308	189.9690	72.0354	2.0938	2.2469	1.4894	1.1851	1.2668	0.7694
50	10	256.9920	211.0809	86.1565	2.0492	2.1162	1.5313	1.4601	1.2854	0.9584
100	0	254.6721	198.6342	56.8067	1.8672	1.4902	1.0273	0.9094	1.1451	0.5671
100	0	228.9261	224.1106	83.4873	1.8426	1.4000	1.3487	0.8498	0.8983	0.4390
100	5	245.2864	176.0669	77.1914	2.2359	1.1866	0.9766	1.2330	0.6408	0.5254
100	5	257.2192	178.4016	66.1663	1.1658	1.5902	1.3834	0.8856	1.2258	0.5022
100	10	260.0436	207.5514	70.7568	1.7044	1.7500	1.0500	0.9551	0.9379	0.5736
100	10	229.0649	198.4936	53.1994	1.5870	1.3513	1.0902	1.4342	1.2286	0.7037
100	20	271.7741	185.9095	55.6684	1.9000	2.0091	1.2158	0.9927	0.9933	0.6727
100	20	266.3963	222.4936	67.2908	1.9793	1.4994	1.0372	0.7522	0.8498	0.4914

Experiments on PC-MR images

Actually the correlation map between the sequence and the sinusoid function (heart cycle)

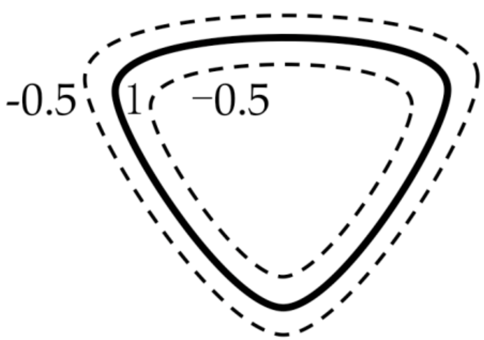


Experiments on PC-MR images

- Goodness measurement
  - We generate 50 seed shapes to evolve, and select the best fit
  - Goodness is measured by

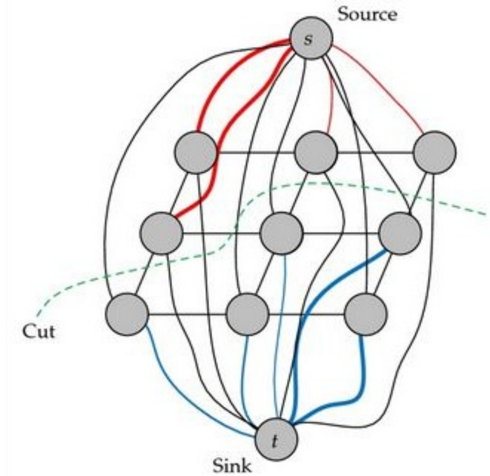
$$\mathcal{F}(\mathcal{P}) = \frac{1}{N}\sum_{i=1}^{N}\underbrace{\|\mathbf{F}(x_i, y_i)\|}_{\text{Current shape}} - \frac{1}{2N'}\sum_{i=1}^{N'}\underbrace{\|\mathbf{F}(x_i', y_i')\|}_{\text{Shrunken shape}} - \frac{1}{2N''}\sum_{i=1}^{N''}\underbrace{\|\mathbf{F}(x_i'', y_i'')\|}_{\text{Expanded shape}}$$

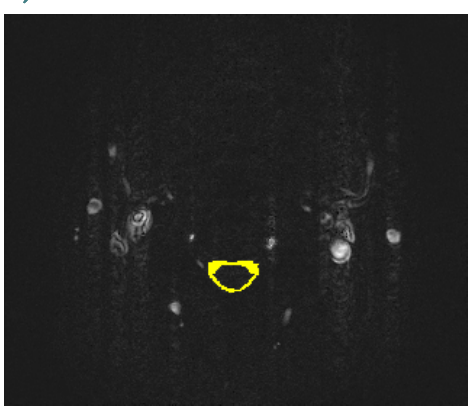
  - Smaller value is better



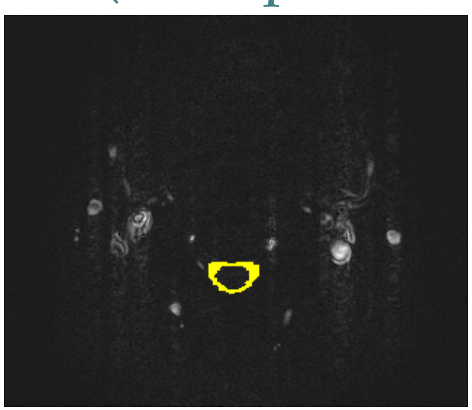
Experiments on PC-MR images

- CSF segmentation
  - Detection + Graph cuts → Segmentation
  - *s*-*t* graph cuts:
    - On the detected contour: positive seeds
    - Far away from the contour: negative seeds
  - We have achieved a mean Dice similarity coefficient (DSC) of 86.4% on our dataset (unsupervised!)





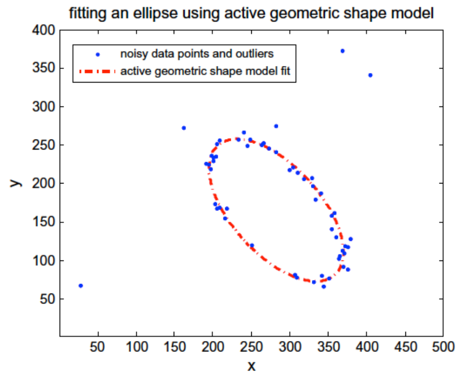
ground truth

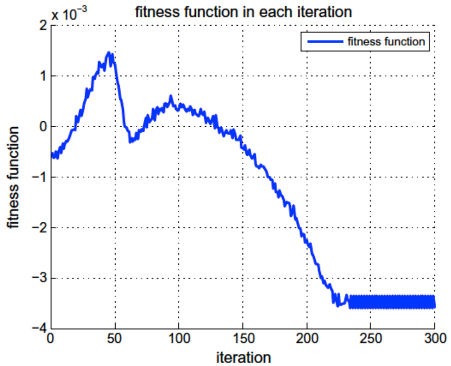


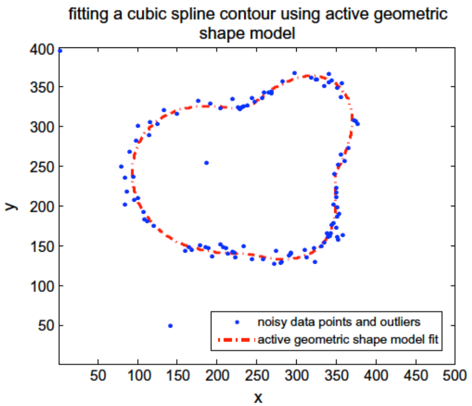
graph cuts segmentation

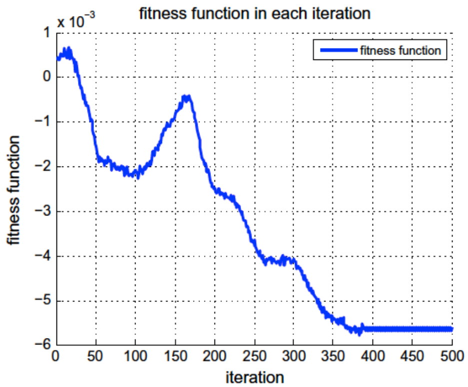
Difficulties of direct optimization methods

- Our AGSM method is heuristic (inspired by physics)
- AGSM iteratively adjusts parameters, and results in decreasing fitness function (better fit)







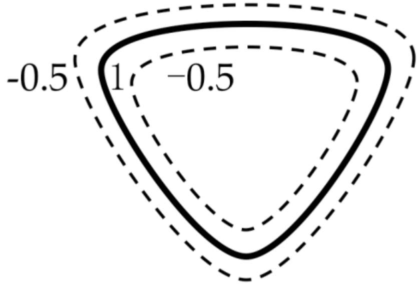


AGSM results in decreasing fitness function

Difficulties of direct optimization methods

- Question: Can we directly minimize the fitness function using **gradient descent** or **genetic algorithms**?

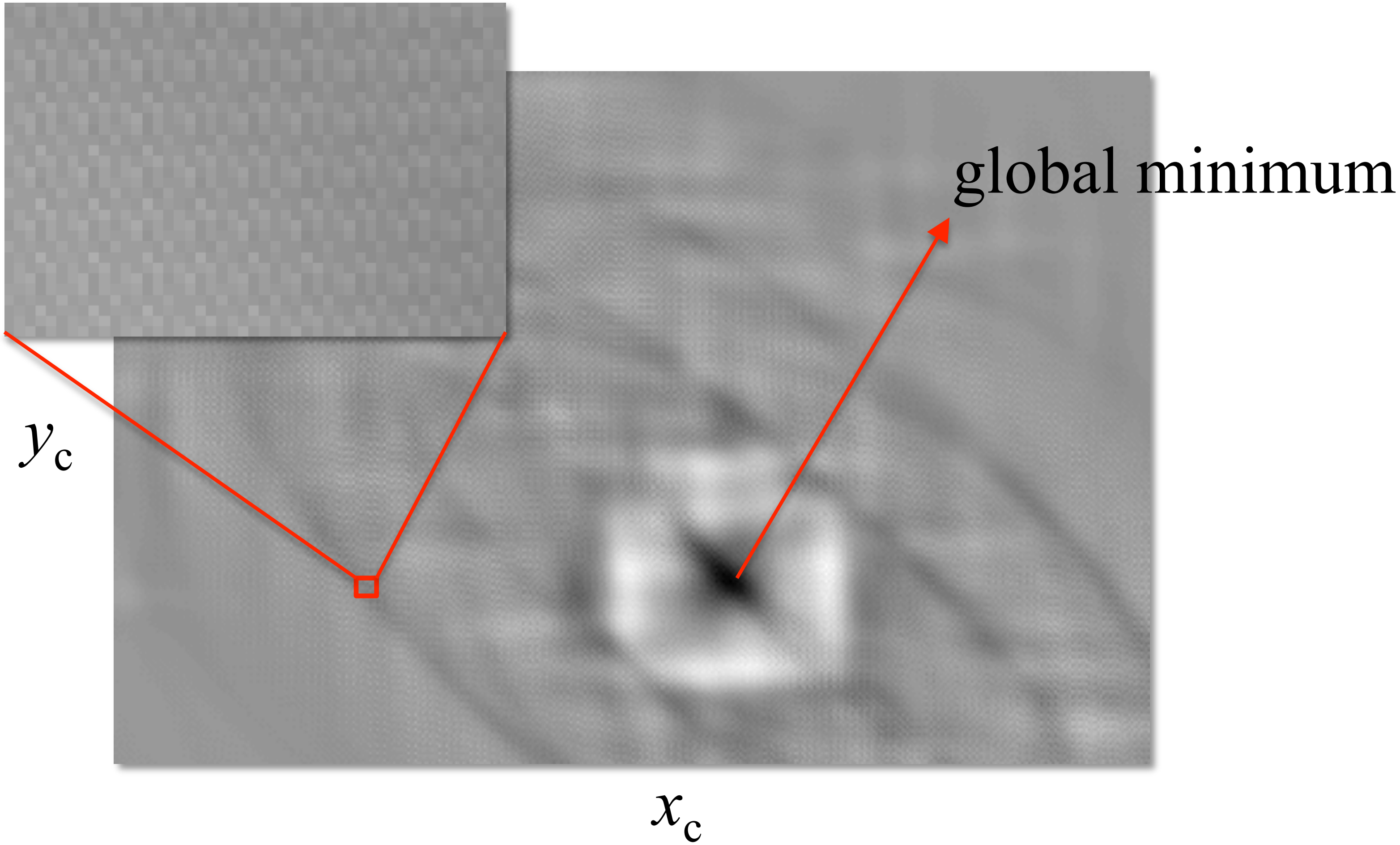
- Answer: It sounds feasible. But actually the fitness function:
  - Is not continuous
  - Is non-convex
  - Has local minima almost everywhere
  - Is slow to compute (render three shapes)



Evaluating the fitness function for one parameter combination requires the rendering of three shapes

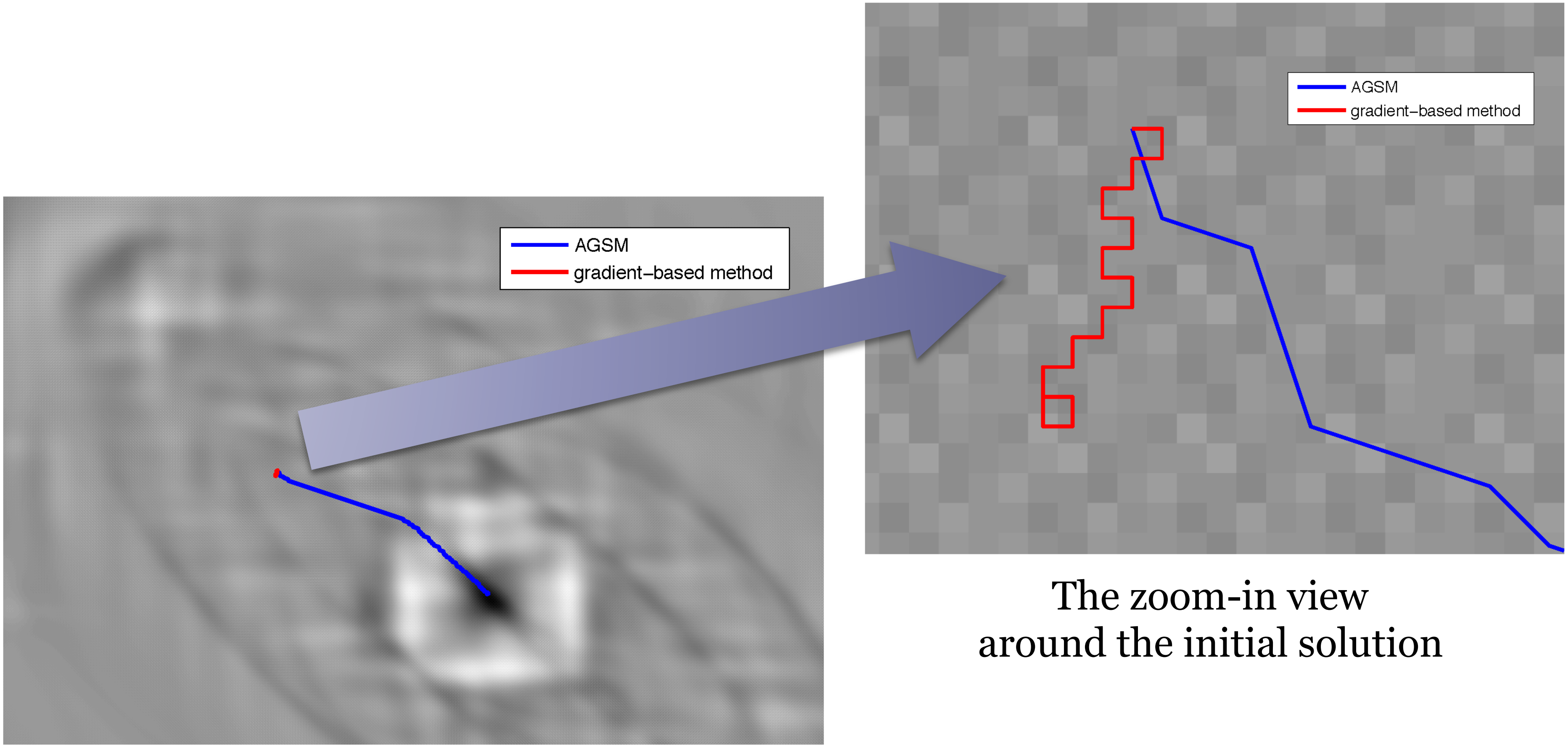
Fitness function of ellipse

- If we know the ground truths of a, b and φ
- The fitness function with respect to x_c and y_c:



Fitness function of ellipse

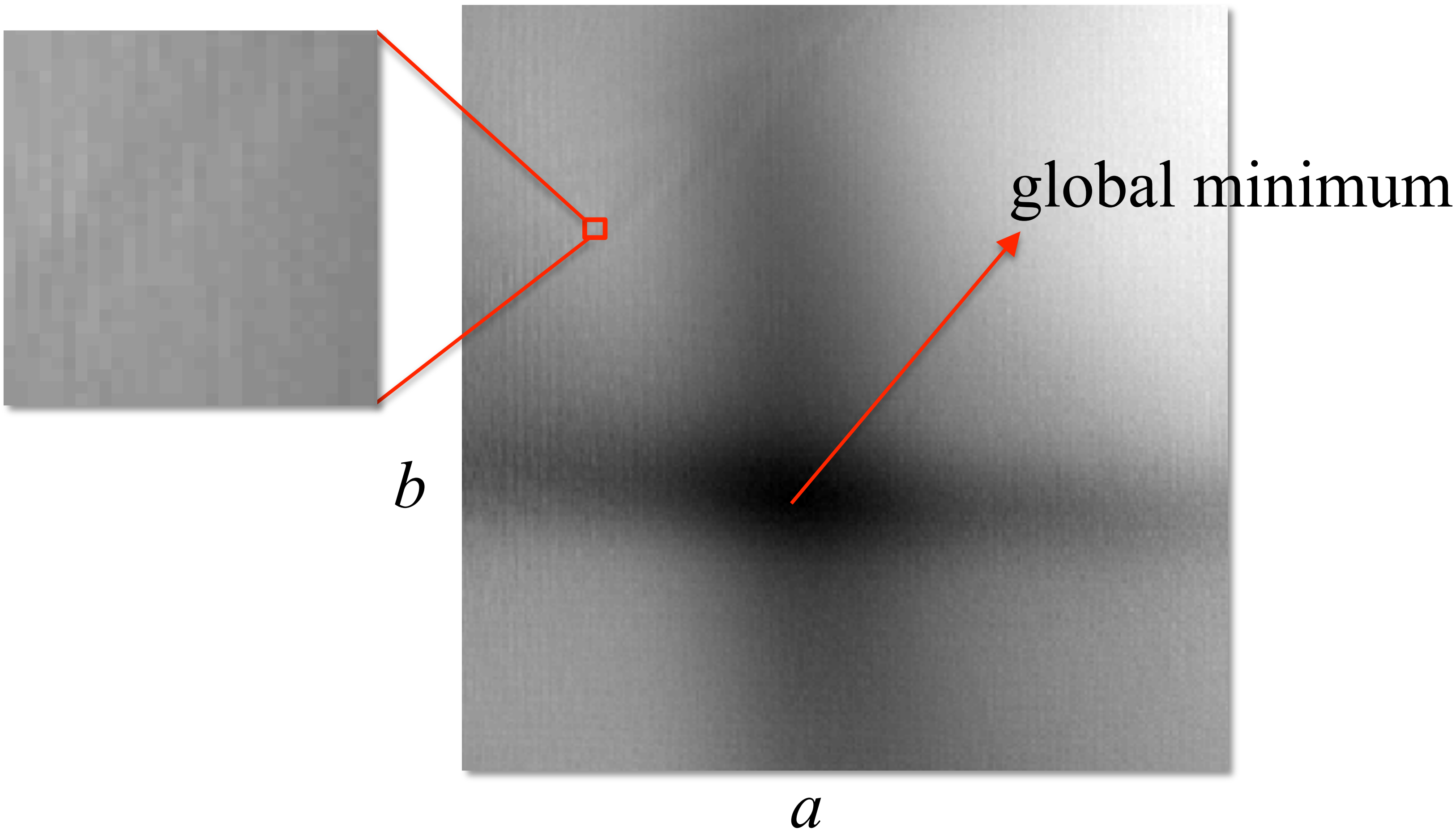
- The solution paths of AGSM and gradient-based method on the fitness map



The zoom-in view
around the initial solution

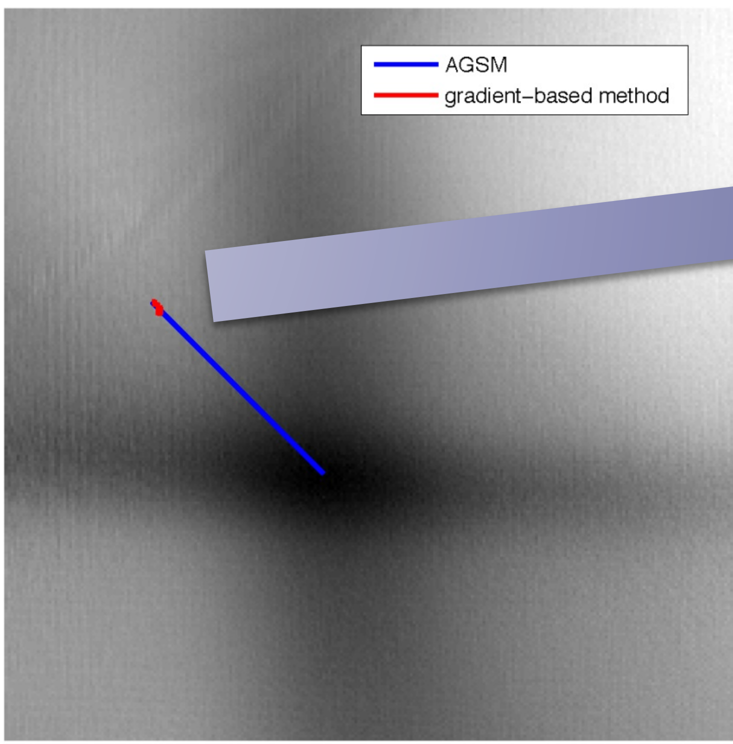
Fitness function of ellipse

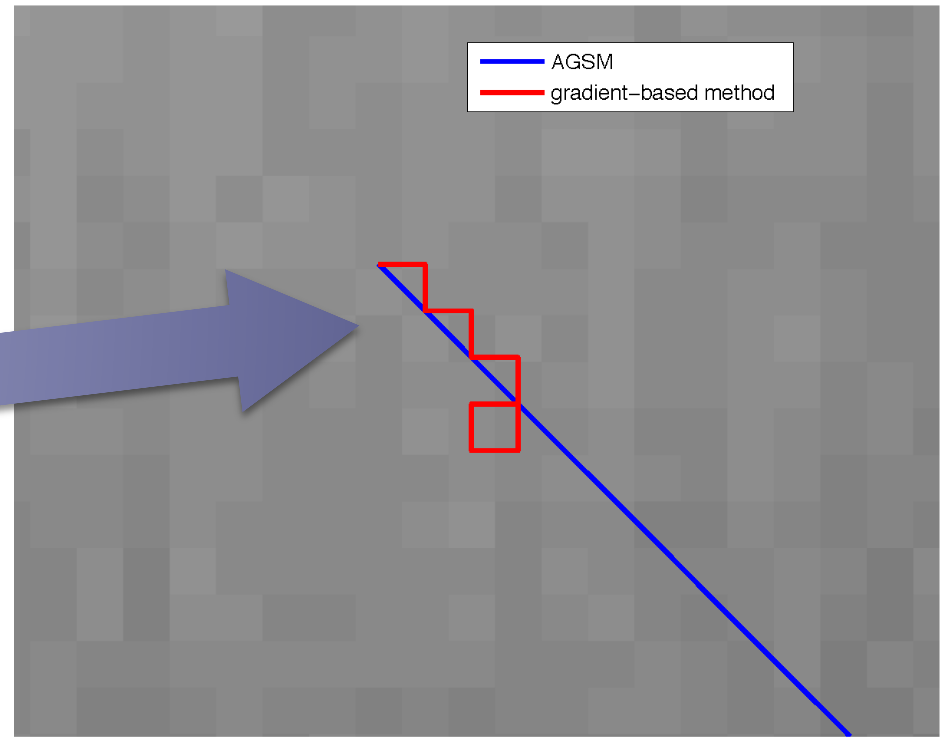
- If we know the ground truths of x_c, y_c and φ
- The fitness function with respect to a and b:



Fitness function of ellipse

- The solution paths of AGSM and gradient-based method on the fitness map





The zoom-in view
around the initial solution

Conclusion

- Our active geometric shape model (AGSM) is a novel and powerful approach to fit a geometric shape to image
- This model is validated on both synthetic data and PC-MR image sequences
- These slides are only a quick view of the work. For more technical details (some are very important) and more experiments, please look at our CVIU paper, and check our website:
  - https://sites.google.com/site/agsmwiki/

The active geometric shape model: A new robust deformable shape model and its applications

Computer Vision and Image Understanding, December 2012

Quan Wang, Kim L. Boyer

Signal Analysis and Machine Perception Laboratory

Department of Electrical, Computer, and Systems Engineering

Rensselaer Polytechnic Institute