ECE 435, Medical Image Processing
Assignment 3: Segmentation via Active Contours

Spring 2020

In this assignment, we explore the active contour (or 'snakes') models, which try to find the contours of regions by optimizing an energy function. You may refer to the original paper [1] and the "*L6.2 Active contours - Snakes*" lecture, available on CourseSpaces, for additional references.

There are different criteria taken into account when defining the energy of a snake, v. Typically, the energy is composed by two components (internal and external), where the internal encodes the geometrical structure (elastic energy) of the curved closed contour and the external component encodes image information (e.g., edges) or external forces (defined by the user). The total energy of the snake is calculated using following Equation:

$$E_{total} = \oint_v (E_{internal}(v(s)) + E_{external}(v(s)))ds \tag{1}$$

Active contours may be initialized manually. The user selects a set of seed points around the object of interest, and the algorithm uses an optimization process to decrease E_{total} iteratively and update the snake upon the seed points, until it reaches convergence.

In this assignment, we use the continuity and the smoothness (curvature) of contours to define the internal energy, and the image's edge information to define the external energy. This way, the internal energy of the contour is defined as follows:

$$E_{internal} = E_{cont} + E_{curv} = \frac{1}{2}\left(\alpha(s)\left\|\frac{d\bar{v}}{ds}(s)\right\|^2 + \beta(s)\left\|\frac{d^2\bar{v}}{ds^2}(s)\right\|^2\right). \tag{2}$$

The external energy is based on the image's gradient:

$$E_{external} = -\gamma|\nabla I(x,y)|^2 \tag{3}$$

In Equations 2 and 3, α, β and γ are weights that control the importance of their corresponding components when updating the snake. For example, a

[1] https://link.springer.com/content/pdf/10.1007/BF00133570.pdf

large value for $\alpha(s)$ penalizes changes in the distance between successive points in the contour. A large value for $\beta(s)$ penalizes oscillations in the contour.

To implement the energy equation in the discrete domain, we represent a discrete curve by an array of points $v_i = (x_i, y_i)$ for $i = 0, \ldots, n$, as shown in Fig. 1. The continuous derivatives in Equation 2 for point v_i are replaced by

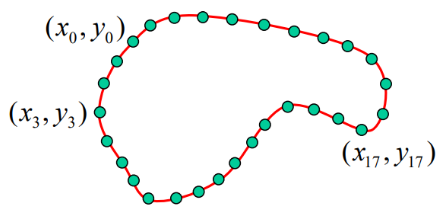


Figure 1: Discrete representation of a curve.

finite differences:

$$\left\| \frac{d\bar{v}}{ds}(s) \right\|^2 \simeq \| v_i - v_{i-1} \| , \tag{4}$$

$$\left\| \frac{d^2\bar{v}}{ds^2}(s) \right\|^2 \simeq \| v_{i-1} - 2v_i + v_{i+1} \| \tag{5}$$

The external energy is defined by the magnitude of the image gradient.

Problems

1. **(5 points)** Implement Algorithm 1 by filling out the MATLAB script template provided in the assignment folder. Each item from 1 through 3 in the algorithm is worth one point, while two points are awarded for the correct implementation of the optimization loop.

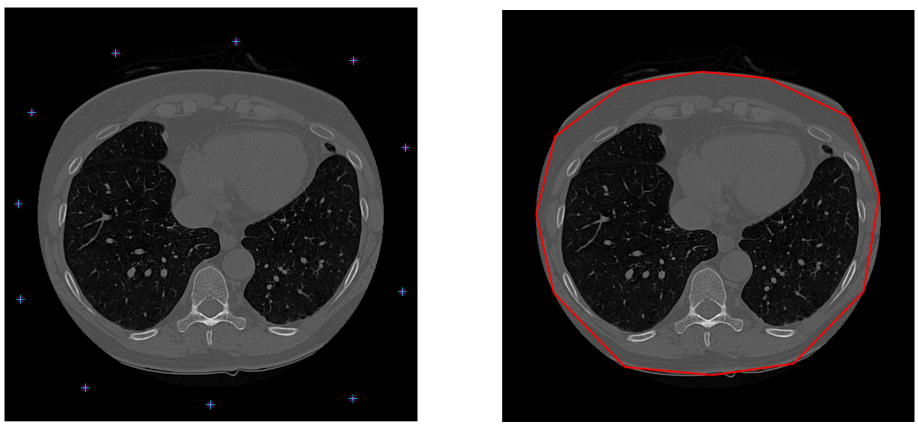
   Hint: To avoid the misleading of the snake progression by the gradient energy in smooth areas, the normalization of E_{grad} in each $k \times k$ window should consider a global insight about the image gradient. To do that, instead of normalizing E_{grad} by $Range = (LocalMaxValue - LocalMinValue)$, normalize the matrix by $max(Range, GradT * (GlobalMaxValue - GlobalMinValue))$. $GradT$ is one of the hyper parameters to set. Local values refer to entries of the $k \times k$ gradient matrix and global ones point to the gradient in the entire image.

2. **(1 point)** When calculating E_{total} in the *for-loop*, what should be the sign of the γE_{grad} term? Why?

3. **(1 point)** Apply your algorithm to the 'CTImage.png' file located in the assignment folder to detect the outer boundary of the shape. Tune the hyper parameters to achieve the best segmentation results.

Full marks are only awarded for segmentation results that (1) are similar to those of Figure 2b; (2) were obtained using 10 seed points roughly around the locations shown in Figure 2a.

4. **(1 Point)** Detect the outer boundary of the left lung in the 'CTImage.png' with active contours using a maximum of 15 seed points. Report your best results and discuss eventual imperfections in the segmentation.



(a) (b)

Figure 2: a) Selected seed points, b) active contour after 100 iterations.

Deliverables: When submitting your assignment, use the following format for the name of the folder possessing all the required files:

'FirstName_VNumber_Assignment3.zip' (e.g. 'Claire_V00888888_Assignment3.zip'). The folder should include:

- A '.pdf' file with your written report, which includes answers to all the questions, screenshots (when applicable), and a description of your results.
- A MATLAB script named as 'SnakesTemp_Solution.m' which possesses the implementation of Algorithm 1.

Algorithm 1: Active Contour Algorithm

Input: Grey level image $ImgIn$
Parameters: α
β
γ
Number of Iterations, max_{itr}
Gaussian Window Length, $GaussWinSize$
Gradient Threshold, $GradT$
Begin:
1. Use Gaussian filter with size $GaussWinSize$ to denoise the input image.
2. Utilizing Sobel vertical and horizontal filters, calculate image gradient, $EdgeImg$.
3. Ask the user to select arbitrary number of seed points, $v_i, i = 1, \ldots, n$, around the boundary of interest in the input image (Hint: use 'getpts' function in Matlab)

for max_{itr} **do**
  for *i in n* **do**
    for *a $k \times k$ neighborhood around each seed point v_i* **do**
      - Calculate the average distance of subsequent points on the snake when v_i is moved in the $k \times k$ window. This gives you a $k \times k$ continuity energy matrix for each v_i.
      - Move v_i in the $k \times k$ window and calculate the curvature of the snake at point v_i using Eq. 5. This gives you a $k \times k$ curvature energy matrix for each v_i.
      - Save the gradient information in the adjacency of v_i in a $k \times k$ gradient energy matrix.
      - Normalize all the energy matrices.
      - Calculate the total energy matrix: $E_{total} = \alpha E_{cont} + \beta E_{curv} \pm \gamma E_{grad}$.
      - Update the location of v_i based on the minimum value of E_{total}.
  Update the seed points and plot the new snake (Hint: use the 'convhull' or 'boundary' function in Matlab)