Lecture 10
Segmentation, Part II (ch 8)
Active Contours (Snakes)
ch. 8 of *Machine Vision* by Wesley E. Snyder & Hairong Qi

Spring 2020
16-725 (CMU RI) : BioE 2630 (Pitt)

Dr. John Galeotti

Review: Segmentation

- A partitioning...
  - Into connected regions...
- Where each region is:
  - Homogeneous
  - Identified by a unique label

There is room for interpretation here...

And here
Review: The “big picture:” Examples from *The ITK Software Guide*

![Image of figures from ITK Software Guide](image)

Review: The Nature of Curves

- A curve is a 1D function, which is simply bent in (“lives in”) ND space.

- That is, a curve can be parameterized using a single parameter (hence, 1D).

- The parameter is usually arc length, $s$
  - Even though not invariant to affine transforms.
Review: The Nature of Curves

- The speed of a curve at a point \( s \) is:
  \[
  \Psi(s) = \sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2}
  \]
- Denote the outward normal direction at point \( s \) as \( n_{\Psi}(s) \)
- Suppose the curve is closed:
  - The concepts of INSIDE and OUTSIDE make sense
  - Given a point \( x = [x_i, y_i] \) not on the curve,
  - Let \( \Psi_x \) represent the closest point on the curve to \( x \)
    - The arc length at \( \Psi_x \) is defined to be \( s_x \).
  - \( x \) is INSIDE the curve if:
    \[
    [x - \Psi_x] \cdot n_{\Psi}(s_x) \leq 0
    \]
  - And OUTSIDE otherwise.

Active Contours (Snakes)

- Most whole-image segmentation methods:
  - Connectivity and homogeneity are based only on image data.
- In medical imaging, we often want to segment an anatomic object:
  - Connectivity and homogeneity are defined in terms of anatomy, not pixels
  - How can we do this from an image?
  - We definitely have to make use of prior knowledge of anatomy!
    - Radiologists do this all the time
Let’s look at ultrasound of my neck.

Examine the CCA:
- Large parts of the boundary are NOT visible!
- We know the CCA doesn’t include the large black area at the bottom-right

How can we automatically get a “good” segmentation?
- This is (usually) hard.

How do we know where the edges/boundaries are?

Why are they missing in some places?
- Ultrasound & OCT frequently measure pixels as too dark
- Nuclear medicine often measures pixels as too bright
- X-Ray superimposes different objects from different depths

What can we do about it?
- Edge closing won’t work.
- “Hallucinate”
Active Contours (Snakes)

- Another example & underlying idea:

- Active contours can insure* that:
  - The segmentation is not "drastically" too large or too small
  - It is approximately the right shape
  - There is a single, closed boundary
  - Active contours can still be very wrong
    - Just like every other segmentation method

* Requires careful usage.
Active Contours (Snakes)

- Step 1:
  - Initialize the boundary curve (the active contour)
    - Automatically,
    - Manually, or
    - Semi-automatically
- Step 2:
  - The contour moves
    - “Active” contour
    - Looks like a wiggling “snake”
- Step 3:
  - The contour stops moving
    - When many/most points on the contour line up with edge pixels

Initialization

- Good initialization is critical!
  - Especially around small neighboring objects
  - Especially if the image is really noisy/blurry
- Some snake algorithms require initialization entirely inside or outside of the object.
  - It is usually best to initialize on the “cleaner” side of the boundary.
- Clinically, this is often involves a human, who:
  - Marks 1 or more points inside the object
  - Marks 1 or more boundary points
  - And/or—
  - Possibly draws a simple curve, such as an ellipse
Moving the Contour

- Two common philosophies:
  - Energy minimization
    - “Ad-hoc” energy equation describes how good the curve looks, and how well it matches the image
    - Numerically optimize the curve
  - Partial differential equations (PDEs)
    - Start the curve expanding or contracting
    - Points on the curve move more slowly as:
      - They become more curved
      - They lie on top of image “edginess”
    - The curve ideally stops moving when it lies over the appropriate image boundaries

Active Contours:
Energy Minimization

- “Visible” image boundaries represent a low energy state for the active contour
  - ...If your equations are properly set up
  - This is usually a local minima
  - This is one reason why initialization is so important!
- The curve is (typically) represented as a set of sequentially connected points.
  - Each point is connected to its 2 neighboring points.
  - The curve is usually closed, so the “first” and “last” points are connected.
Active Contours: Energy Minimization

- Active contour points ≠ pixels
  - At any given time, each point is located at some pixel location
    - (Think itk::Index or itk::ContinuousIndex)
  - But points move around as the curve moves
  - And neighboring points are usually separated by several pixels
    - This allows room for each point to “move around”

Active Contours: Snake Energy

- Two Terms
  - Internal Energy + External Energy
- External Energy
  - Also called image energy
  - Designed to capture desired image features
- Internal Energy
  - Also called shape energy
  - Designed to reduce extreme curvature and prevent outlier points
Active Contours: External (Image) Energy

- Designed to capture desired image features
- Example:
  \[ E_E = \sum \exp(-\|\nabla f(X_i)\|) \]
  - Measures the gradient magnitude in the image at the location of each snake point

Active Contours: Internal (Shape) Energy

- Designed to reduce extreme curvature and prevent outlier points
- Example:
  \[ E_I = \sum a \|X_i - X_j\| + b \|X_{i-1} - 2X_i + X_{i+1}\| \]
  - Minimizes:
    - How far apart the snake points are from one another
    - How much the curve bends
  - Can add a -d term for avg. segment length
  - Looks like a 2nd derivative kernel
  - “Rubber band tension”
  - \( \beta \) is usually larger than \( \alpha \)
Active Contours: Selecting New Points

- Need choices to evaluate when minimizing snake energy
- Scenario 1: Snake can only shrink
  - Useful to execute between (large) initialization and normal execution
  - Look at points only inside the contour, relative to current point locations
- Scenario 2: Each snake point can move 1 step in any direction
  - Useful if the snake is close to the correct boundary
  - Look at all vertex-connected neighbors of each point’s current location
- Other scenarios possible

Active Contours: Energy Minimization

- Numerical minimization methods
- Several choices
  - In 2D, dynamic programming can work well
  - In 3D (i.e. “active surfaces”), simulated annealing can be a good choice
- Both methods require a finite (typically sampled) number of possible states.
  - The solution obtained is hopefully the best within the set that was sampled, but...
  - If the best solution in the region of interest is not included in the sample set, then we won’t find it!
Active Contours: Partial Differential Equations (PDEs)

- A different method for moving the active contour’s points
- Used by “Level Sets”
- Operates on discrete “time steps”
- Snake points move normal to the curve (at each “time step”).
- Snake points move a distance determined by their speed.

Active Contours using PDEs: Typical Speed Function

- Speed is usually a product of internal and external terms:
  \[ s(x,y) = s_f(x,y)s_E(x,y) \]
- Internal (shape) speed:
  \[ s_f(x,y) = 1 - \| \kappa(x,y) \| \]
  where \( \kappa(x,y) \) measures the snake’s curvature at \((x,y)\)
- External (image) speed:
  \[ s_E(x,y) = (1+\Delta(x,y))^{-1} \]
  where \( \Delta(x,y) \) measures the image’s edginess at \((x,y)\)
Active Contours using PDEs: Typical Problems

- Curvature measurements are very sensitive to noise
  - They use 2nd derivatives
- These contour representations don’t allow an object to split
  - This can be a problem when tracking an object through multiple slices or multiple time frames.
  - A common problem with branching vasculature or dividing cells
- How do you keep a curve from crossing itself?
  - One solution: only allow the curve to grow

Level Sets

- A variation of the PDE framework
- Address the problems on the previous slide
- We will go over these in detail in the next lecture