CS-512 Computer Vision: Assignment 1

Task: Active Contours

In addition to implementing active contours, perform the following image manipulation tasks using OpenCV and Python:

- Allow the user to specify the initial position of the input curve though mouse interaction
- Implement the greedy algorithm for minimizing the energy functional
- Show the evolution of active contour until convergence
- Interactively controlled parameters should include: time delay between iterations, continuity/curvature/external energy coefficients, and the fraction of points that has to reach a local minimum to determine convergence
- Display the results overlaid in color over the original grayscale image

Proposed Solution:

- The energy functional with parametric curve f(s) is defined as:
  E(f(s)) = sum(alpha*continuity_energy + beta*curvature_energy + gamma*external_energy)
- Start with a parametric curve, and begin shrinking it as we greedily minimize the total energy of the image
- Gaussian filter(sigma=3) has been used to perform smoothing

Implementation Details

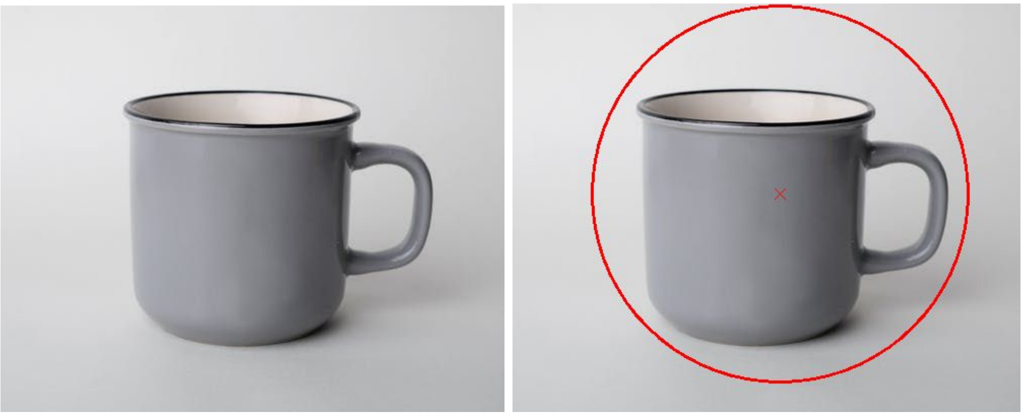
- The initial parametric curve chosen is a circle
- The image window displayed has trackbars for setting the values of: centre of the circle, radius of the circle, alpha, beta, gamma, fraction of points to reach local minima before convergence, time delay (in ms) between iterations and Ready.
- Adjust these values as required and press any key (except ‘q’) to begin fitting the contour.
- In order to begin fitting the contour, make sure that the ‘Ready’ trackbar is switched to 1. Contour fitting will NOT begin otherwise.
- Evolution of contour can be seen 10 iterations at a time, once the contour fitting begins. The current iteration can be seen at the top
- At any point during the program, pressing the key ‘q’ will terminate the currently running operation. For example, if ‘q’ is pressed while the iterations are being performed, the iterations stop and the most recently detected contour is displayed.
- In the final result displayed, the dashed-circle (in red) is the initial circle specified by the user. The contour is displayed by a blue-edged polygon.

- 2 images have been used in this report to showcase the fitting process. Starting with a simpler image (of a cup), it is easier to visualize the process of fitting the contour. The second image is relatively complex (Lenna), to show the robust working of the algorithm.

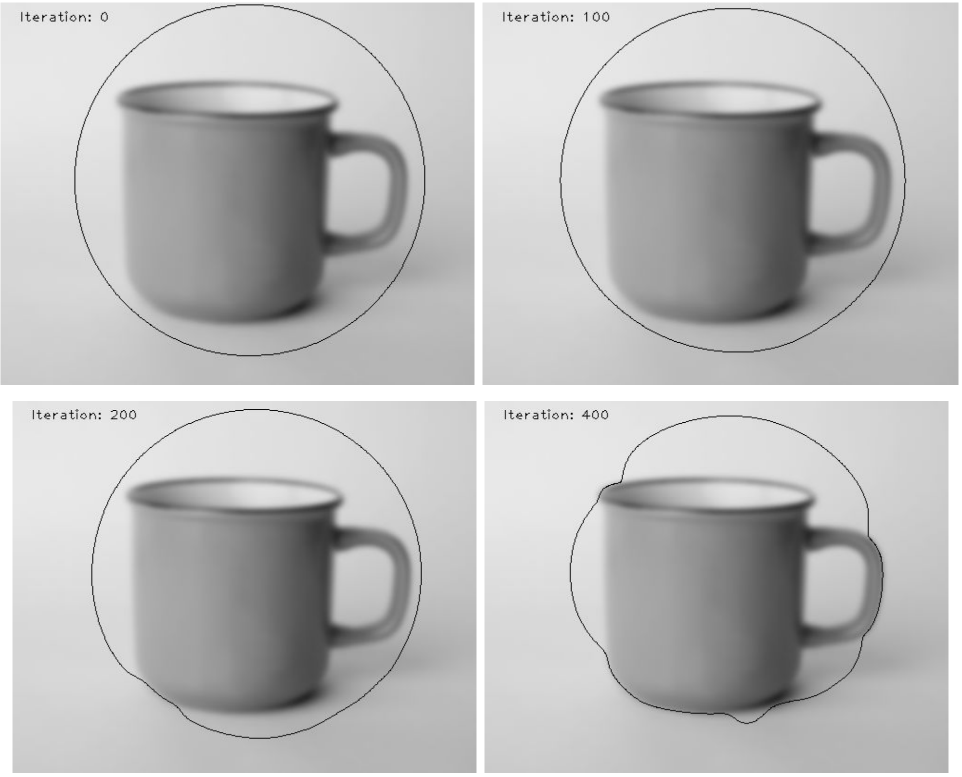
Results:

Image 1: Cup

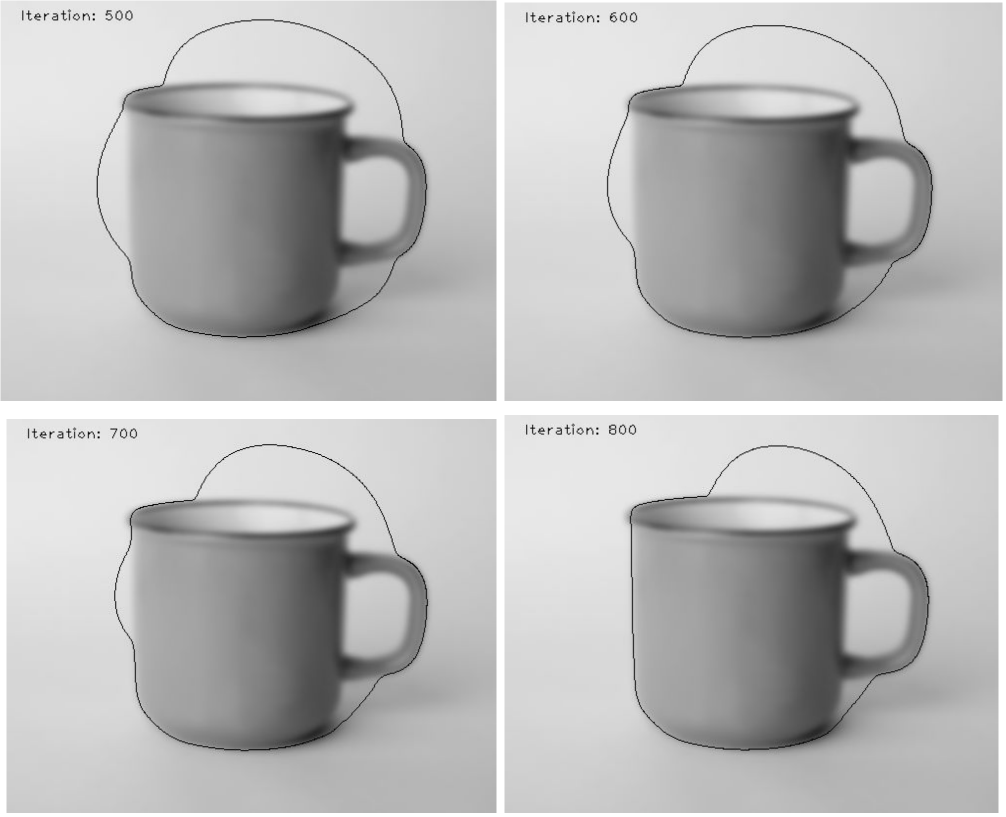
Parameters: alpha=0.05, beta=0.05, gamma=0.01, no. of iterations=1000



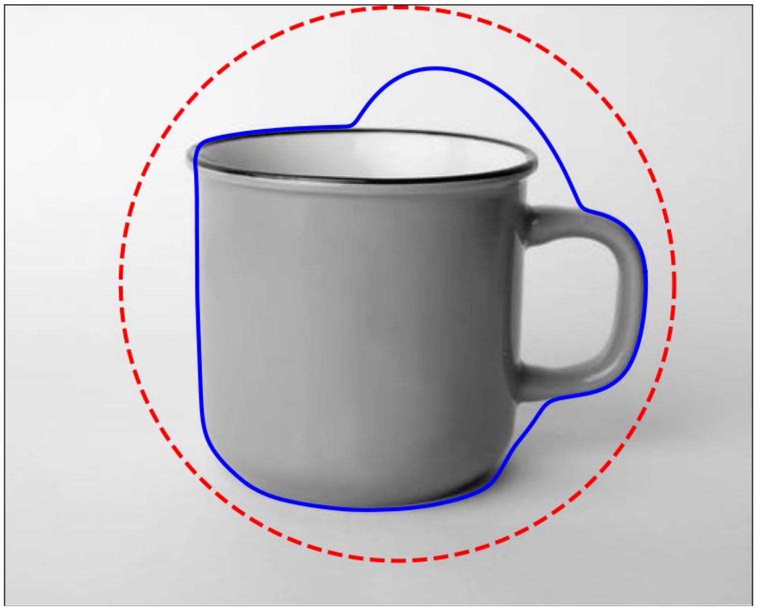
Original image, initially selected region for contour fitting



Evolution of contour every 100 iterations



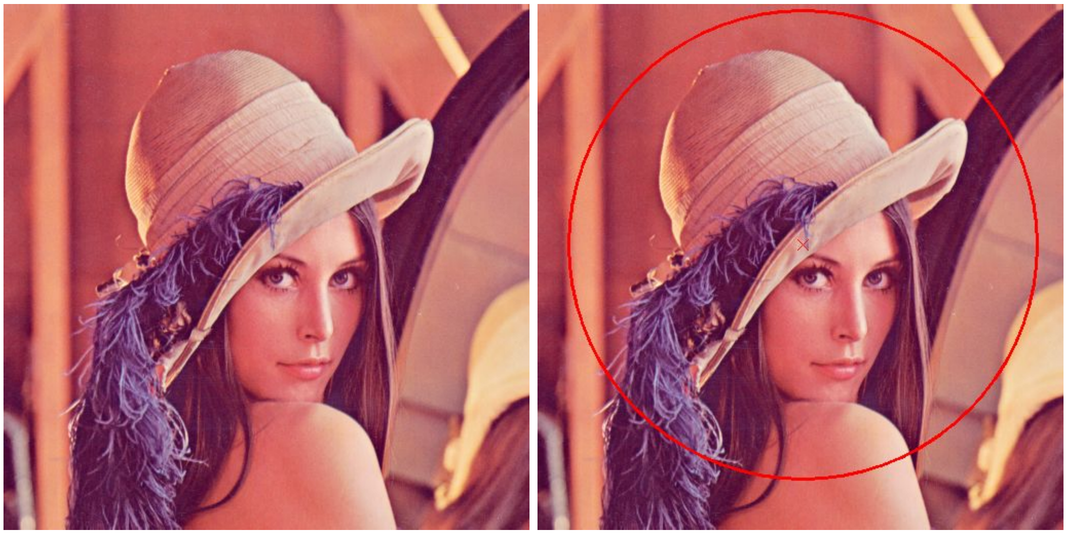
Evolution of contour every 100 iterations



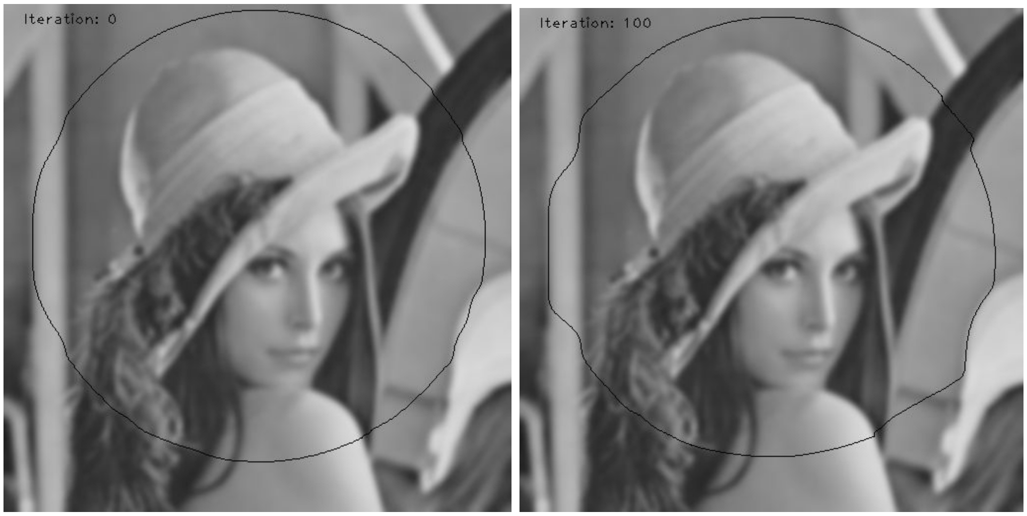
Final result after 1000 iterations

Image 2: Lenna

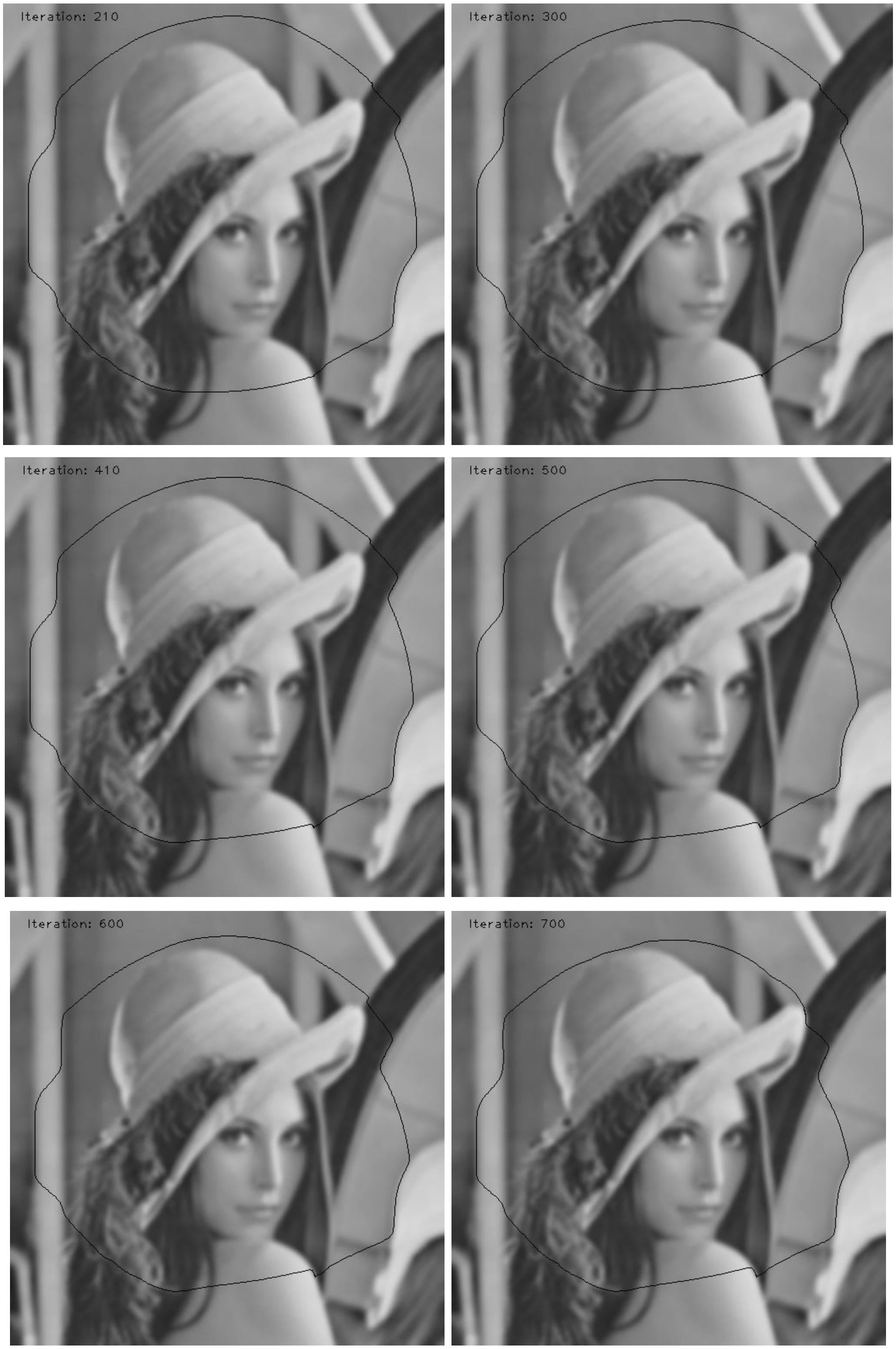
Parameters: alpha=0.05, beta=0.05, gamma=0.01, no. of iterations=1000



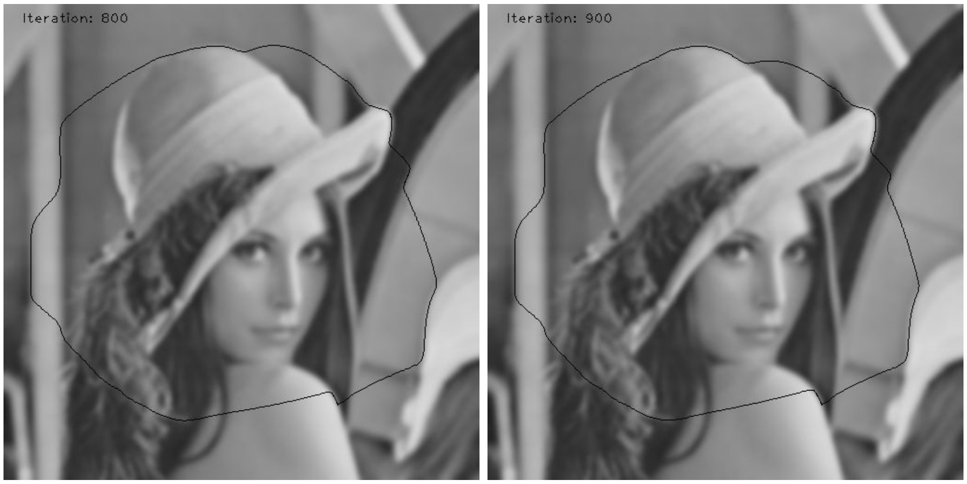
Original image, initially selected region for contour fitting



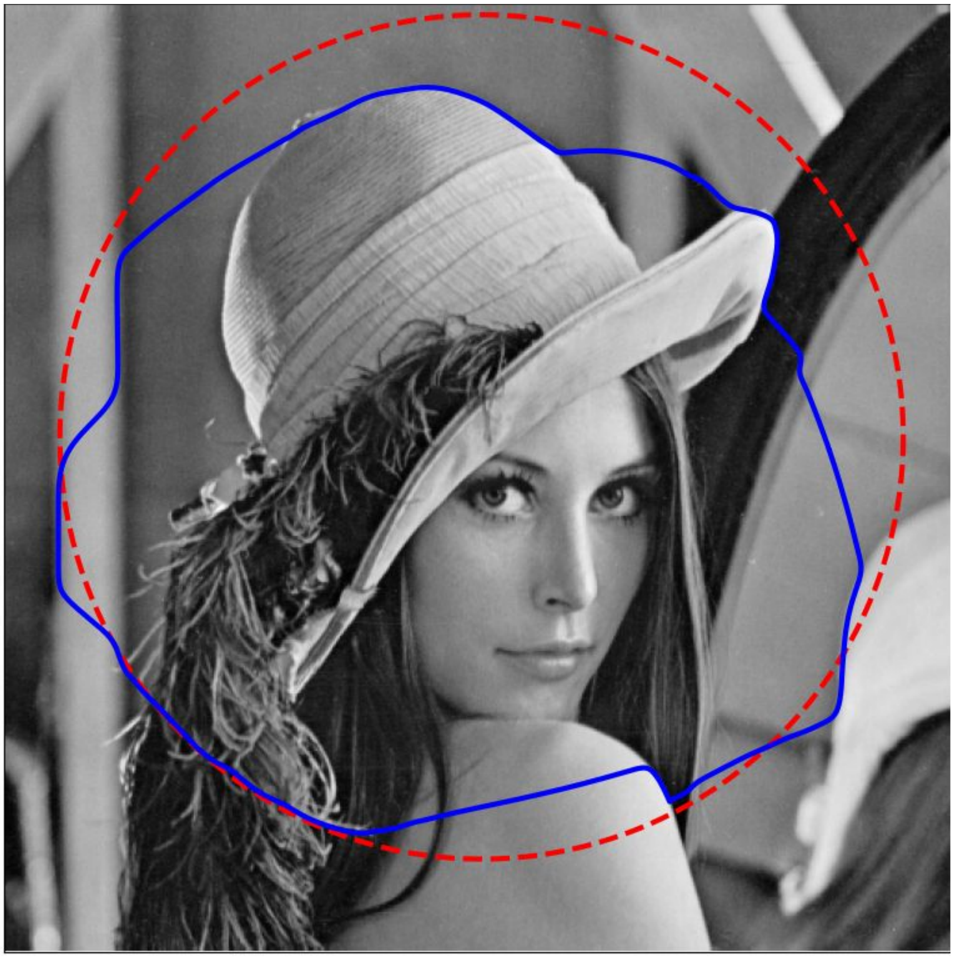
Evolution of contour every 100 iterations



Evolution of contour every 100 iterations



Evolution of contour every 100 iterations



Final Result after 1000 iterations

Inferences:

- As it can be seen from the two examples of contour fitting shown above, the contours start to fit at the expected points in the selected region.
- It is more evident in the example of the cup, how the contours evolve to completely border the cup.
- In the Lenna example, it can be seen that the contour has started to take the shape of the hat and the edges in the background.
- Subsequent iterations show that we can completely fit the contour as per the selected region in the image
- Finally, the user can experiment and see the effects of the different energy coefficients in the contour obtained as a result.

References:

- http://www.cs.iit.edu/~agam/cs512/share/intro-opencv.pdf
- http://www.cs.iit.edu/~agam/cs512/share/filter.pdf
- http://www.cs.iit.edu/~agam/cs512/share/fdetect.pdf
- http://www.cs.iit.edu/~agam/cs512/share/cdetect.pdf
- http://www.cs.iit.edu/~agam/cs512/share/mfit1.pdf
- http://www.cs.iit.edu/~agam/cs512/share/mfitb.pdf
- http://www.cs.iit.edu/~agam/cs512/share/mfitc.pdf
- http://www.cs.iit.edu/~agam/cs512/share/segrec.pdf