计算机辅助手术讲座（15）

Image Guided Surgery (15)

Deformable Models

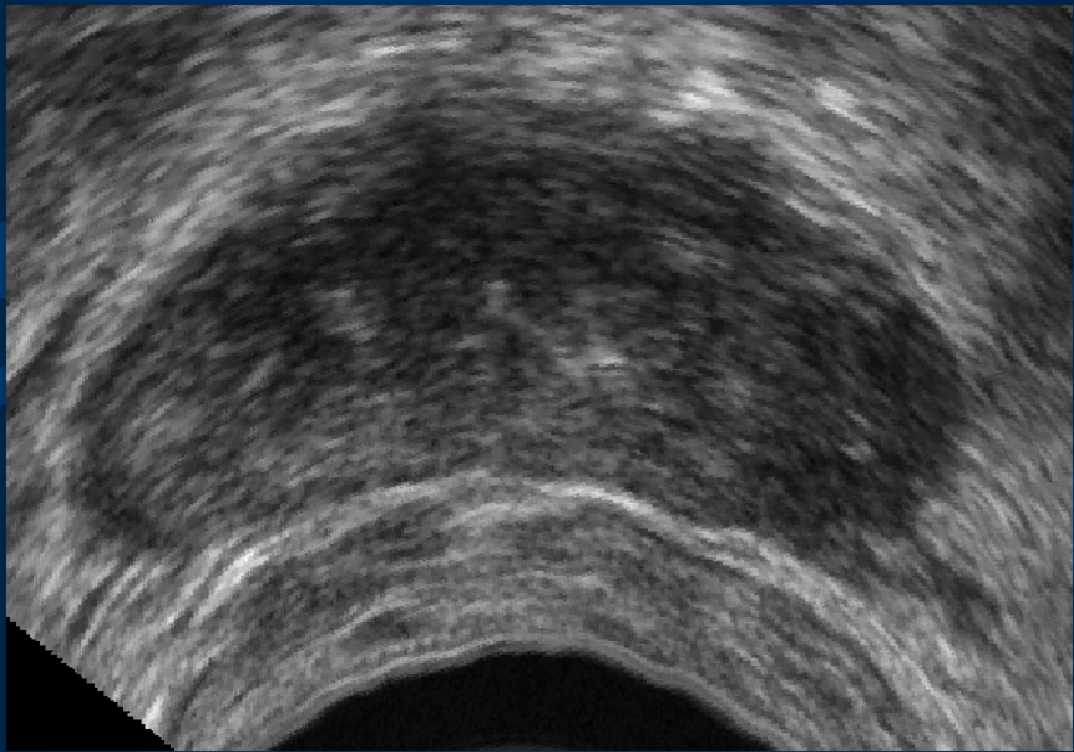
可变形模型

顾 力栩 (***Lixu Gu***)

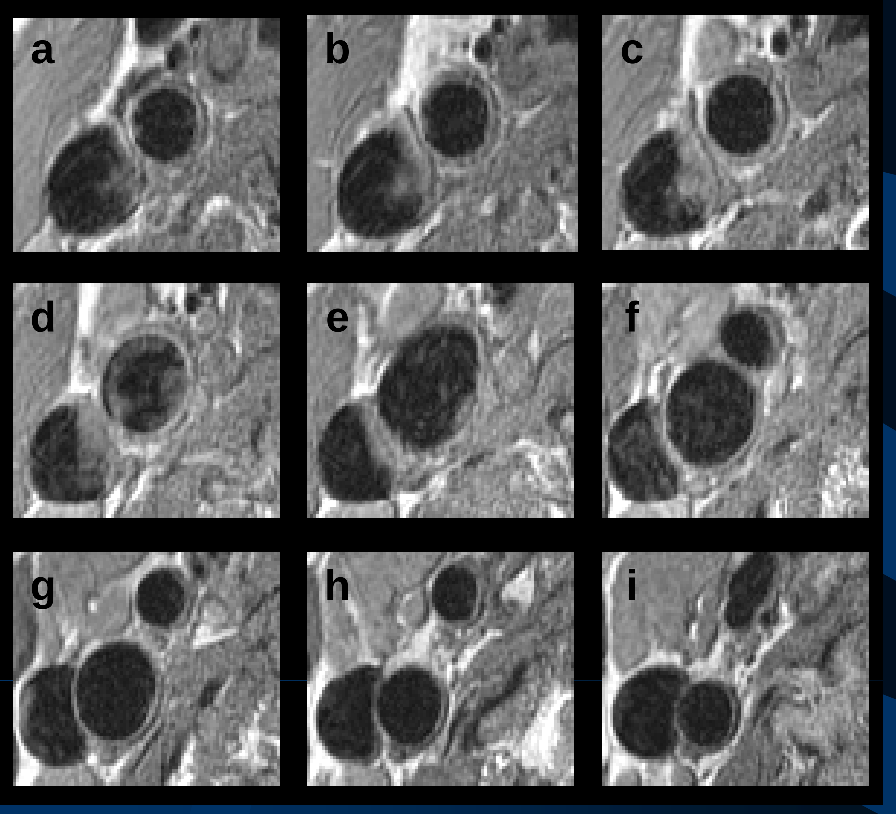
上海交通大学 Med-X研究院

2009.12

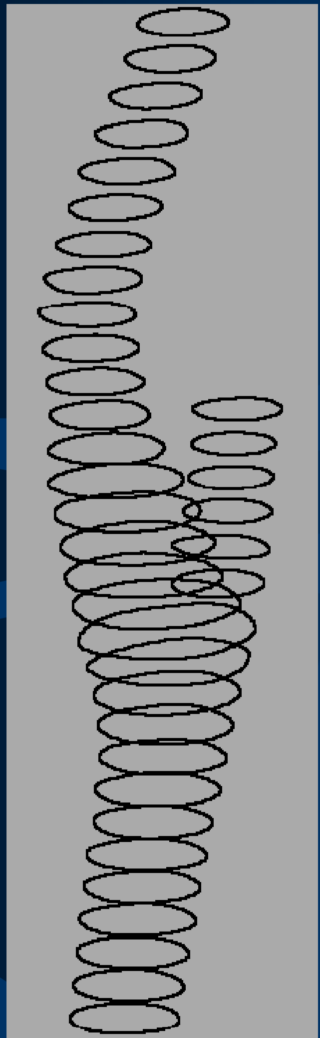
Outlining objects - Prostate example



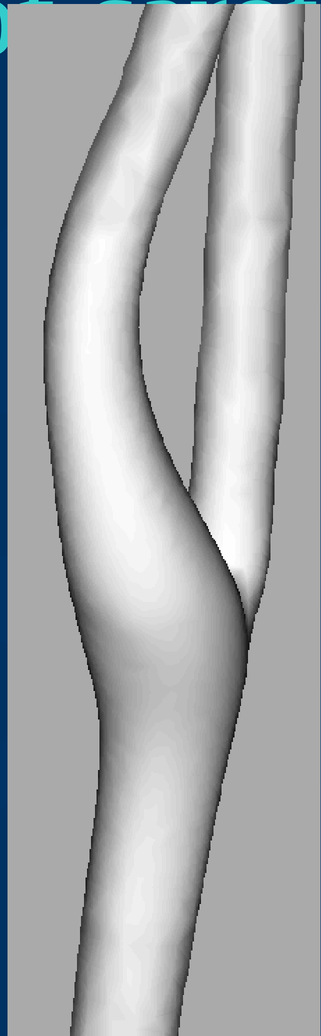
Outlining objects - FE modeling of carotid



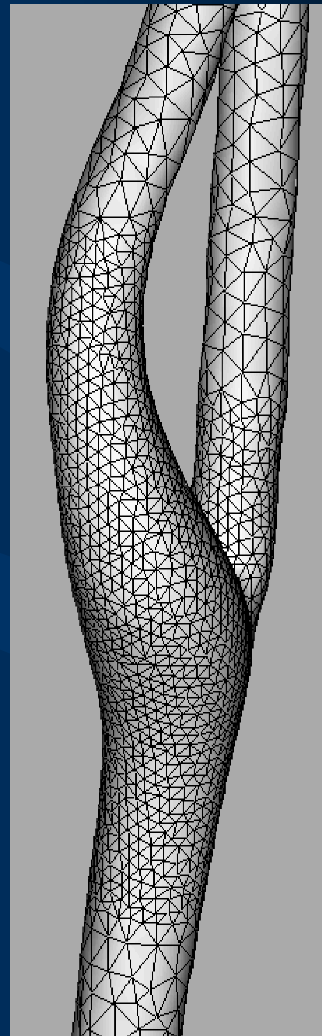
Outlining objects - FE modeling of carotid



segmentation



surface fitting



meshing

Outlining methods

1. Manual
   – user requires expert knowledge & skill in drawing contours
   – tedious & time consuming
   – low degree of reproducibility

2. Fully automatic
   – existing algorithms not sophisticated enough

Outlining methods

3. Automated first guess then manual editing
   - still tedious and time consuming
   - still has a low degree of reproducibility

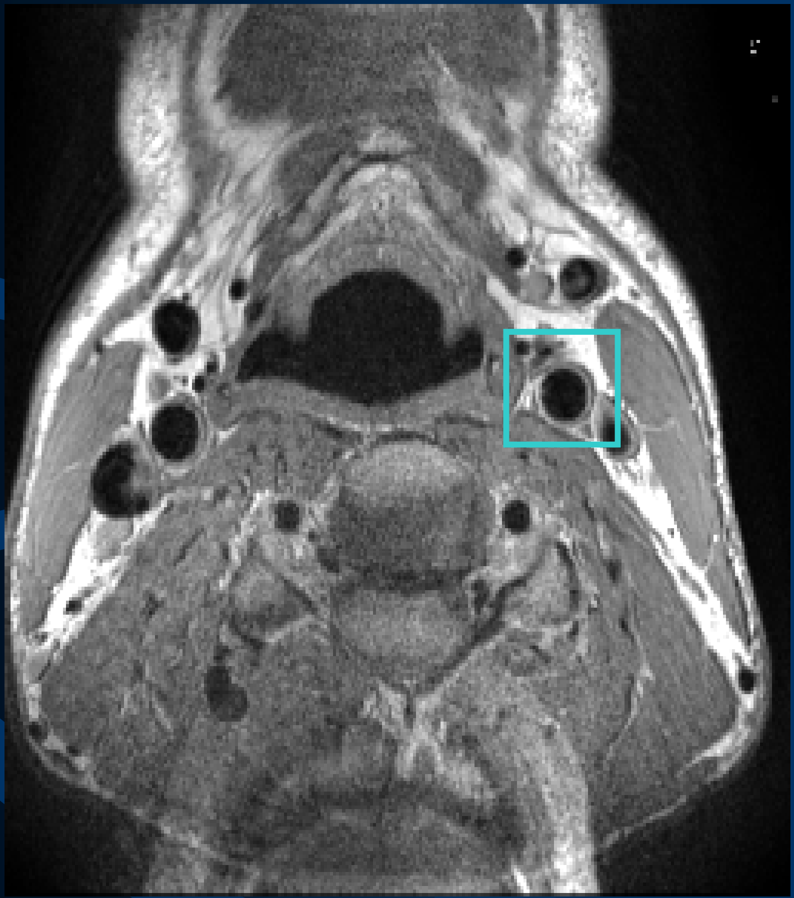
4. Manual rough delineation then automatic refinement
   - quicker than methods (1) and (3)
   - outlines more reproducible than (1) and (3)
   - can be applied to wider range of images than (2)

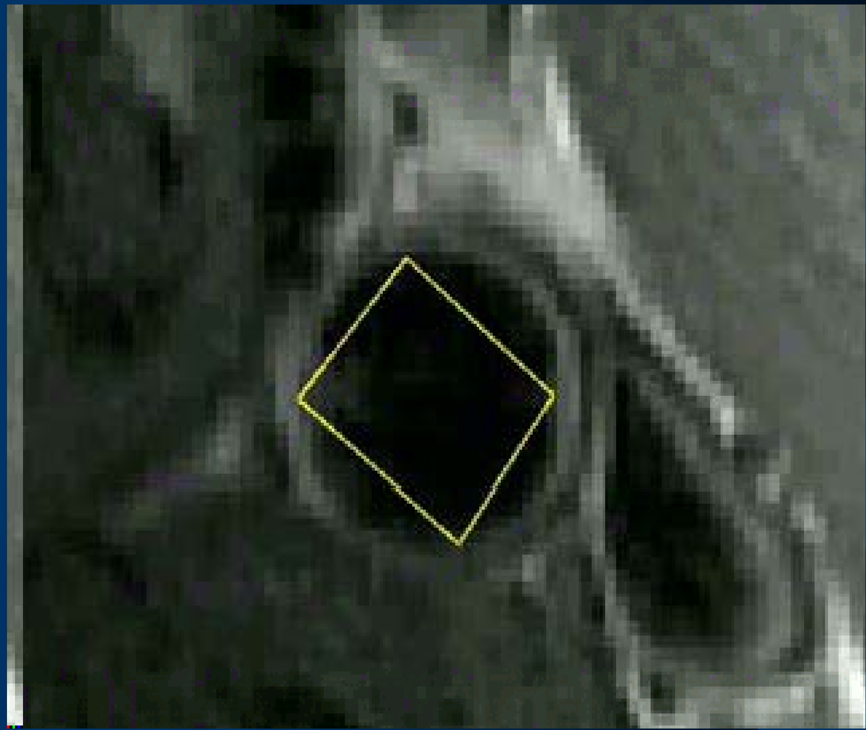
Deformable models

- Contours/surfaces that change shape under the influence of image-based force fields

- Based on method (4):
  1. User draws a rough outline
  2. Outline automatically deformed towards boundary

Deformable models





Details of DDC

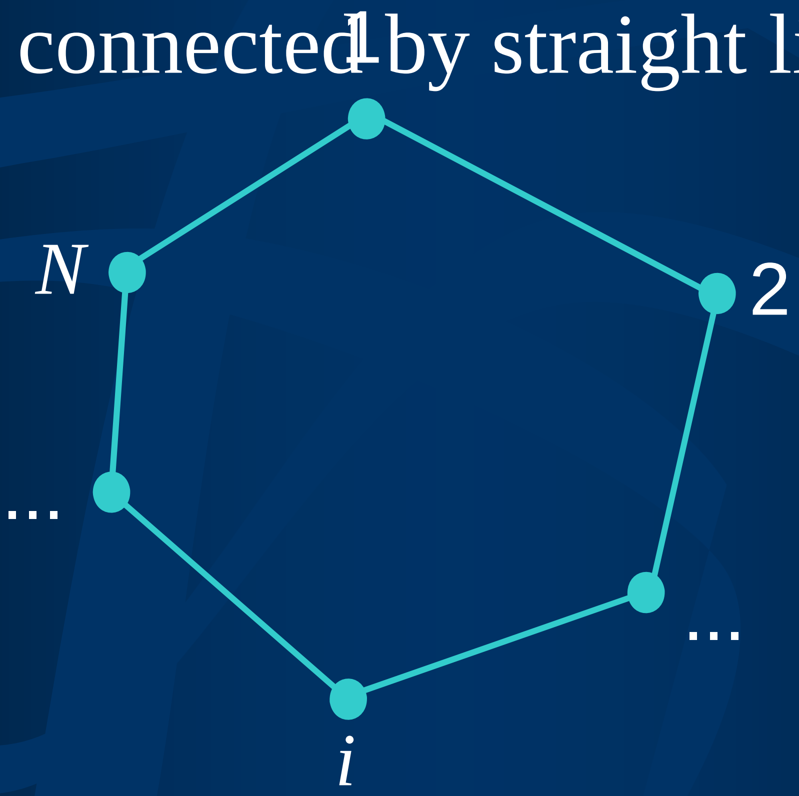
- Will discuss details of one type of deformable: Discrete Dynamic Contour (DDC)

S. Lobregt and M. A. Viergever, “A discrete dynamic contour model,” *IEEE Trans. Med. Imaging*, vol. 14, no. 1, pp. 12 - 24, 1995.

- Only fundamentals covered. Some technical details omitted.

Structure

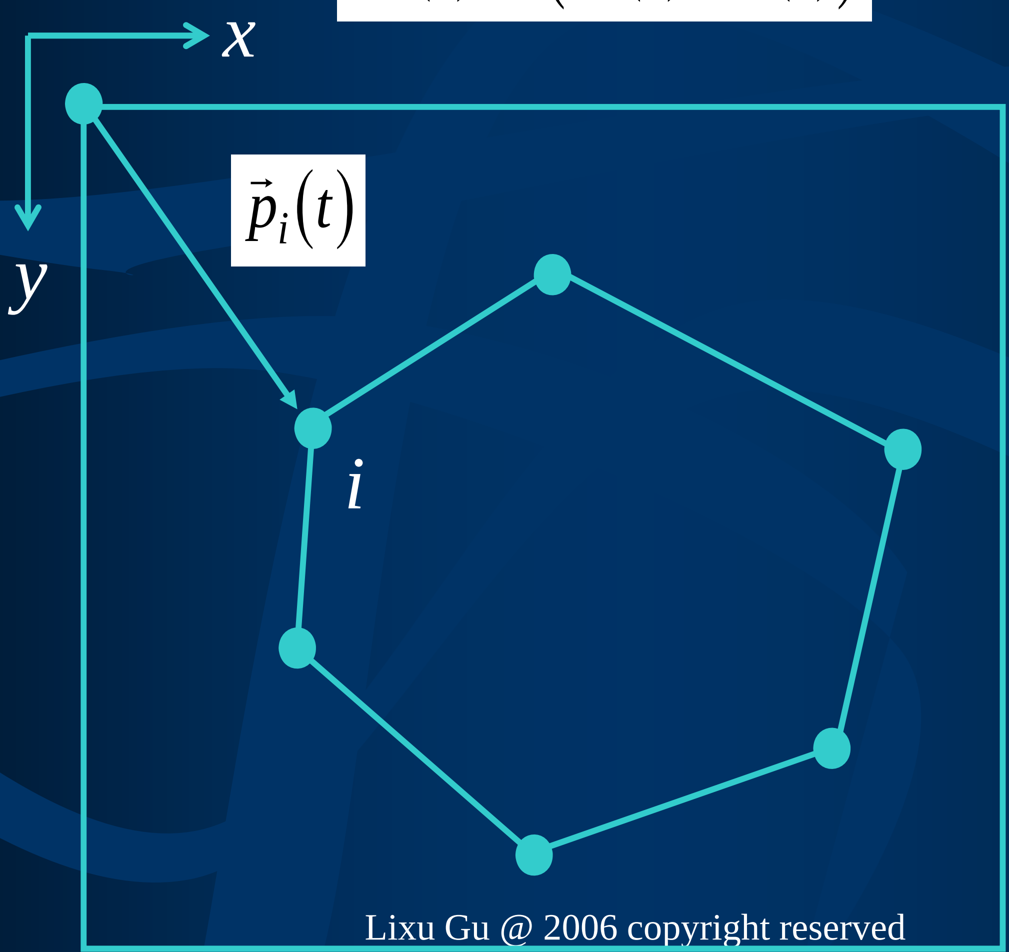
- DDC represented by a set of ordered vertices connected by straight line segments



Coordinates

Each vertex *i* has following coordinates at time t :

$$\vec{p}_i(t) = \left(x_i(t), y_i(t)\right)$$



Forces

Each vertex *i* has a total force acting on it:

$$\vec{f}_i^{\,tot}(t) = w^{im}\,\vec{f}_i^{\,im} + w^{in}\,\vec{f}_i^{\,in} + \vec{f}_i^{\,d}$$

where

$\vec{f}_i^{\,im}(t)$ is an image force that drives each vertex towards "features" that define boundary.

$\vec{f}_i^{\,in}(t)$ is an internal force that keeps the contour "smooth" in the presence of noise in the image.

$\vec{f}_i^{\,d}(t)$ is a damping force that makes the dynamical behaviour of the contour stable (next section).

Dynamics

Total force acting on each vertex *i* at time *t* causes it to accelerate:

$$\vec{a}_i(t) = \frac{1}{m}\vec{f}_i^{\,tot}(t)$$

where *m* is the mass of each vertex. Usually, $m = 1$.

Dynamics

Acceleration of vertex i causes a change in its velocity and position, both of which can be updated from time t to time $t + \Delta t$ by numerical integration:

$$\vec{v}_i(t+\Delta t) = \vec{v}_i(t) + \vec{a}_i(t)\Delta t$$

$$\vec{p}_i(t+\Delta t) = \vec{p}_i(t) + \vec{v}_i(t)\Delta t$$

Overall algorithm

1. Display image
2. Allow user to initialize contour.
   Set velocity & acceleration of each vertex to 0.
3. Calculate total force at each vertex.
4. Calculate acceleration of each vertex.
5. Update position and velocity of each vertex.
6. “Resample” DDC.
7. Repeat (3) to (6) until all vertices stop moving:

$$\left\|\vec{v}_i(t)\right\| < \varepsilon_1, \quad \left\|\vec{a}_i(t)\right\| < \varepsilon_2$$

Image forces

$\vec{f}_i^{\,im}(t)$ is an image force that drives each vertex towards “features” that define boundary.

Derived from an “energy” that is defined at all pixels of the image:

$$\vec{f}_i^{\,im} = -\vec{\nabla} E(x_i, y_i)$$

Image forces drive each vertex to *closest local* minima of energy field.

Image forces

Success of algorithm depends on defining energy that propels each vertex to the desired image feature.

For example, to drive a vertex to areas of maximum gradient magnitude, we can define the energy as:

$$E(x_i, y_i) = \frac{1}{\left\| \vec{\nabla}(G_\sigma * I) \right\| + \varepsilon}$$

Image forces

Image force to localize maximum gradient magnitude

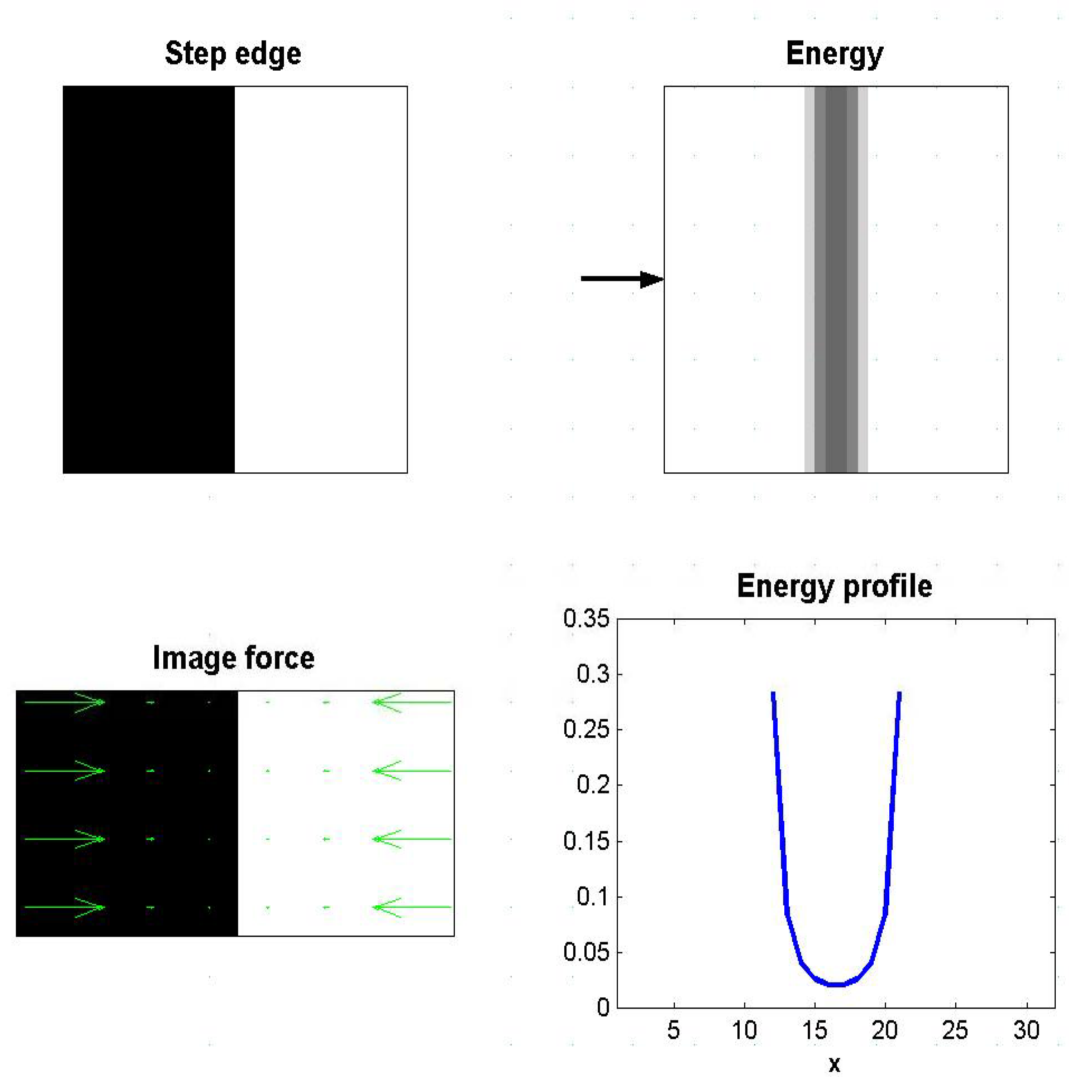
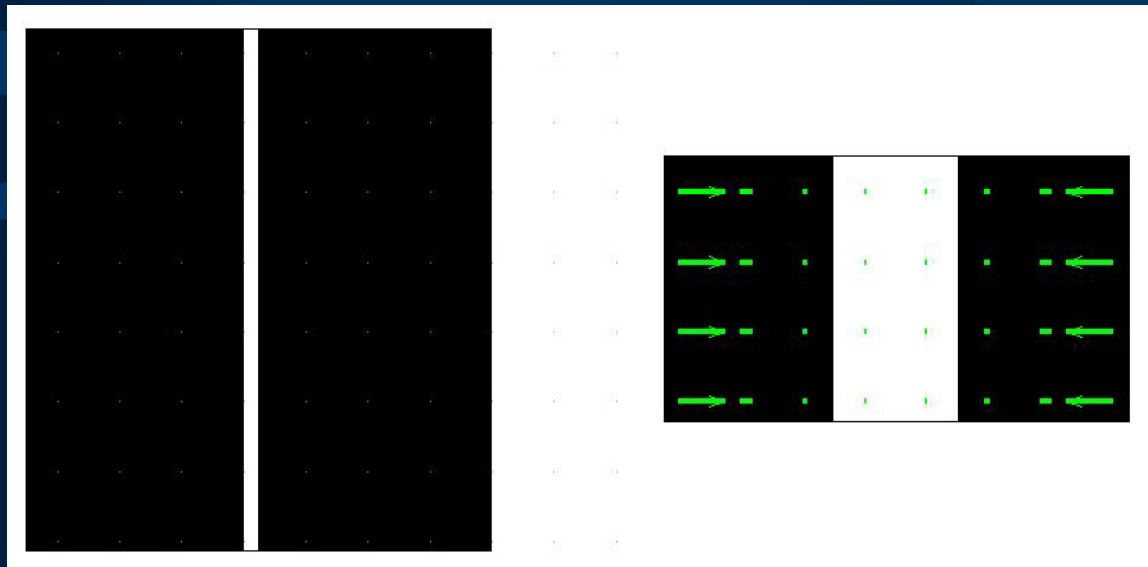


Image forces

The way you choose to define the energy field depends on what features (step edges, line elements, etc) you are trying to localize.



$$E(x_i, y_i) = \frac{1}{G_\sigma * I + \varepsilon}$$

For line elements.

Internal forces

$\vec{f}_i^{\,in}(t)$ is an internal force that keeps the contour “smooth” in the presence of noise in the image.

Noise in the image can cause DDC to become jagged. Internal force keeps DDC smooth by minimizing local curvature. We can take it to be proportional to local curvature:

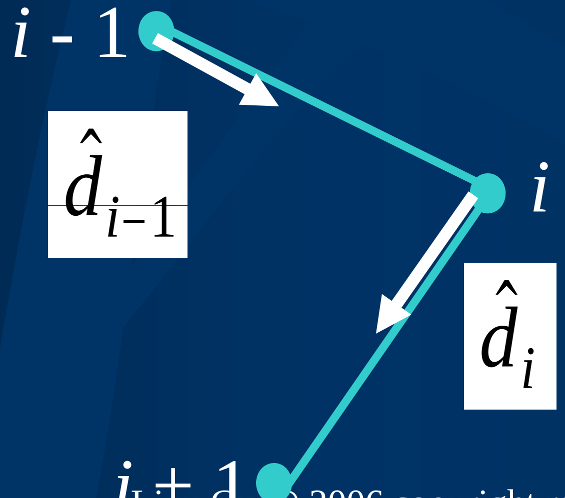
$$\vec{f}_i^{\,in}(t) = \vec{c}_i(t)$$

Internal forces: Curvature

Local curvature defined as:

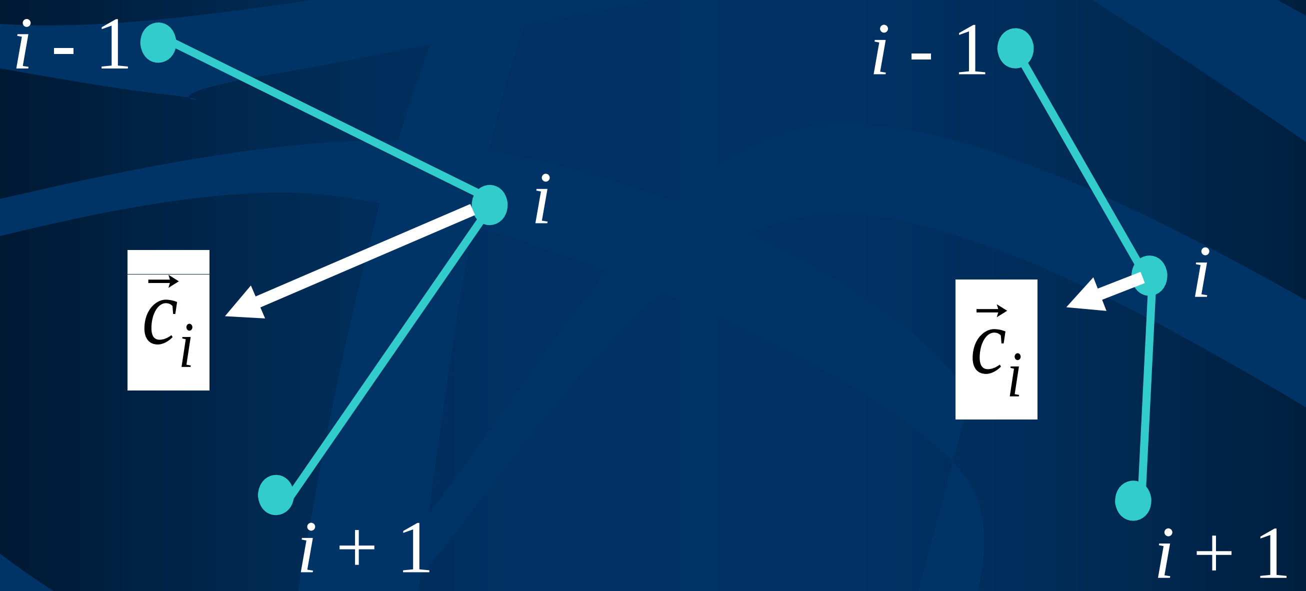
$$\vec{c}_i(t) = \hat{d}_i(t) - \hat{d}_{i-1}(t)$$

$\hat{d}_i(t)$ **is unit edge vector connecting vertex *i* to *i* + 1.**



Internal forces: Curvature

Local curvature is proportional to the angle between the two edges connected to the vertex.



Large curvature **Small curvature**

Damping force

$\vec{f}_i^{\,d}(t)$ is a damping force that makes the dynamical behaviour of the contour stable (next section).

With image and internal forces only, DDC may oscillate between two local minima. Viscous damping is necessary for convergence.

$$\vec{f}_i^{\,d}(t) = w^d \vec{v}_i(t)$$

Require -1 < w^d < 0.

Resampling

As the DDC deforms, the spacing between vertices will change:

• If it becomes too large, the DDC will not follow curved boundaries well.

• If it becomes too small, the DDC will not be efficient in terms of memory and speed.

Resampling adds and deletes vertices to ensure that curved boundaries are modeled accurately and representation is “compact”.

Resampling

Linearly interpolate new vertices so that they are evenly spaced by distance Δ along length of DDC.

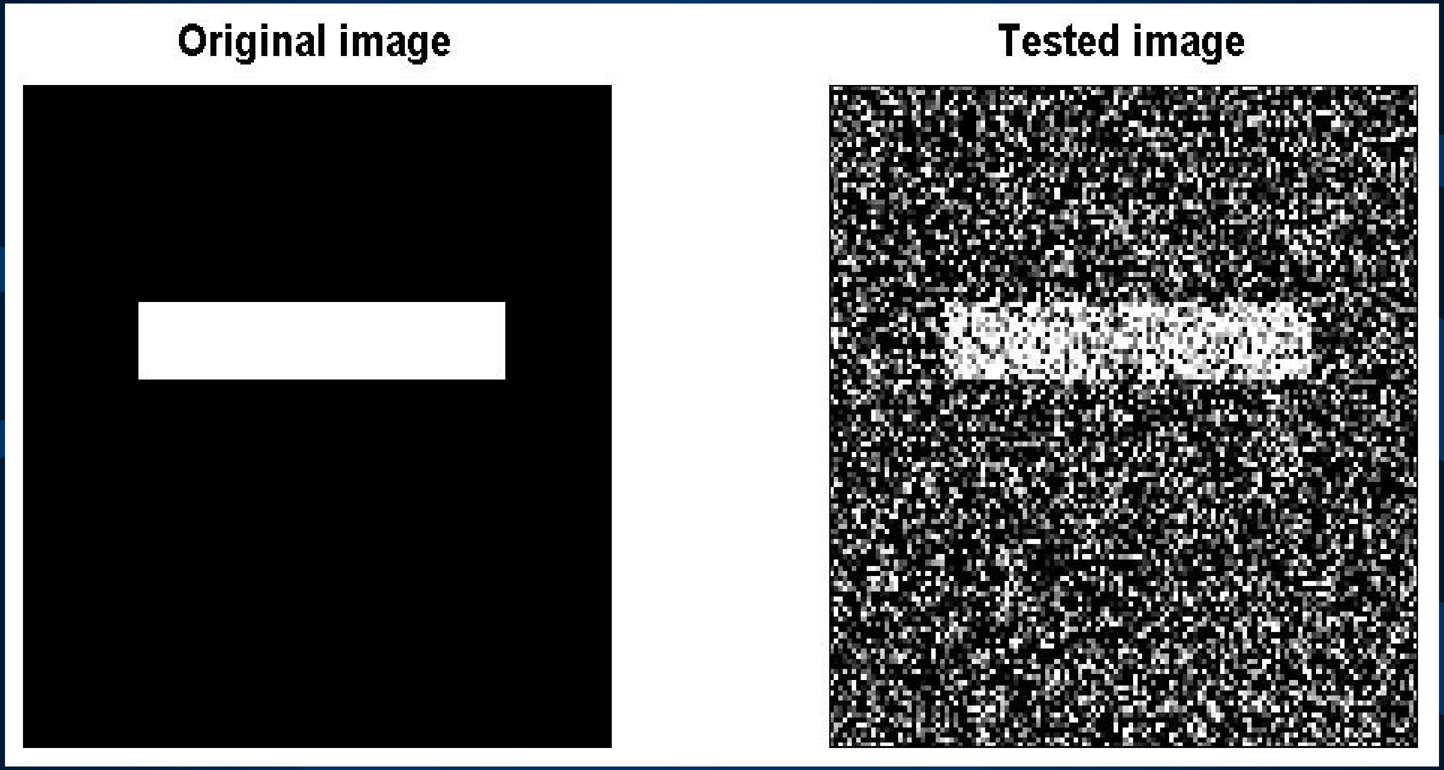
Before resampling **After**

Technical details

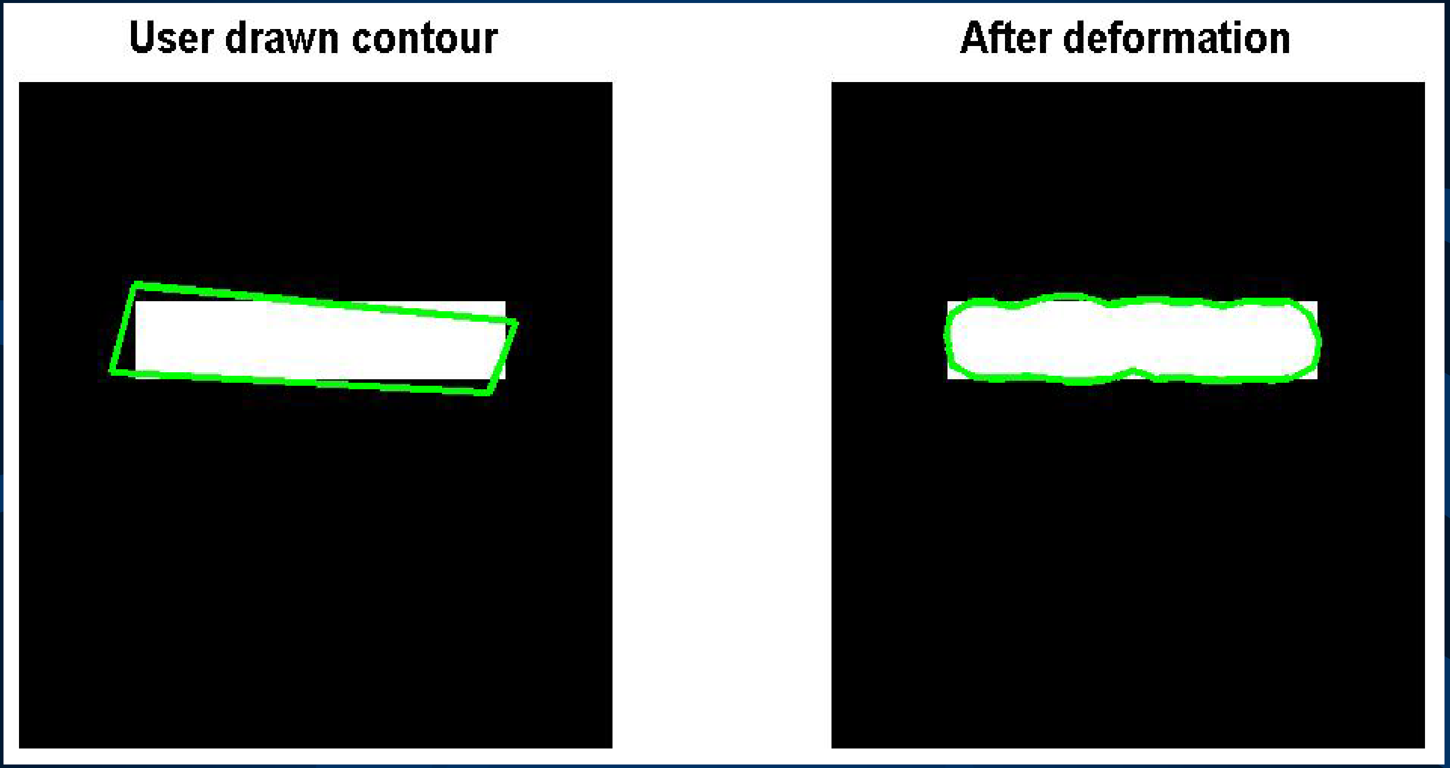
Following details not considered here:

- Clustering of vertices caused by tangential component of image force
- Implosion of polygons caused by internal forces
- Open contours
- Editing (suggestions in paper as well as Ladak *et al*)
- Selection of weights

Results - Example

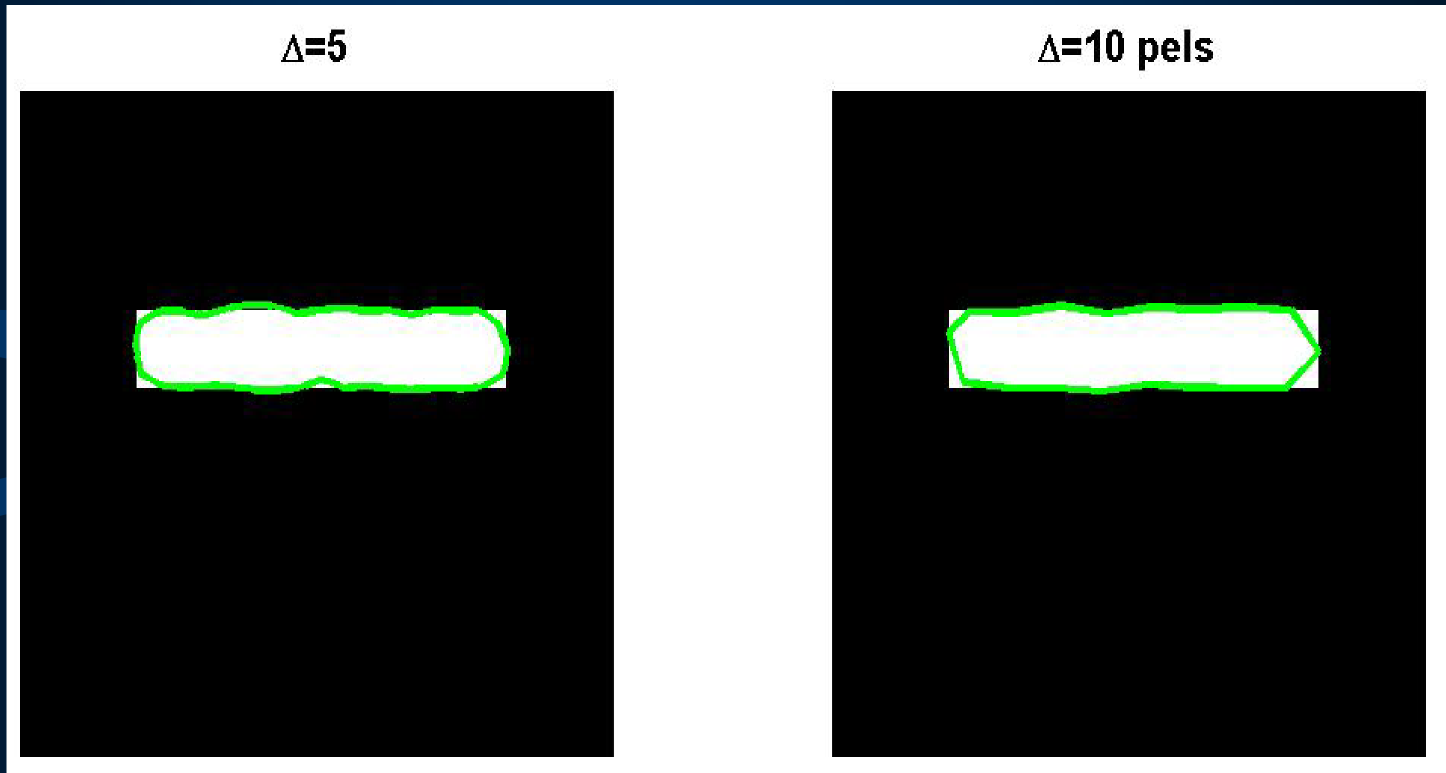


Results - Example



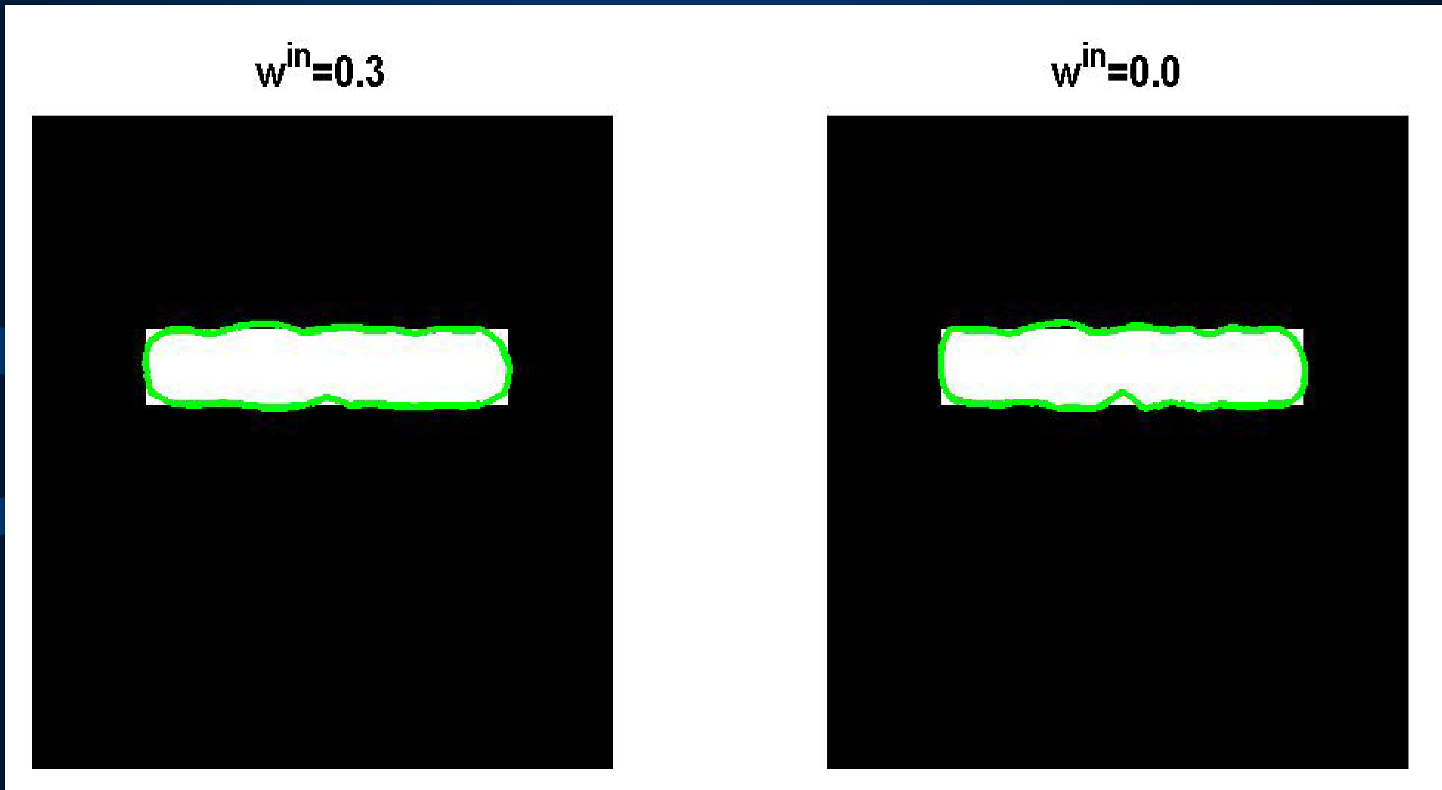
$$w^{in} = 0.3,\ w^{im} = 1.0,\ \sigma = 2,\ \Delta = 5$$

Results - Effect of Δ



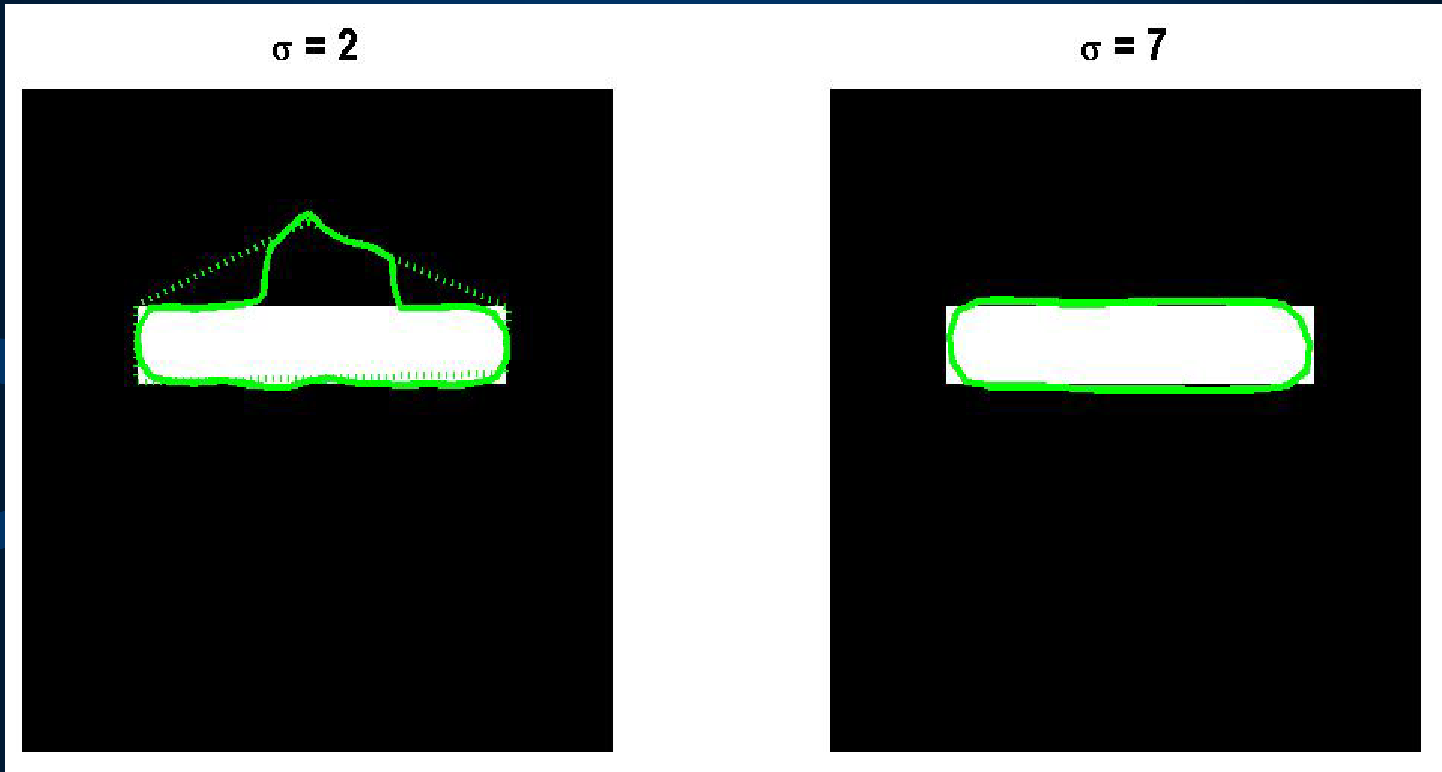
$$w^{in} = 0.3,\ w^{im} = 1.0,\ \sigma = 2$$

Results - Effect of w^{in}



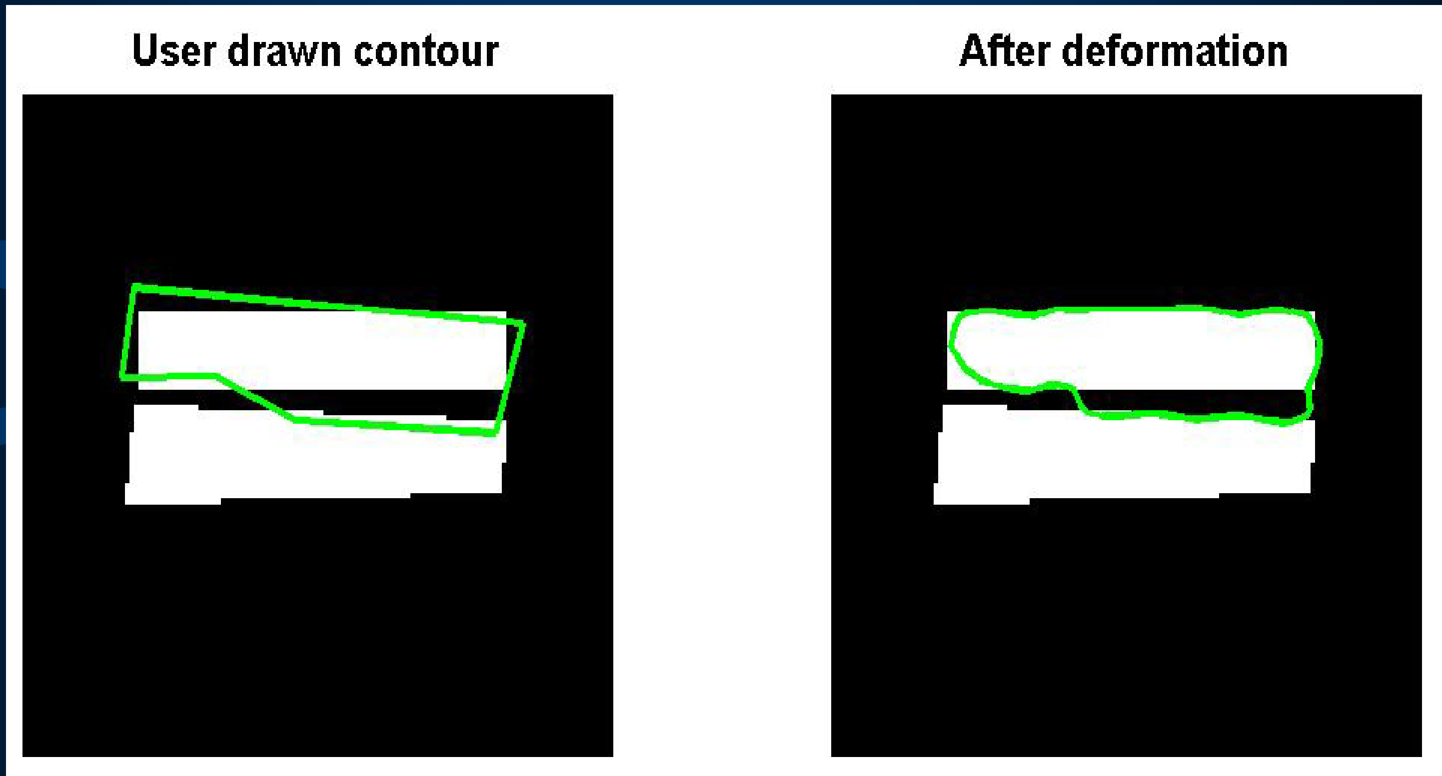
$$w^{im} = 1.0,\ \sigma = 2,\ \Delta = 5$$

Results - Effect of σ

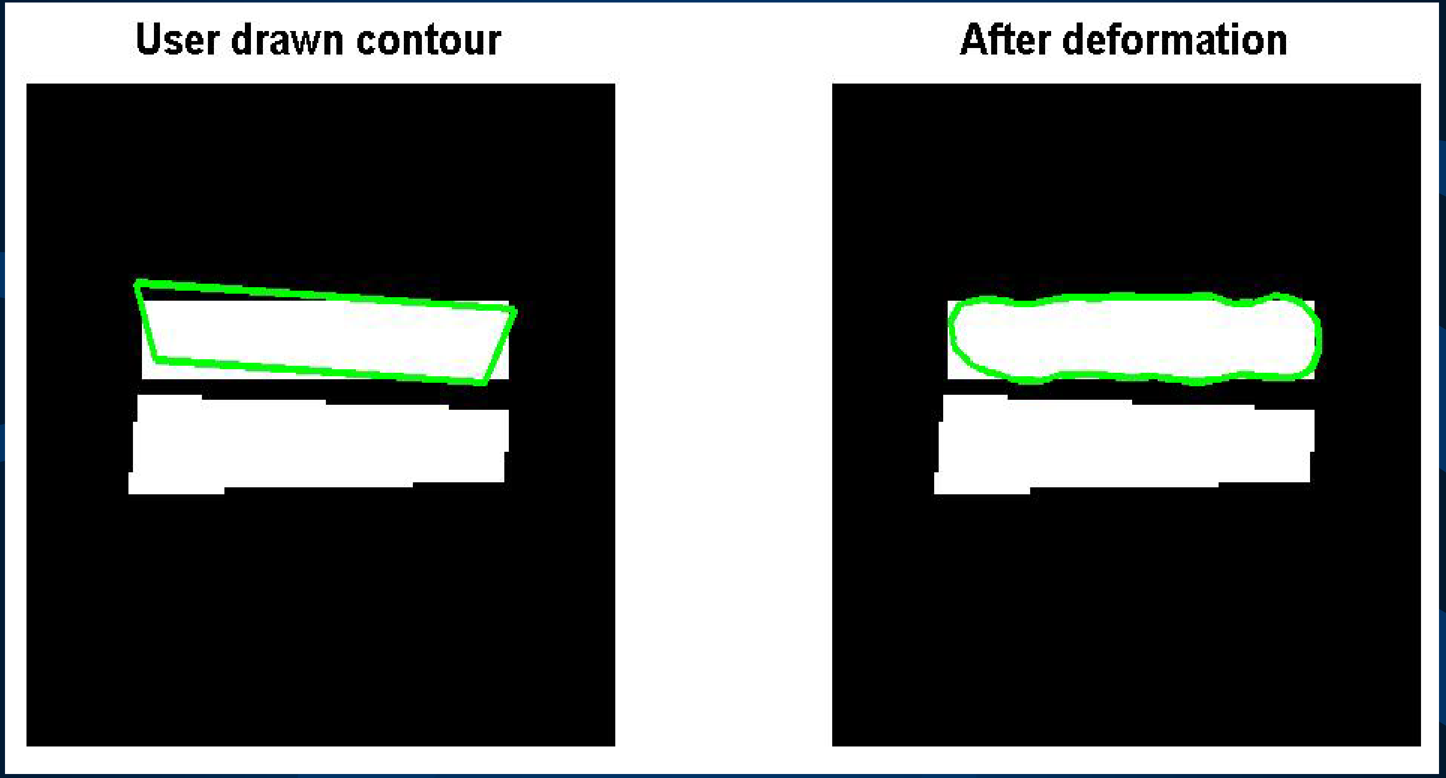


$$w^{in} = 0.3,\ w^{im} = 1.0,\ \Delta = 5$$

Results - Effect of initial contour



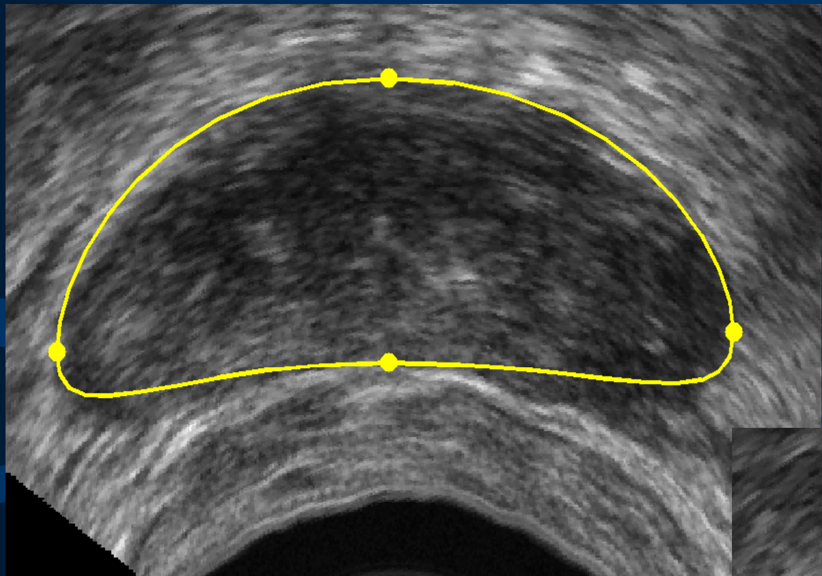
Results - Effect of initial contour

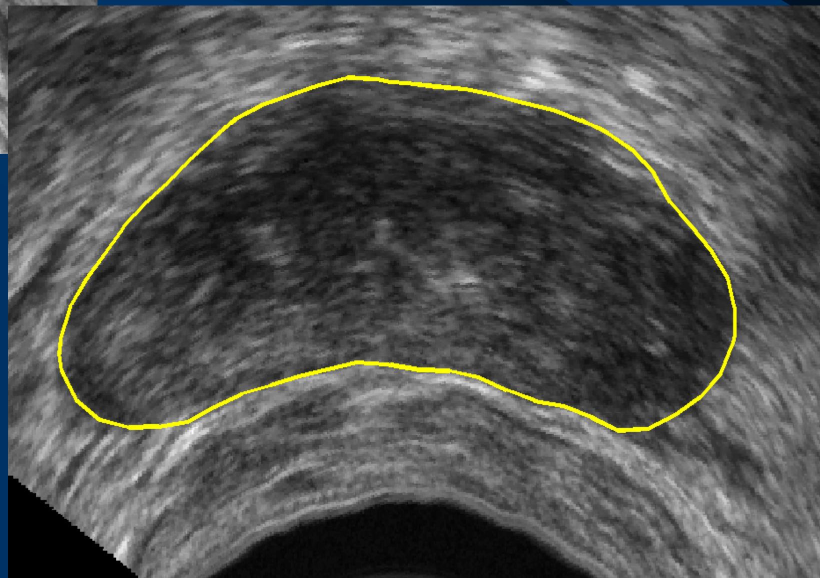


Summary

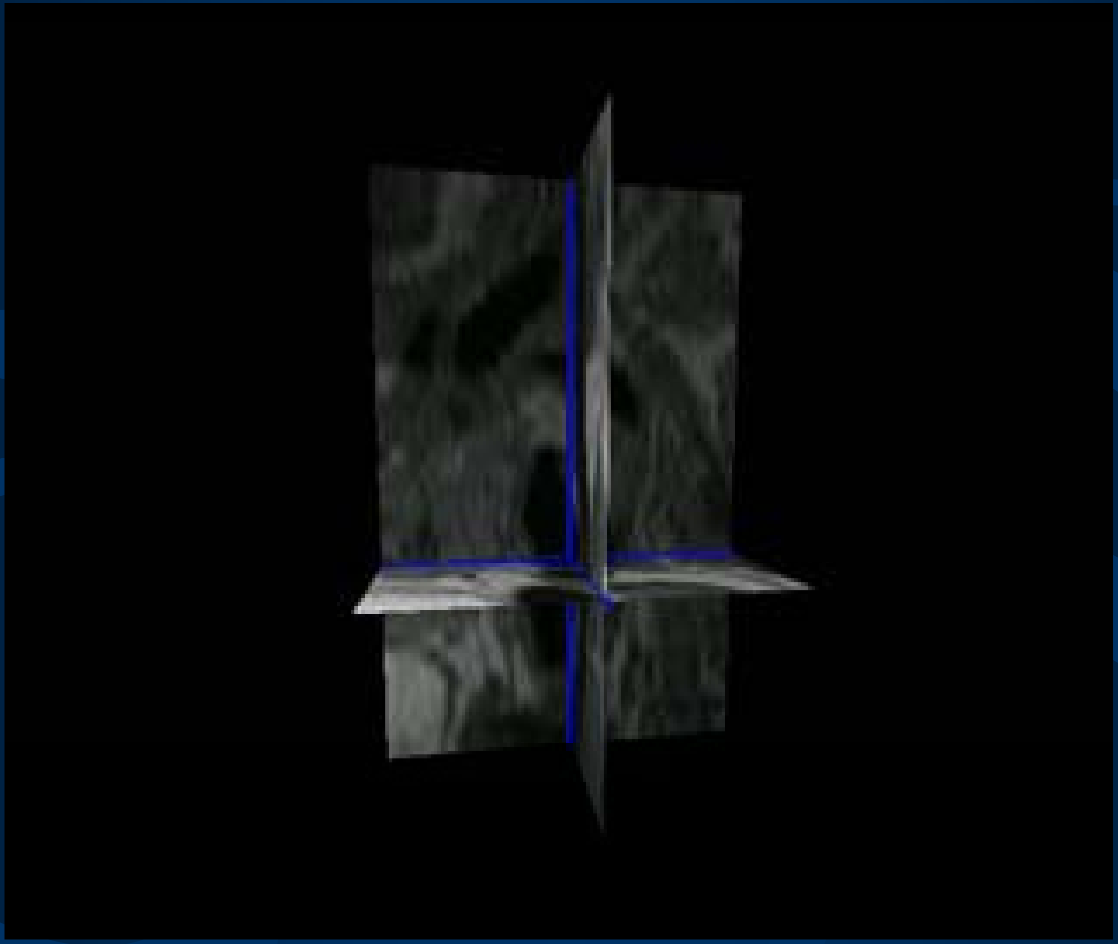
- Deformable models like the DDC automatically change shape to conform to object boundaries
- Initial user-drawn DDC must be relatively close to desired boundary in order to be attracted to it
  - can tolerate small differences in initialization (more reproducible than manual method)
- Choice of parameters can affect final outcome
  - σ should be small to localize boundary accurately
  - $w^{im} > w^{in}$ if images not very noisy
  - Δ determines how well DDC follows curves

Application: Outlining prostate





Extension to 3D



Research trends (1987 - present)

- Various optimization methods for variational formulation of problem
- Automatic initialization
- Boundary representation
- Incorporation of organ shape distributions to constrain deformation to make model more robust
- Incorporation of organ appearance (gray levels and gradients) distributions
- Assessment of accuracy & reproducibility

Discussion