ECE 4310/6310 Introduction to Computer Vision

Semester project – segmenting foods

This semester project is for ECE 6310 students only. In this project each student must implement an active contour algorithm that allows a user to semi-automatically segment food items in a photograph of multiple foods on a dinner plate. Five images are provided at the course website: bacon-eggs-toast.pnm (3 items), eggs-pancakes-milk.pnm (3 items), hushpuppies-biscuits.pnm (2 items), fish-lemon-rice-greens.pnm (4 items) and macaroni-kale.pnm (2 items). The image names indicate the number of food items in each image, which is also the number of segmented regions expected to be produced (multiple pieces of the same food should be segmented into one region).

The program must load a color PNM image but only needs to display the greyscale version (average of the three color bands). The user should have the following options: left-click to draw around a food item that subsequently automatically shrinks to wrap to the food boundary; right-click a point within a food item that subsequently automatically grows a contour to its outer boundary; shift-click (either button) to manually drag a contour point to a new location.

Specifically:

(1) Left click and hold the mouse button to draw a contour anywhere in the image. The contour should store every pixel location that the user moves the mouse through while the button is held down. When the button is released, the contour should be downsampled to every fifth point. It should be assumed that the contour encloses an area, so that the final point connects to the first point for subsequent active contour processing.

(2) Right click (detected after release) any point in the image. The right click should create a small circular contour centered at the clicked point with radius 10 pixels. The contour should be downsampled to every third point on the circle.

(3) After a left-clicked contour is initialized, an active contour should be iterated implementing the rubber band model. An internal energy term should cause the contour to try to shrink.

(4) After a right-clicked contour is initialized, an active contour should be iterated implementing the balloon model. An internal energy term should cause the contour to try to expand.

(5) Both active contours should use gradients and other internal and external energy terms. The specific terms are up to the student. It is expected that some experimentation will be required to determine what works best. Be creative!

(6) All details of the active contour implementation should be customized for this problem, including number of iterations, window size, weights, and choice of terms. The user should NOT have the option to change these items; the specific choices for all these items are fixed in the program. Again, it is expected that some experimentation will be required to determine what works best.

The program needs to have a graphical user interface. It can be built on top of the plus program previously distributed in class and used for lab #4. The GUI must:

(a) Show the image loaded (greyscale version), redrawing it if the window is uncovered or minimized/unminimized.
(b) Show the full path of points drawn using the left-click method.
(c) Show each active contour during processing as it shrinks/grows.
(d) Upon completion of the active contour algorithm, allow the user to grab a point and manually move it to a new location.

The last option, manually moving a point, should initialize another active contour. This third active contour should have neither the rubber band (shrink) nor balloon (grow) energy term. It should be neutral with respect to size. Instead, it should fix the manually moved point to the given location, not allowing it to move. All the other points should be allowed to move normally. It is expected that internal energy terms, for example even spacing of points, might need to be weighted more heavily, at least perhaps during the first few iterations of this active contour. Once again, it is expected that some experimentation will reveal the best working parameters.

The program does not need to save a result.

You must write a brief report that includes the code. The report should show your best result for each image. It should discuss what worked well and what did not work well, and make a suggestion for what could be tried to improve performance.

The report is due on the last day of class. Reports will be collected in class. Demos will be observed by the grader at a time to be determined.