Geometric Active Contours & Level Set Methods

EE 528, ISU

- Geometric contour representation (or any geometric quantity)
  - represent as a function of arclength or derivatives w.r.t. it

- Geometric energy: function of geometric quantities

- Arclength paramaterization
  - ds = $||C_p||$ dp
- Curvature
- Energy, E = total contour length
  - Minimizing flow: kappa N
- Energy E = $\int F(C,C_s)\, ds = \int F(C,C_p)\, ||Cp||dp$
  - Minimizing Flow: in class

Level set Representation

- Represent contour as zero level set of a two dim function u(x,y)
- Parametric v/s level set rep of a circle
- Generalize to any contour: use Signed Distance Function (SDF)
- Finding zero level set (contour)

Why level set method?

- Accuracy/stability when contour length (or local length in diff regions) changes
  - Marker points bunch up or move too far

- Deal with singularities, e.g. corners

- Topology changes automatically handled

- Contour evolution ←→ Level set evolution
- Extension velocity: extend along normal direction to zero level set
- PDE implementation: upwind method for advection term, central differences for curvature term

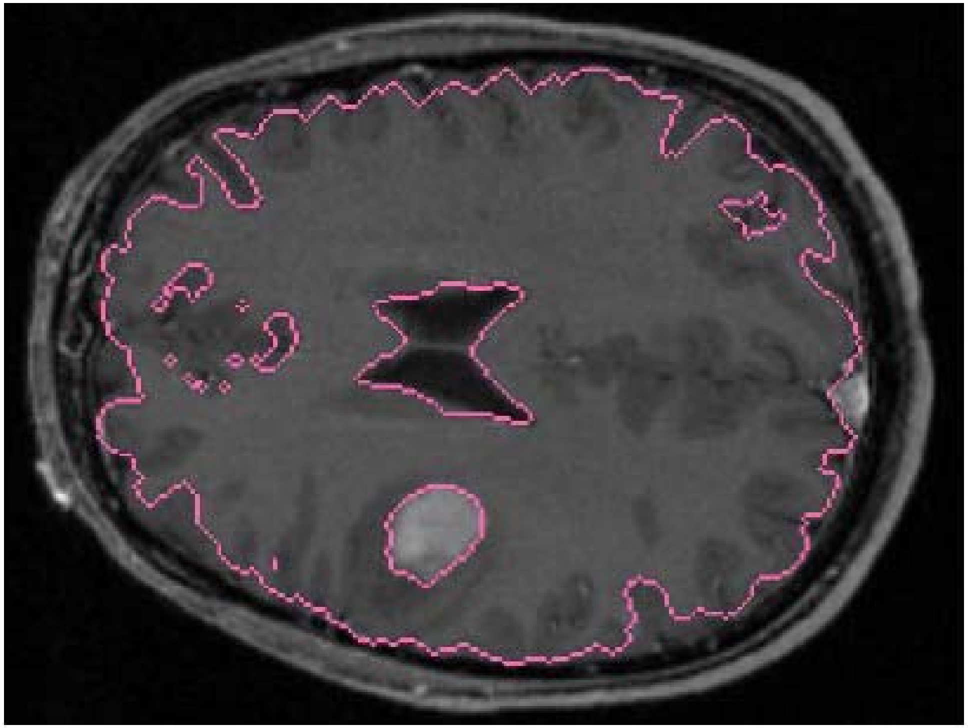
- Re-initialization to get back an SDF
  - Needed because level sets bunch up due to numerical errors (even when using normal extension velocities)

- Faster implementation: narrow-band method or local method

Edge & Region based GAC

- Edge based: Casselles et al, Yezzi et al
  - “geodesic active contours”
  - E = $\int$ g(grad I) ds
- Region based: Chan-Vese
  - E = variance inside contour + variance outside contour + total contour length
- Region based: Yezzi et al
  - E = squared diff b/w average intensities inside and outside contour + total contour length

Edge based GAC: Advantage



- Fast computation
- Simultaneously segment many diff regions with diff intensities (or other features)

Region based GAC

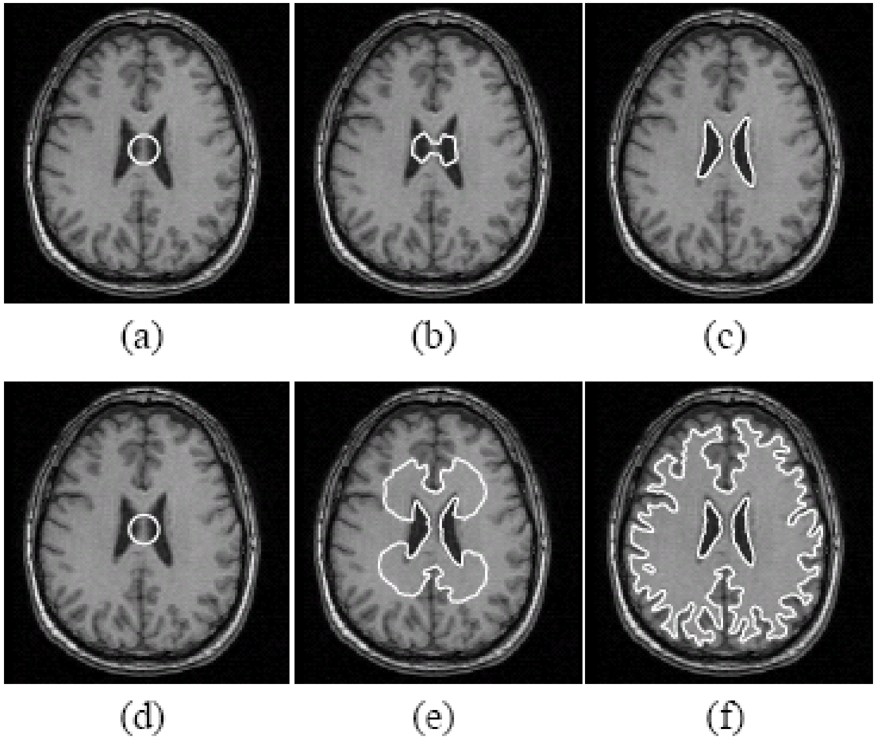


Fig. 3: A region-based geometric active contour can extract either the ventricle (top row) or the white matter (bottom row).

- Cannot segment multiple types of regions simultaneously

Disadvantages of Edge based

- Gap penetration (penetrates regions where there is gap in large gradient mag.)
- Gets stuck at points of local max of gradient magnitude: may not be the desired boundary
- Sensitive to noise: gradient computation is sensitive, here also taking grad of grad
- Much more sensitive to initialization

Gap penetration: Fix by increasing curvature weight (also smoothes more)

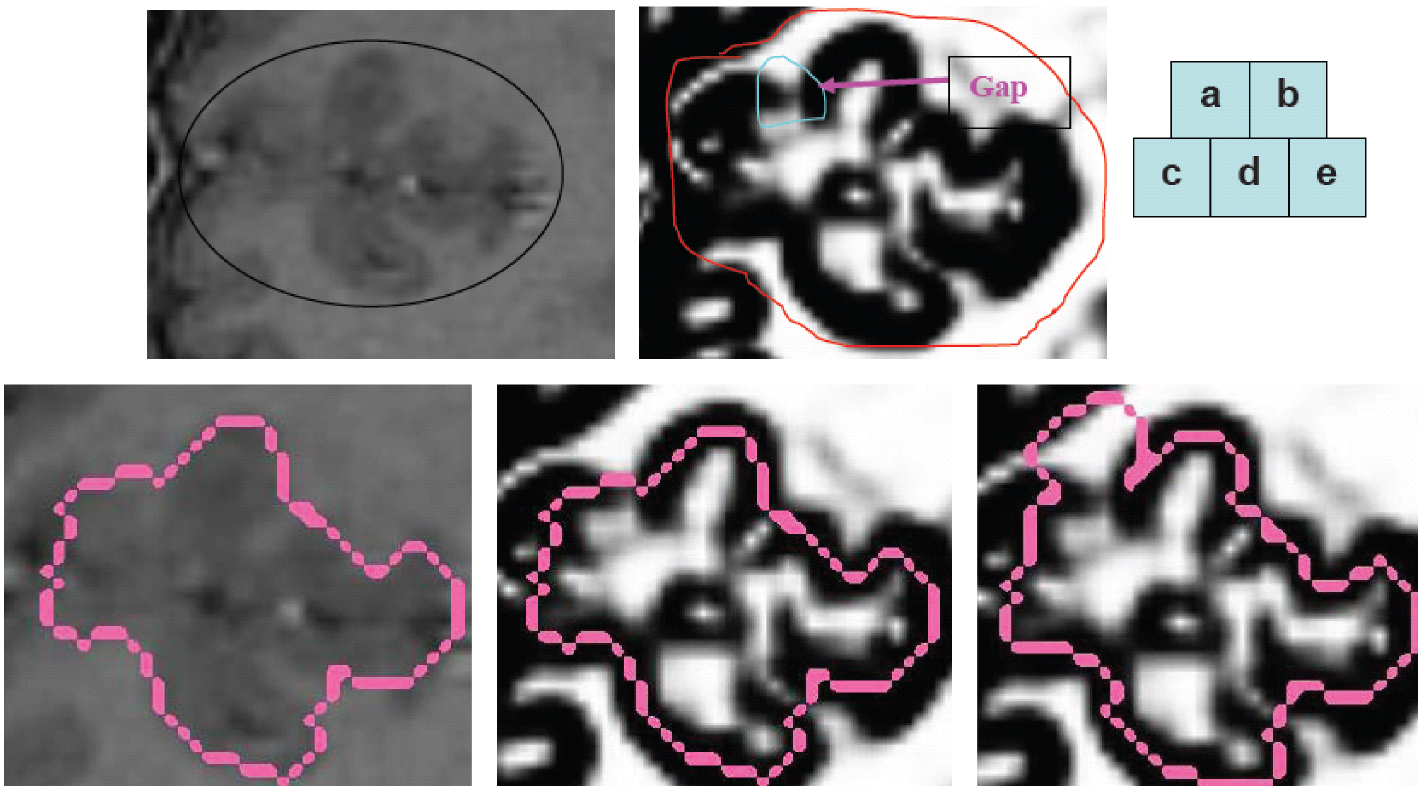


Figure 6, a: original image with highlighted feature; b: feature image after three stages of filtering showing a gap; c: overlay of zero level set function after 450 iteration with propagation weight 2, advection weight 3 and **curvature weight 2** over original image; d: the same result as c, but overlay on top of the feature image; e: overlay of zero level set function after 450 iteration with propagation weight 2, advection weight 3 and **curvature weight 0** over feature image

Sensitivity to local grad max: fix by increasing advection weight

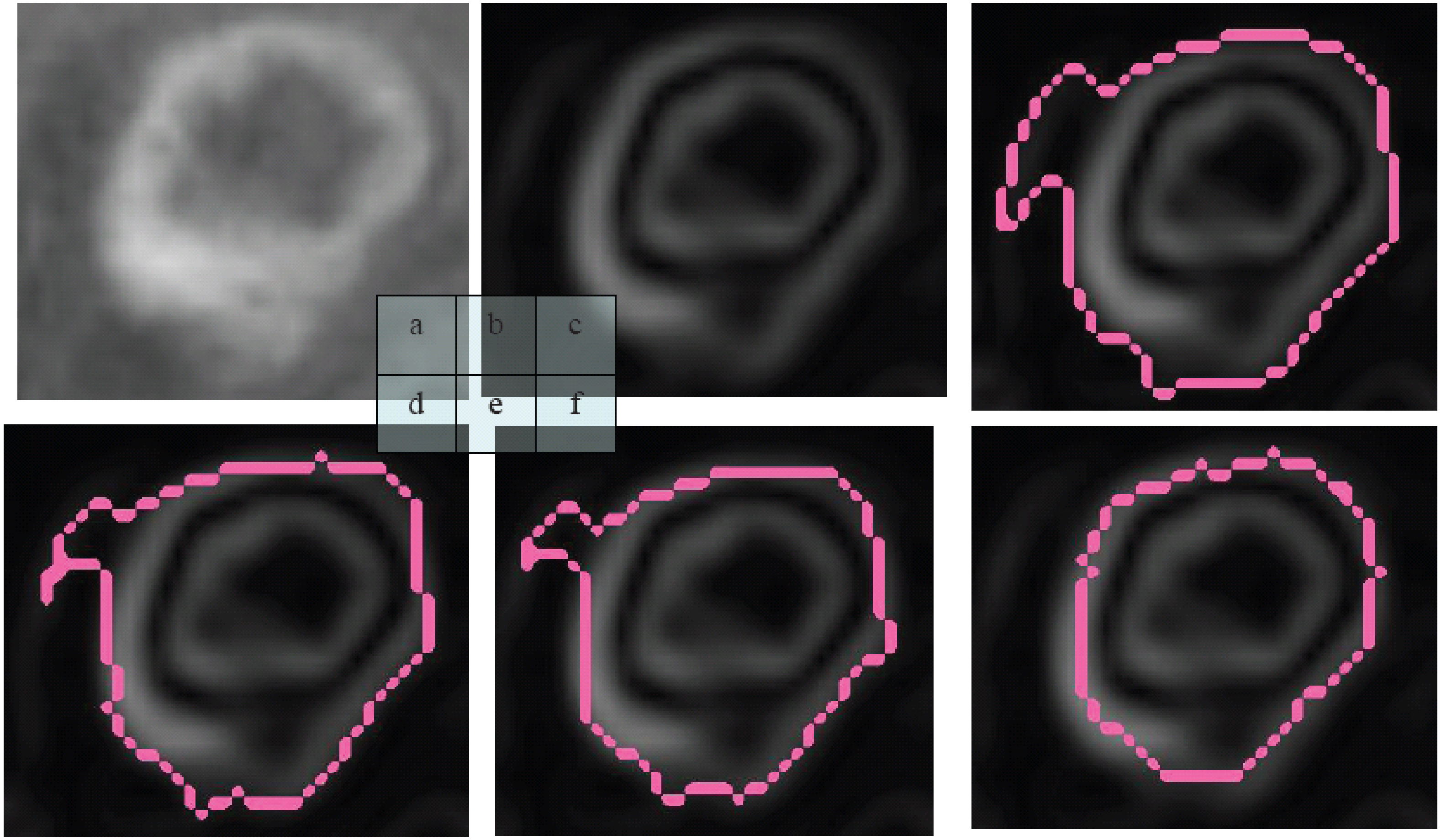


Figure 7. a: original image with a tumor; b: gradient magnitude image; c-f: segmentation results after 700 iterations with propagation weight -2, curvature weight 1 (advection weight for c:0, d:1, e:2, f:3)

Region based GAC

- Robust to noise
- Supports difference in various image properties (not just edges), e.g. texture
  - Works well with blurred images and also not sensitive to local grad max
- Less sensitive to initialization
- Disadvantage
  - Hard to implement on multiple “groups”
  - Expensive: re-compute contour evolution force every time

Region: Not sensitive to initialization

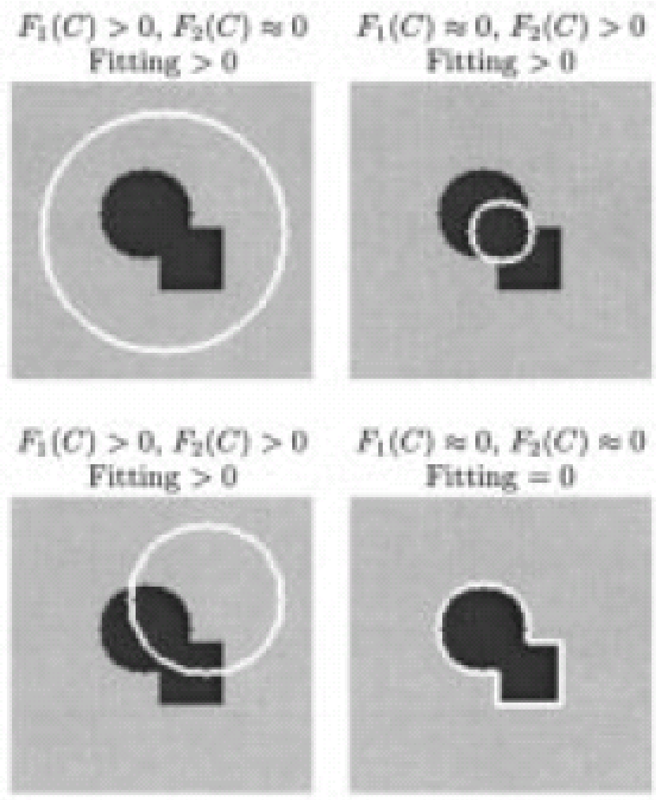


Fig. 1. Consider all possible cases in the position of the curve. The fitting term is minimized only in the case when the curve is on the boundary of the object.

Region: Less sensitive to noise

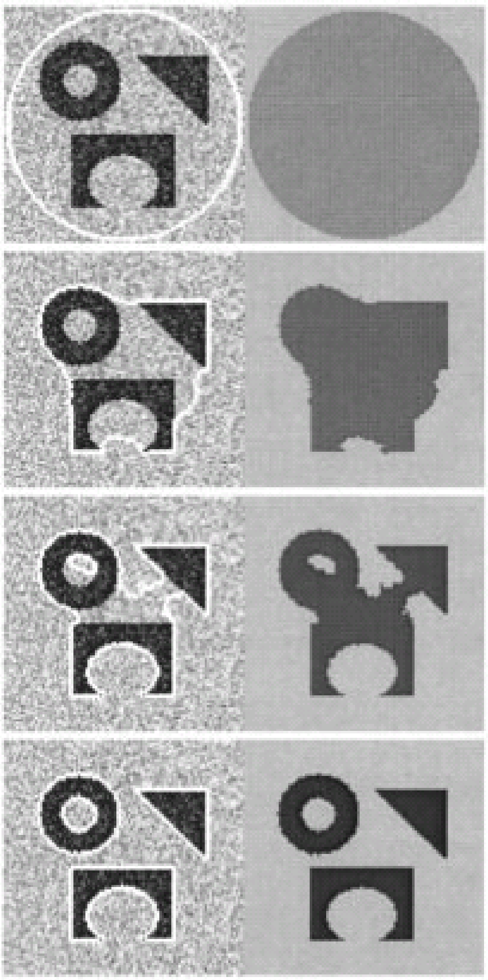


Fig. 4. Detection of different objects from a noisy image, with various shapes and with an interior contour. Left: u_0 and the contour. Right: the piecewise-constant approximation of u_0. Size $= 100 \times 100$, $\phi_0(x, y) = -\sqrt{(x-50.5)^2 + (y-50.5)^2} + 48.5$, $\mu = 0.1 \cdot 255^2$, no reinitialization, cpu $= 4.60$ s.