Assignment3 Report:

EVOLVE ACTIVE CONTOUR

CSE 691: Image and video processing

Cheng Wang(cwang76@syr.edu)

A. Implementation

Implement the Greedy Algorithm described by Williams and Shah to evolve active contours.

a) Read an image.

b) Compute the smoothed gradient of the image, and find the gradient magnitude at each pixel.

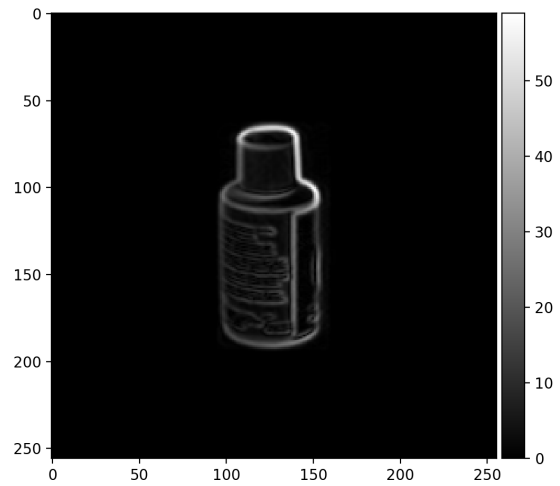
c) Obtain the initial position of a contour from the user.

d) If the distance between the points in the user input is large, interpolate to add extra points. The distance between the points should be around 5 pixels.

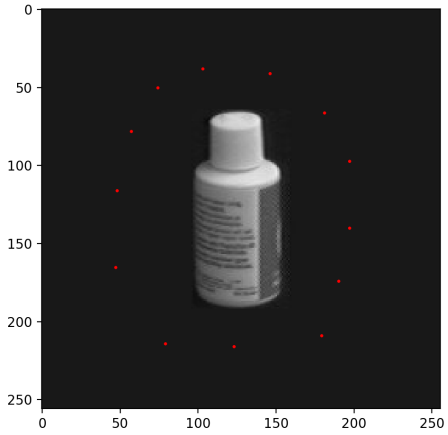
e) Implement the rest of the Greedy Algorithm according to the paper by Williams and Shah.

Use image1 as a example, I will demonstrate the process of this algorithm.

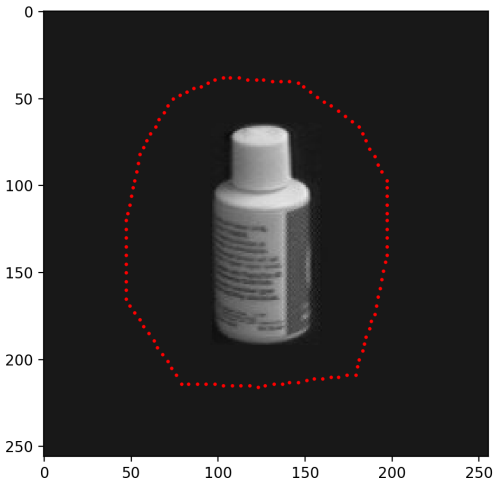
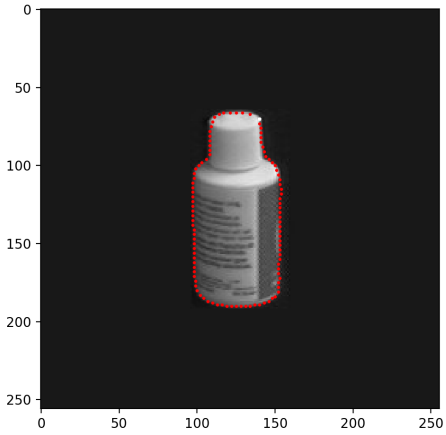
- First, read in a image.
- Compute the smoothed gradient of the image using the functions of last assignment.
- Then get the points interactively.
- Implement the interpolation after we get the points.
- Then apply the greedy algorithm.



Magnitude Image



Get the points manually

	
Interpolate	**evolve active contours**

B. Experiment, evaluation and comments

These parameters are:

– The width of the Gaussian used in the smoothed gradient computation.

– The size of the neighborhood that is searched to move a point p.

– α, β and γ terms.

– The minimum fraction of points that must move in each iteration before convergence.

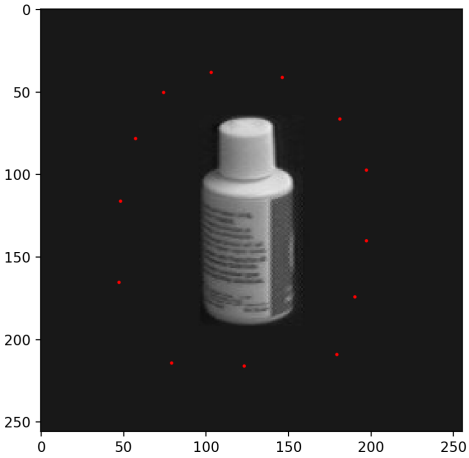
a) [8%] Start with the following set of parameters, and test your code on images "Image1.jpg" through "Image8.jpg":

The width of Gaussian used for smoothing: 3 pixels, the neighborhood size: 3x3, α = 1, β = 1, γ = 1, the fraction: 10%. Please show the initial contour, the final contour, and also two intermediate steps during the evolving of each active contour. In addition, please mark the points where the corners are allowed.

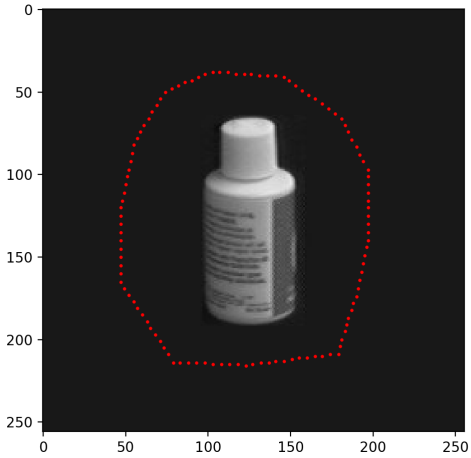
My outputs are listed below:

Contour points are marked in red and the corner points are marked in white

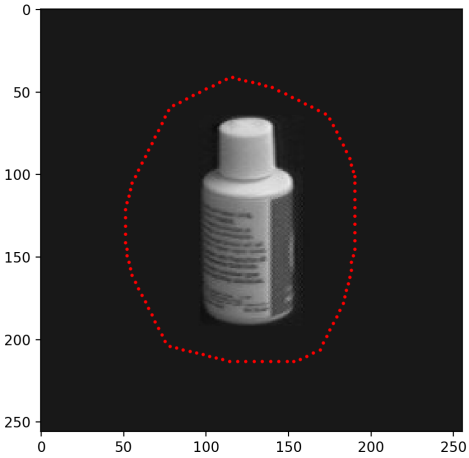
Image 1



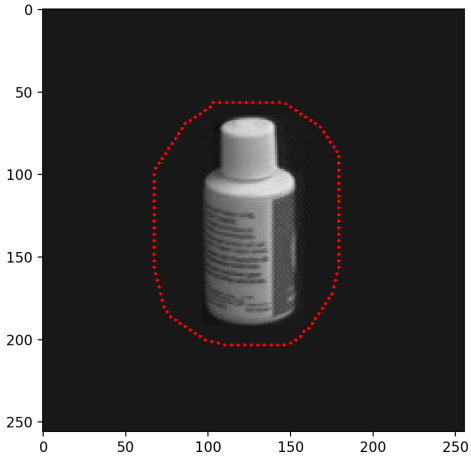
Get User input points



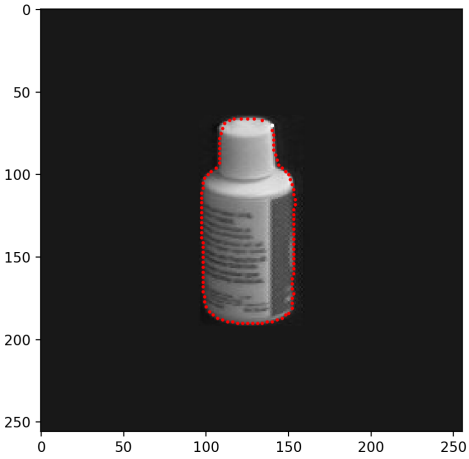
Interpolate



After 10 interation



After 30 interations

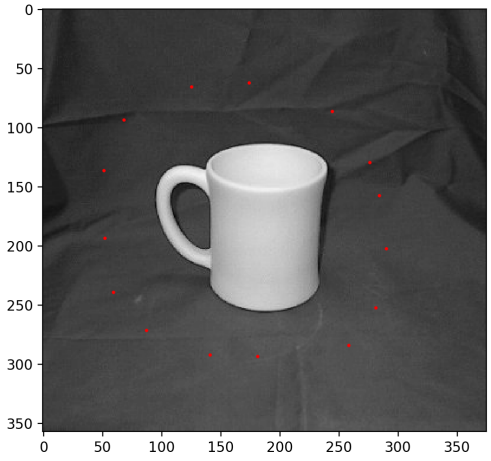


contor

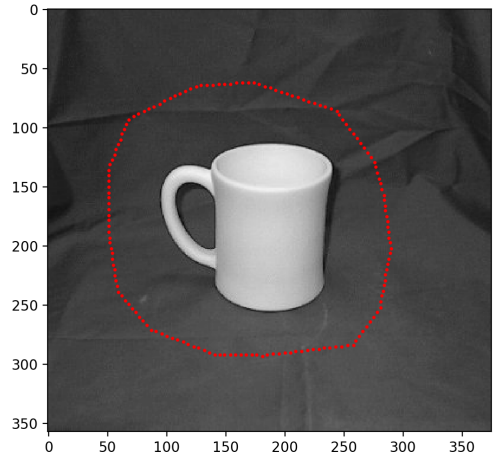
Parameters:

THRESHOLD1: 0.5
THRESHOLD2: 50
THRESHOLD3: 10%
SIGMA:3
SIZE OF NEIGHBORHOOD: 9
ALPHA: 1
BETA: 1
GAMMA: 1

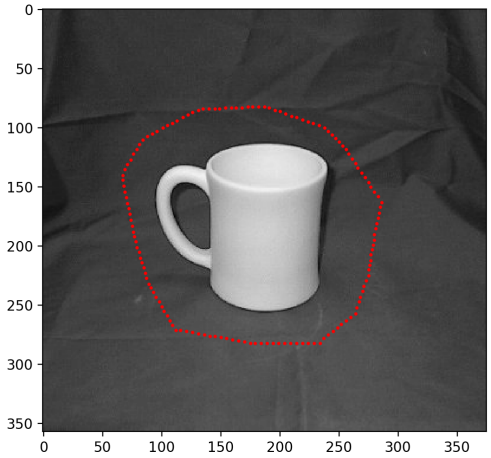
Image 2



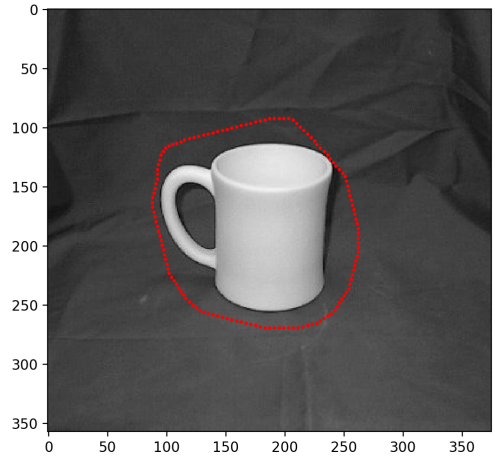
Get User input points



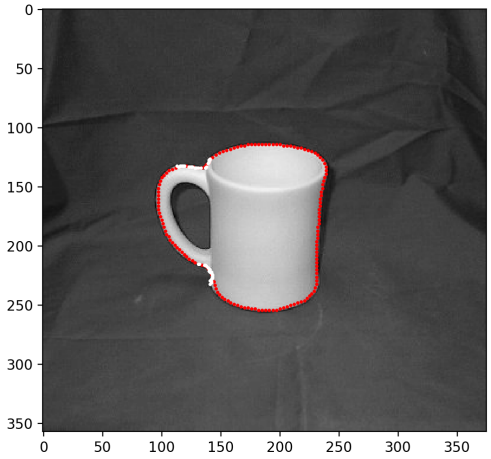
Interpolate



After 10 interation



After 30 interations

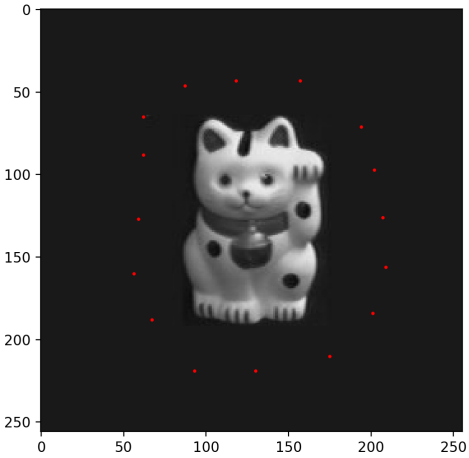


contor

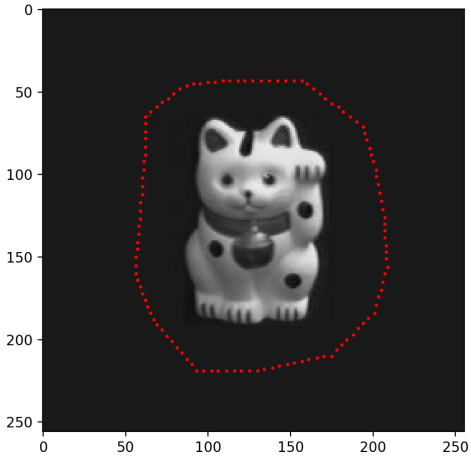
Parameters:

THRESHOLD1: 1
THRESHOLD2: 10
THRESHOLD3: 10%
SIGMA:3
SIZE OF NEIGHBORHOOD: 9
ALPHA: 1
BETA: 1
GAMMA: 1

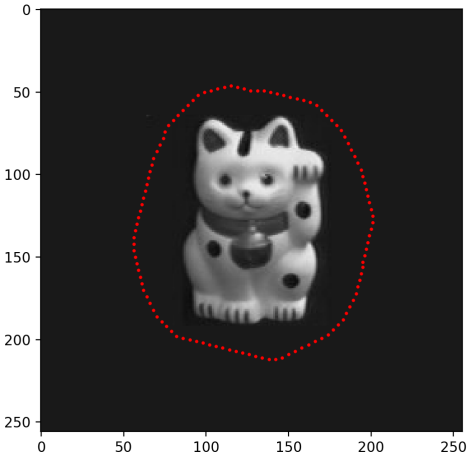
Image 3



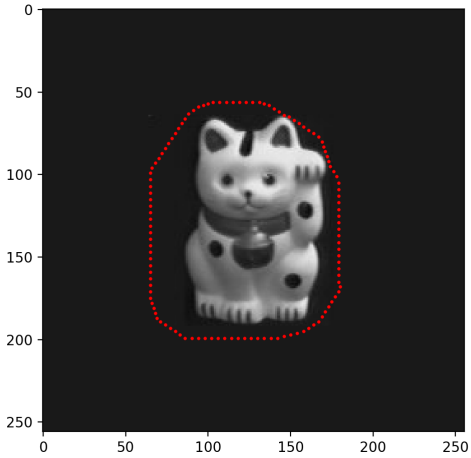
Get User input points



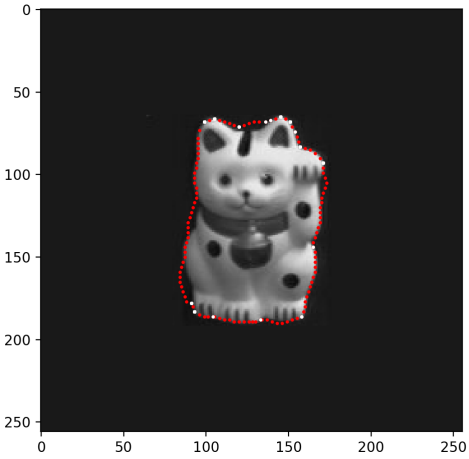
Interpolate



After 10 interation



After 30 interations

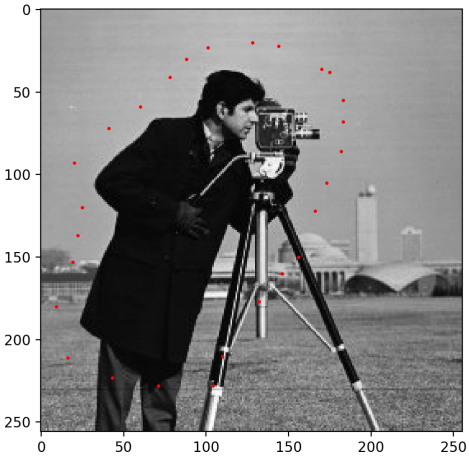


contor

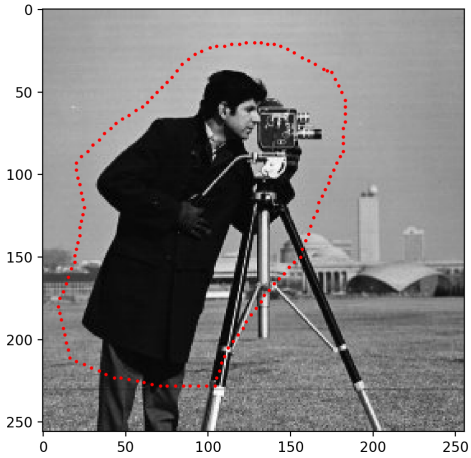
Parameters:

THRESHOLD1: 0.3
THRESHOLD2: 10
THRESHOLD3: 10%
SIGMA:3
SIZE OF NEIGHBORHOOD: 9
ALPHA: 1
BETA: 1
GAMMA: 1

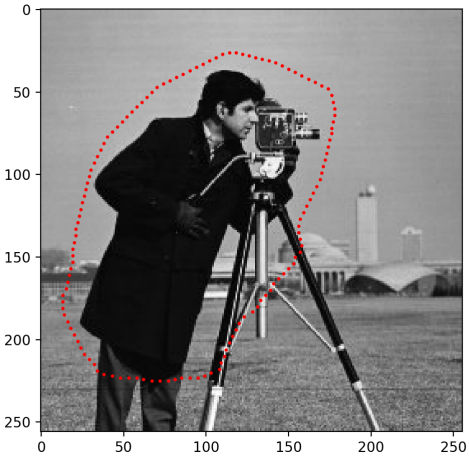
Image 4



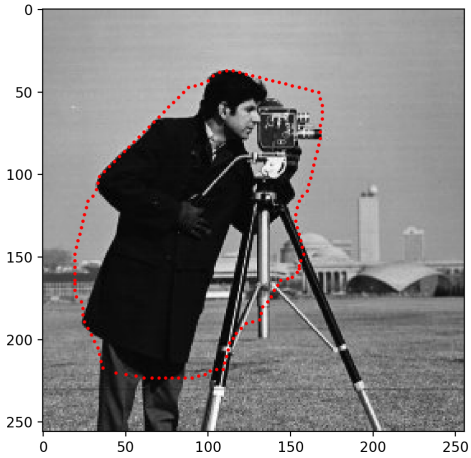
Get User input points



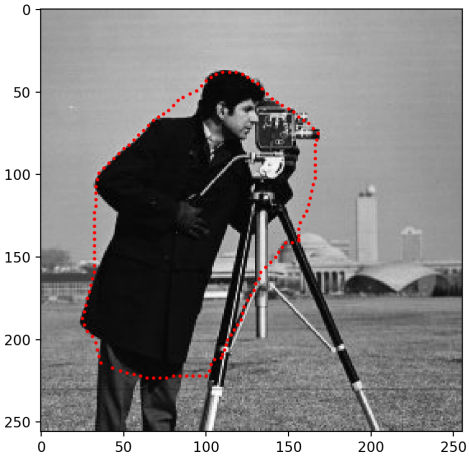
Interpolate



After 10 interation



After 30 interations

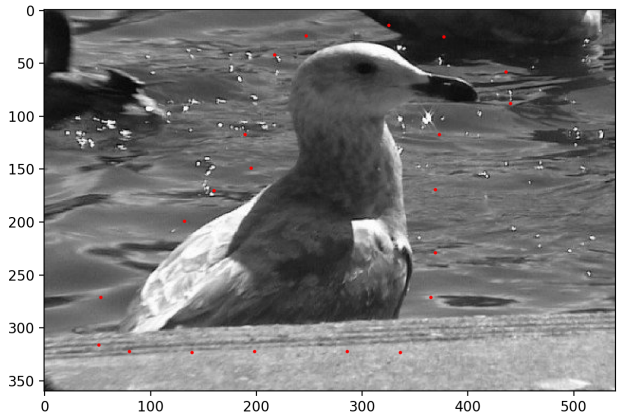


contor

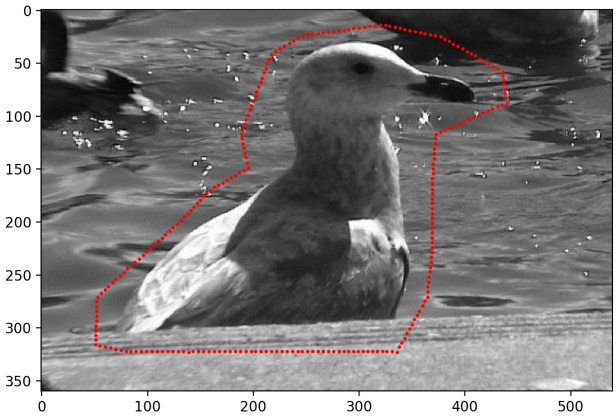
Parameters:

THRESHOLD1: 0.5
THRESHOLD2: 50
THRESHOLD3: 10%
SIGMA:3
SIZE OF NEIGHBORHOOD: 9
ALPHA: 1
BETA: 1
GAMMA: 1

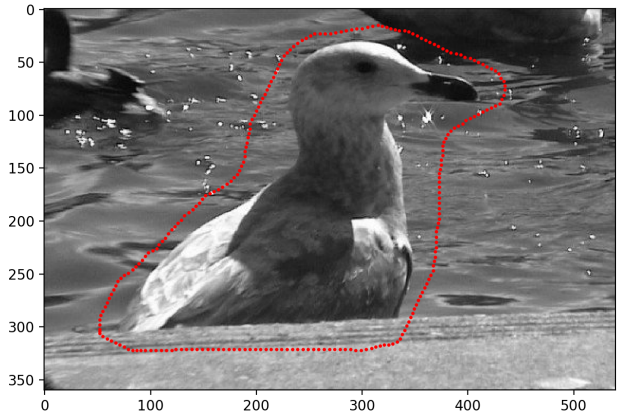
Image 5



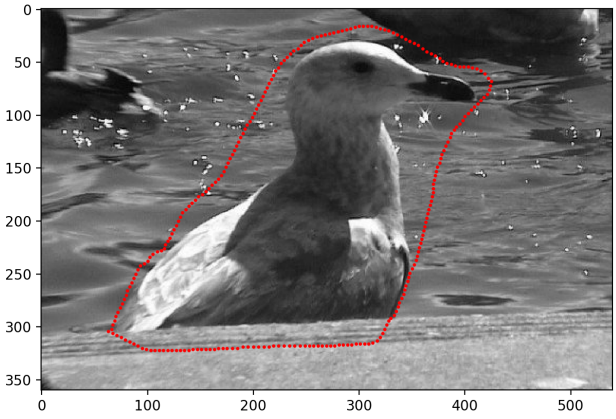
Get User input points



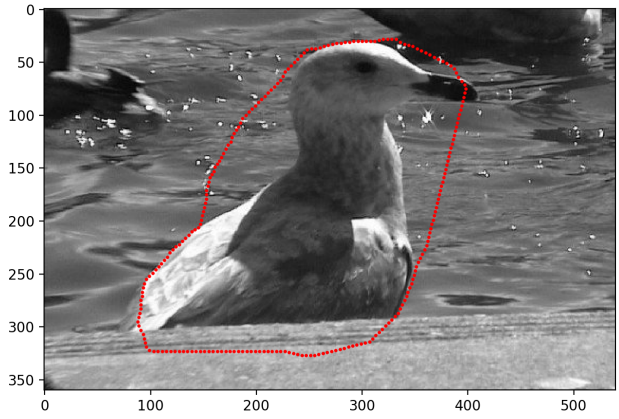
Interpolate



After 10 interation



After 30 interations

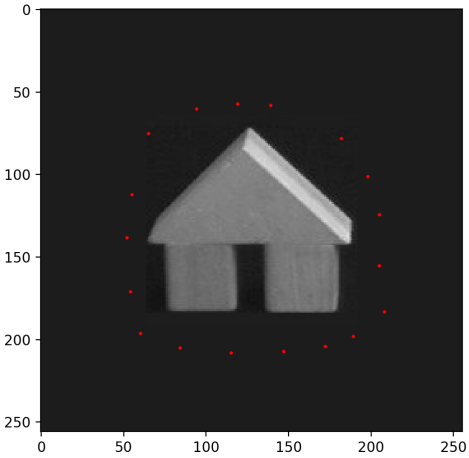


contor

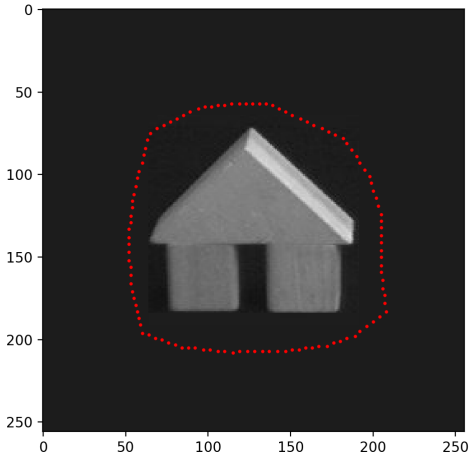
Parameters:

THRESHOLD1: 0.5
THRESHOLD2: 50
THRESHOLD3: 10%
SIGMA:3
SIZE OF NEIGHBORHOOD: 9
ALPHA: 1
BETA: 1
GAMMA: 1

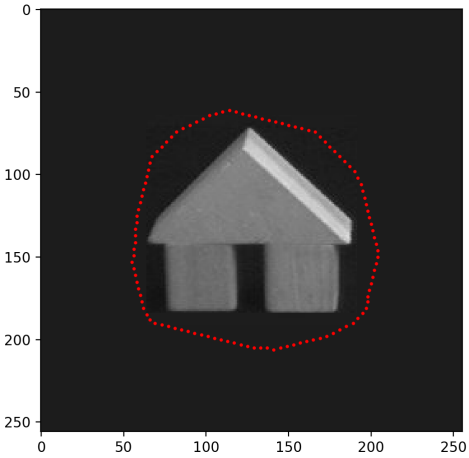
Image 6



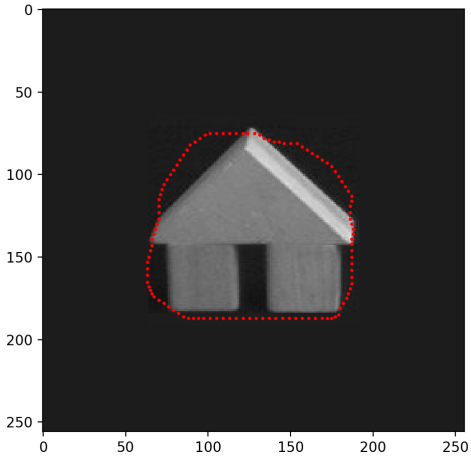
Get User input points



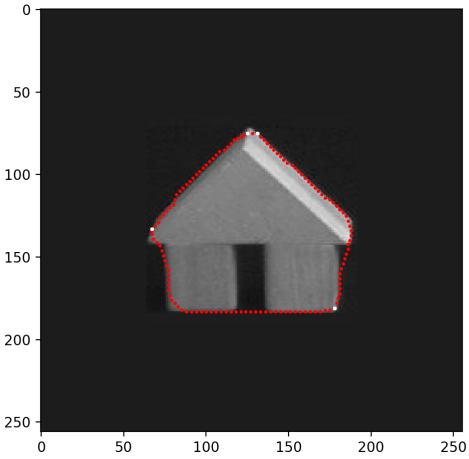
Interpolate



After 10 interation



After 30 interations

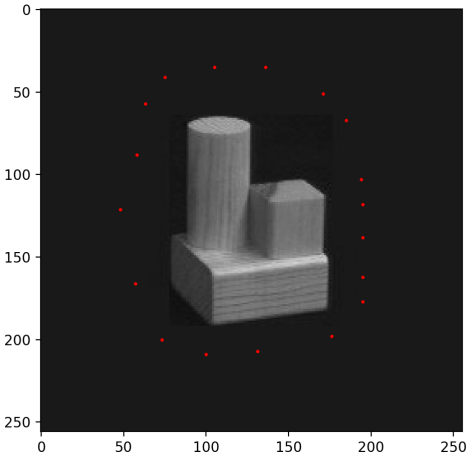


contor

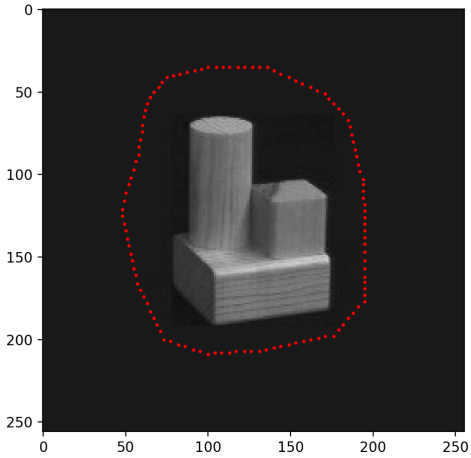
Parameters:

THRESHOLD1: 0.3
THRESHOLD2: 10
THRESHOLD3: 10%
SIGMA:3
SIZE OF NEIGHBORHOOD: 9
ALPHA: 1
BETA: 1
GAMMA: 1

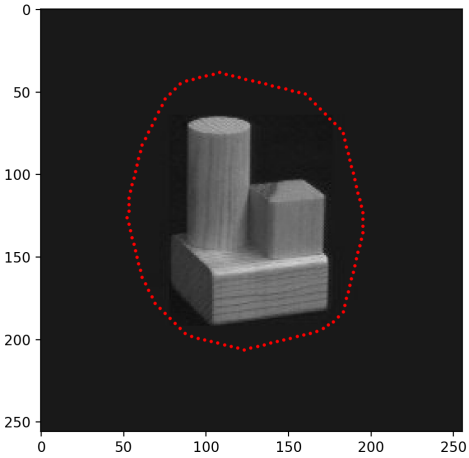
Image 7



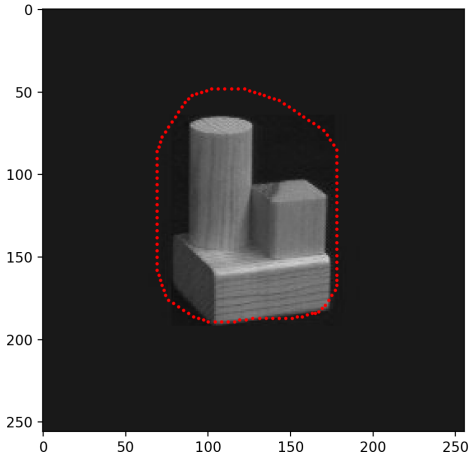
Get User input points



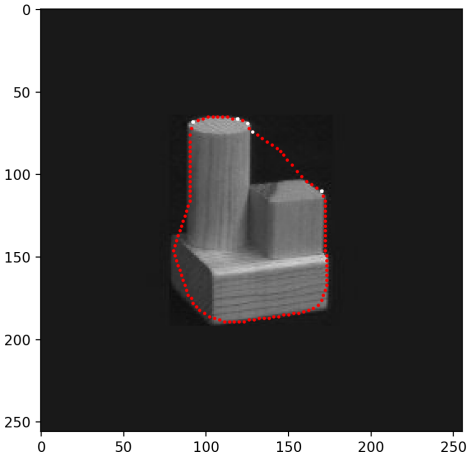
Interpolate



After 10 interation



After 30 interations

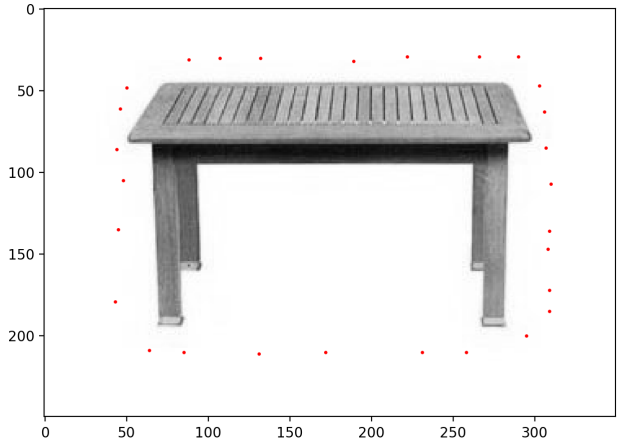


contor

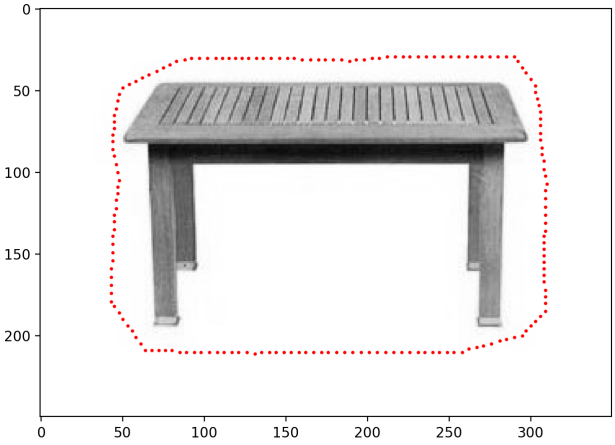
Parameters:

THRESHOLD1: 0.2
THRESHOLD2: 10
THRESHOLD3: 10%
SIGMA:3
SIZE OF NEIGHBORHOOD: 9
ALPHA: 1
BETA: 1
GAMMA: 1

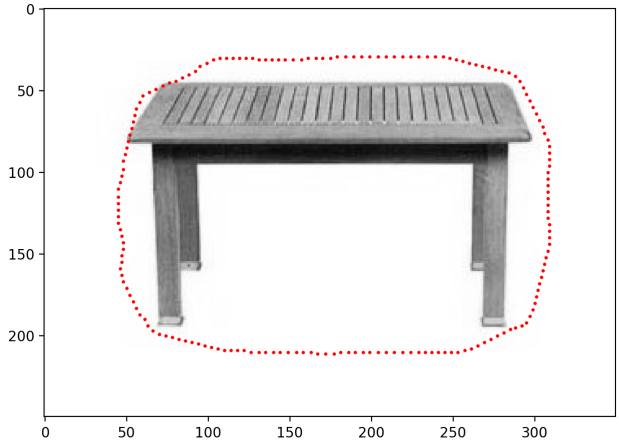
Image 8



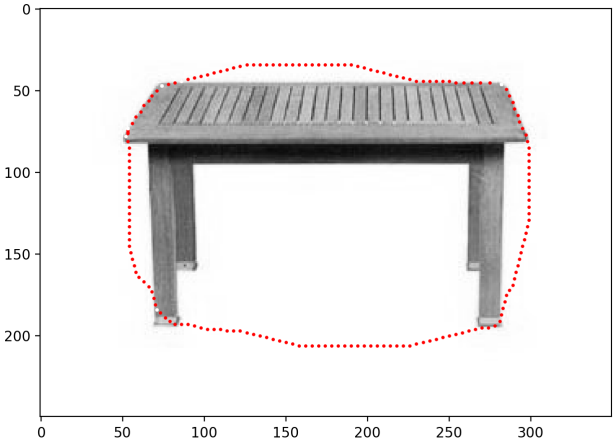
Get User input points



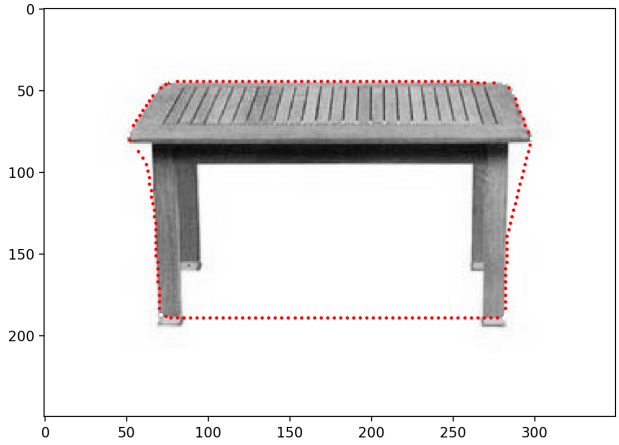
Interpolate



After 10 interation



After 30 interations



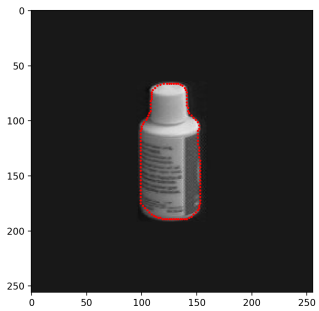
contor

Parameters:

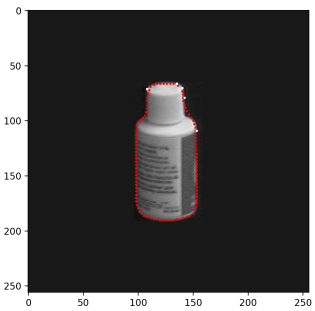
THRESHOLD1: 0.1
THRESHOLD2: 10
THRESHOLD3: 10%
SIGMA:3
SIZE OF NEIGHBORHOOD: 9
ALPHA: 1
BETA: 1
GAMMA: 1

b) [5%] Vary the width of Gaussian used for smoothing and comment on the effects.

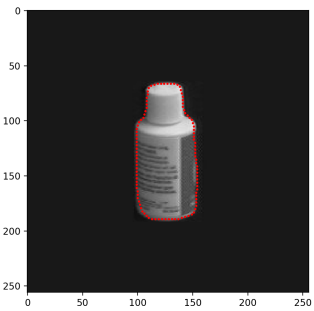
Vary the width of Gaussian used for smoothing-Image 1



Sigma = 1

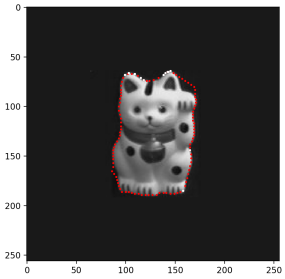


Sigma = 2

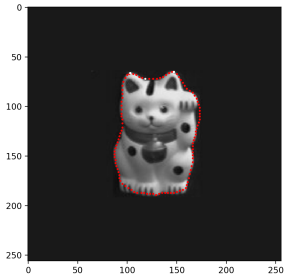


Sigma = 3

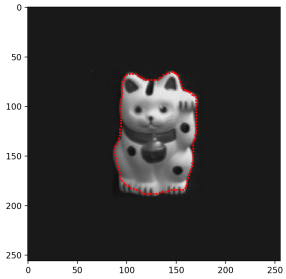
Vary the width of Gaussian used for smoothing-Image 3



Sigma = 1

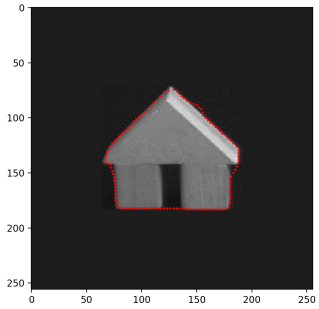


Sigma = 2

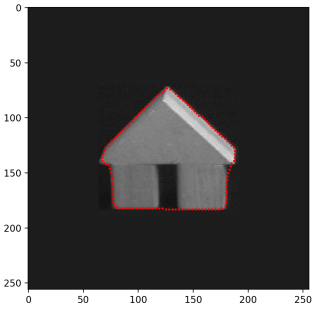


Sigma = 3

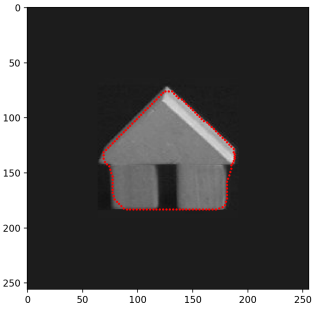
Vary the width of Gaussian used for smoothing-Image 6



Sigma = 1



Sigma = 2



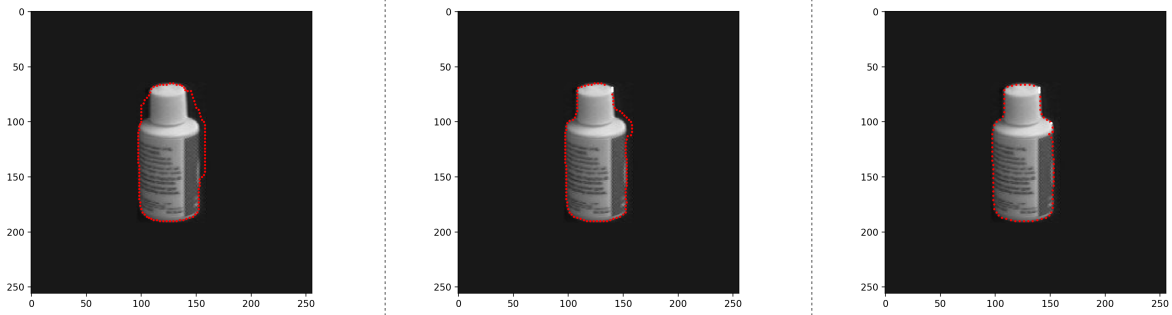
Sigma = 3

Comment:

The width of Gaussian controls how blur the image is or how sharp the edge is. When we evolve the contour, sharper the edge is, the contour are more possible to be around the real edge. Otherwise, the contour would converge inside the real edge. As we can see from the output, the contour converge into inner shape with greater sigma value.

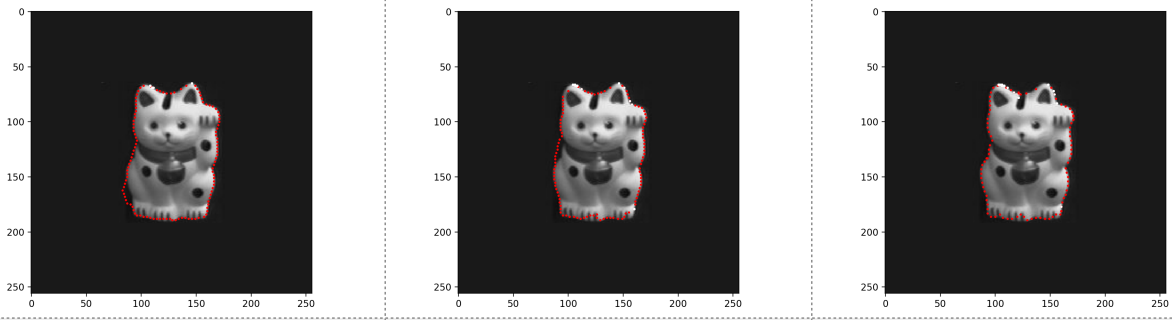
c) [5%] Vary the size of the search neighborhood and comment on the effects.

Vary the size of Neighborhood-Image 1



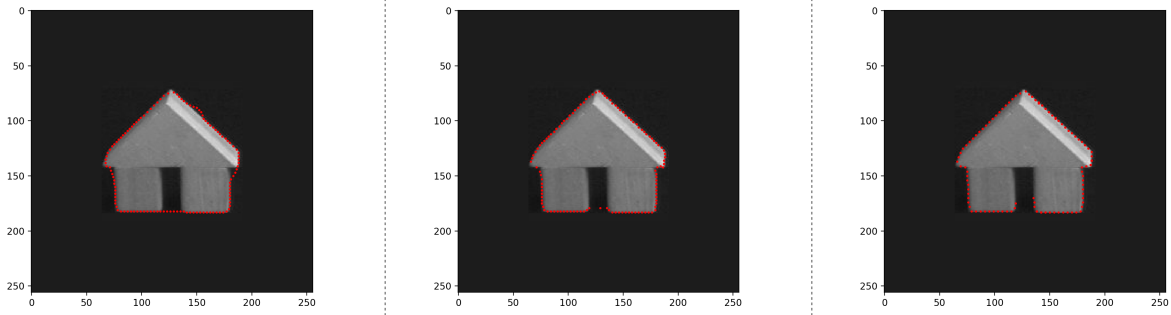
size = 9	size = 25	size = 49

Vary the size of Neighborhood-Image 3



size = 9	size = 25	size = 49

Vary the size of Neighborhood-Image 6



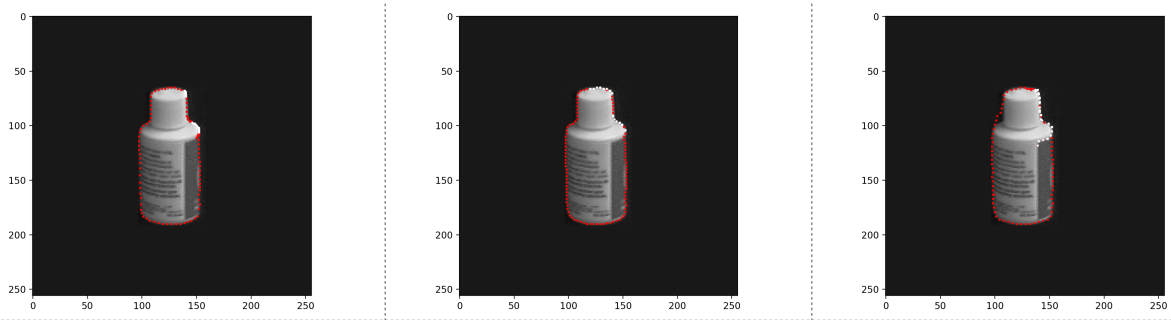
size = 9	size = 25	size = 49

Comment:

With the same other parameters, when we increase the size of neighborhood, the contour are more likely to converge fit to the edge. Every iteration, we move the point to its neighbor position based on the energy term, which means if the size is greater then we have more candidates. Because we terminate the loop when few points are moving. With the small size, one point are more likely to be trapped in its position. The output shows size 49 gives us the best result.

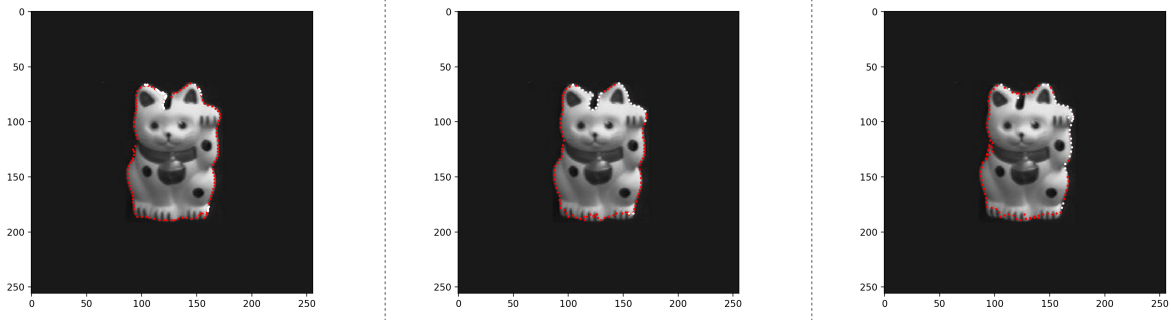
d) [3%] Vary α and comment on the effects.

Vary the alpha - Image 1



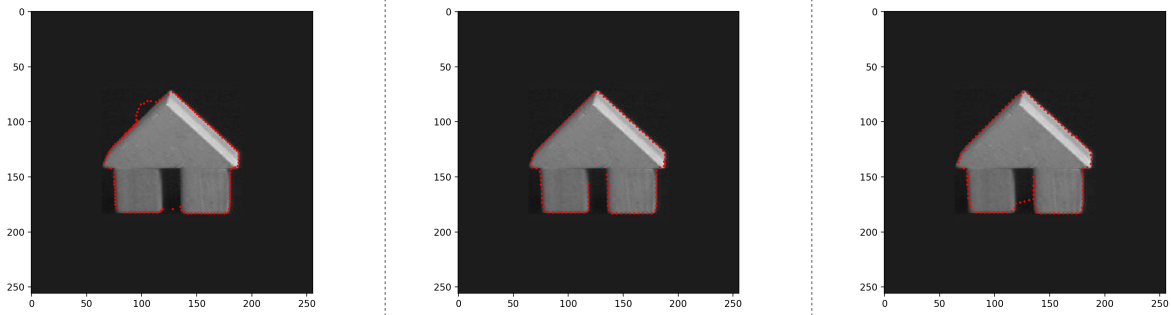
alpha = 0.5	alpha = 1	alpha = 3

Vary the alpha - Image 3



alpha = 0.5	alpha = 1	alpha = 3

Vary the alpha - Image 6



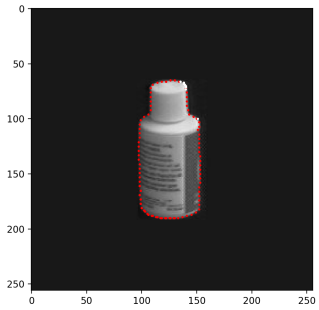
alpha = 0.5	alpha = 1	alpha = 3

Comment:

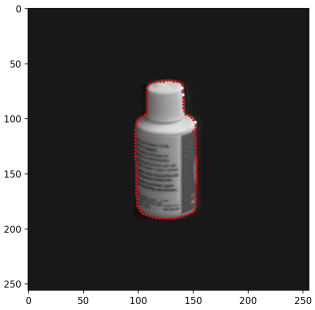
The alpha is the coefficient of the continuity term. The continuity term measures the continuity feature of the contours. Also, another job of this term is to avoid points chaining together and bunching up on strong portions of the contour. When alpha is small, say 0.5, some points gather together and overlap each other. When the alpha is greater, like 3, the first term get overweighted, then other two terms are not working good. The contour can not converge well as a result. Alpha = 1 is the best choice.

e) [3%] Vary β and comment on the effects.

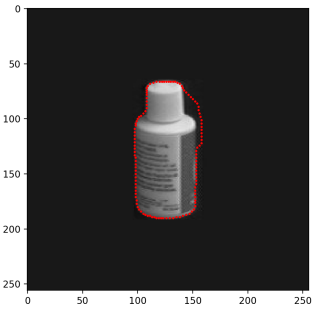
Vary the beta - Image 1



beta = 0.5

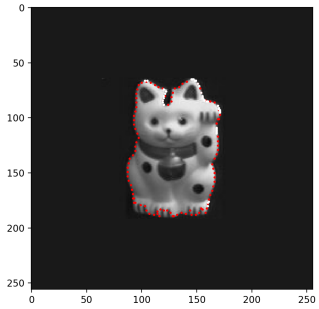


beta = 1

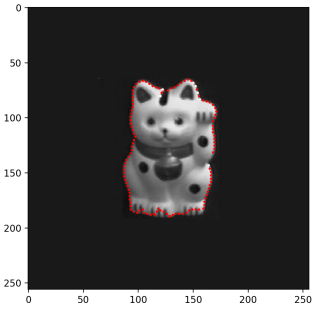


beta = 3

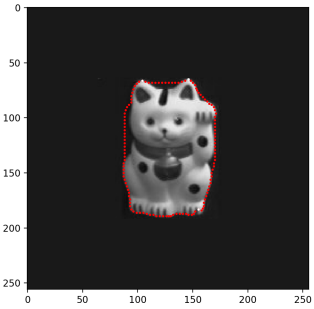
Vary the beta - Image 3



beta = 0.5

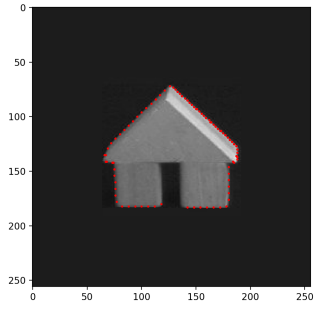


beta = 1

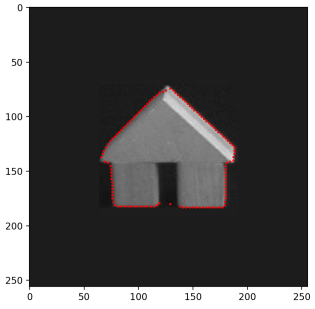


beta = 3

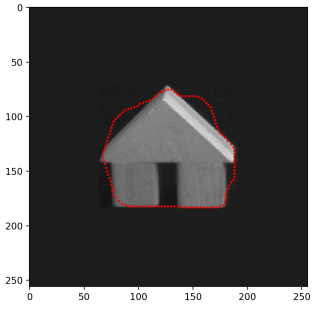
Vary the beta - Image 6



beta = 0.5



beta = 1



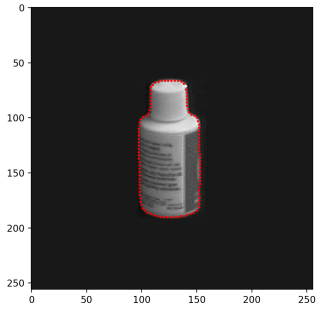
beta = 3

Comment:

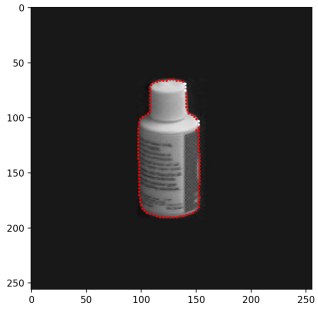
Since the continuity term causes the points to be relatively evenly spaced, this term gives a estimate of curvature. When there is a corner, the curvature gets greater. When the beta is small, say 0.5, we can see that the distance between points are too big or too small. When beta is too big, say 3, the image term is underestimated, so the contour can not fit to the edge properly.

f) [3%] Vary γ and comment on the effects.

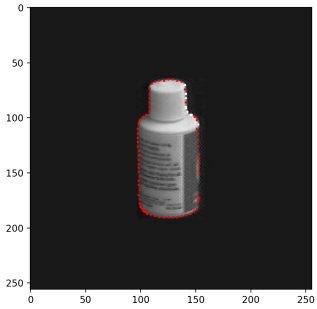
Vary the gamma - Image 1



gamma = 0.5

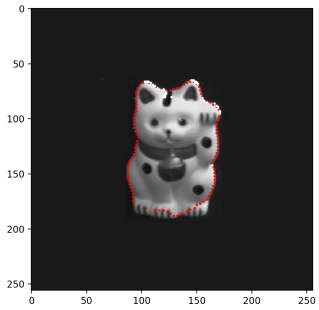


gamma = 1

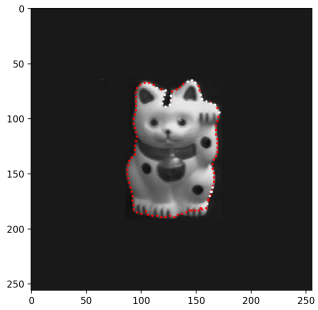


gamma = 3

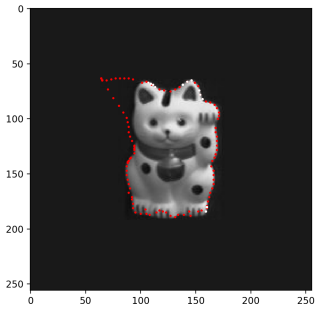
Vary the gamma - Image 3



gamma = 0.5

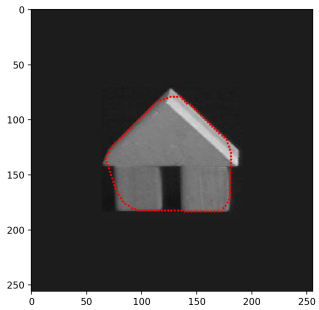


gamma = 1

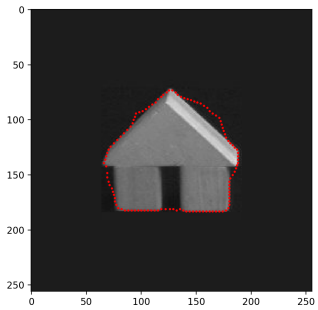


gamma = 3

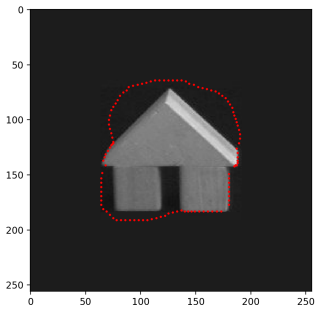
Vary the gamma - Image 6



gamma = 0.5



gamma = 1



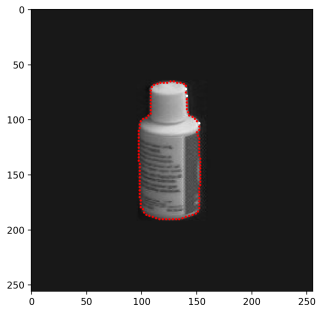
gamma = 3

Comment:

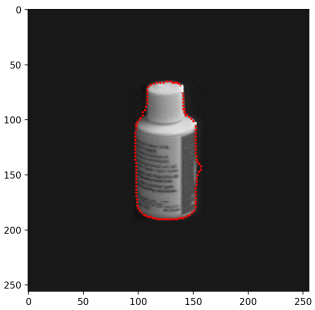
The parameter gamma is the coefficient of the third term-image force term, which is responsible for pulling the contour to the shape. The image force term is a negative term. When the gamma is too small, the force gets great. We can see from the output that when gamma = 0.5, the contour is inside the shape. On the other hand, when gamma = 3, the force is not enough to pull back the contour.

g) [3%] Vary the fraction and comment on the effects.

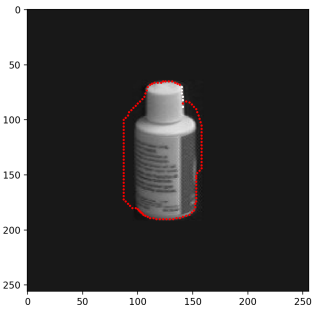
Vary the fraction - Image 1



fraction = 1%

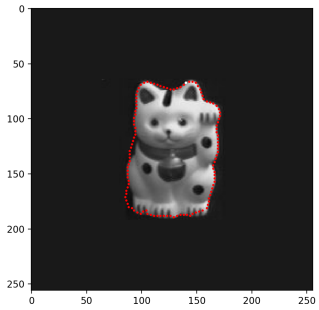


fraction = 10%

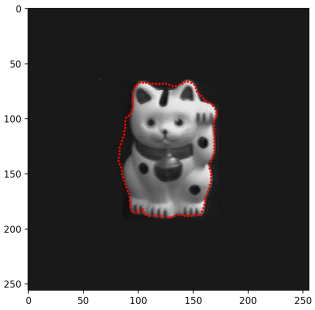


fraction = 30%

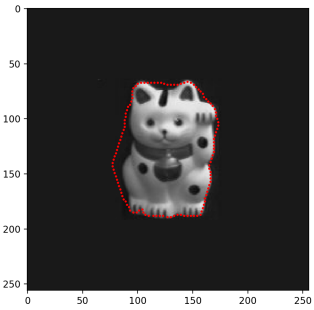
Vary the fraction - Image 3



fraction = 1%

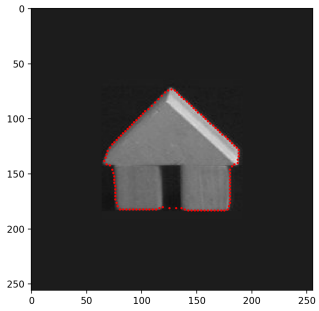


fraction = 10%

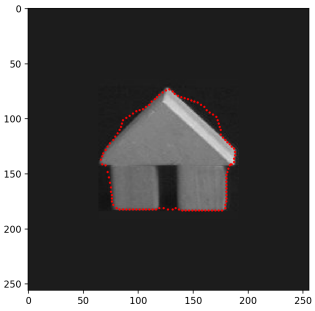


fraction = 30%

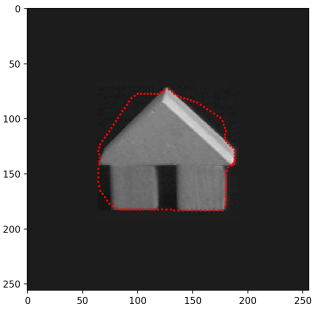
Vary the fraction - Image 6



fraction = 1%



fraction = 10%



fraction = 30%

Comment:

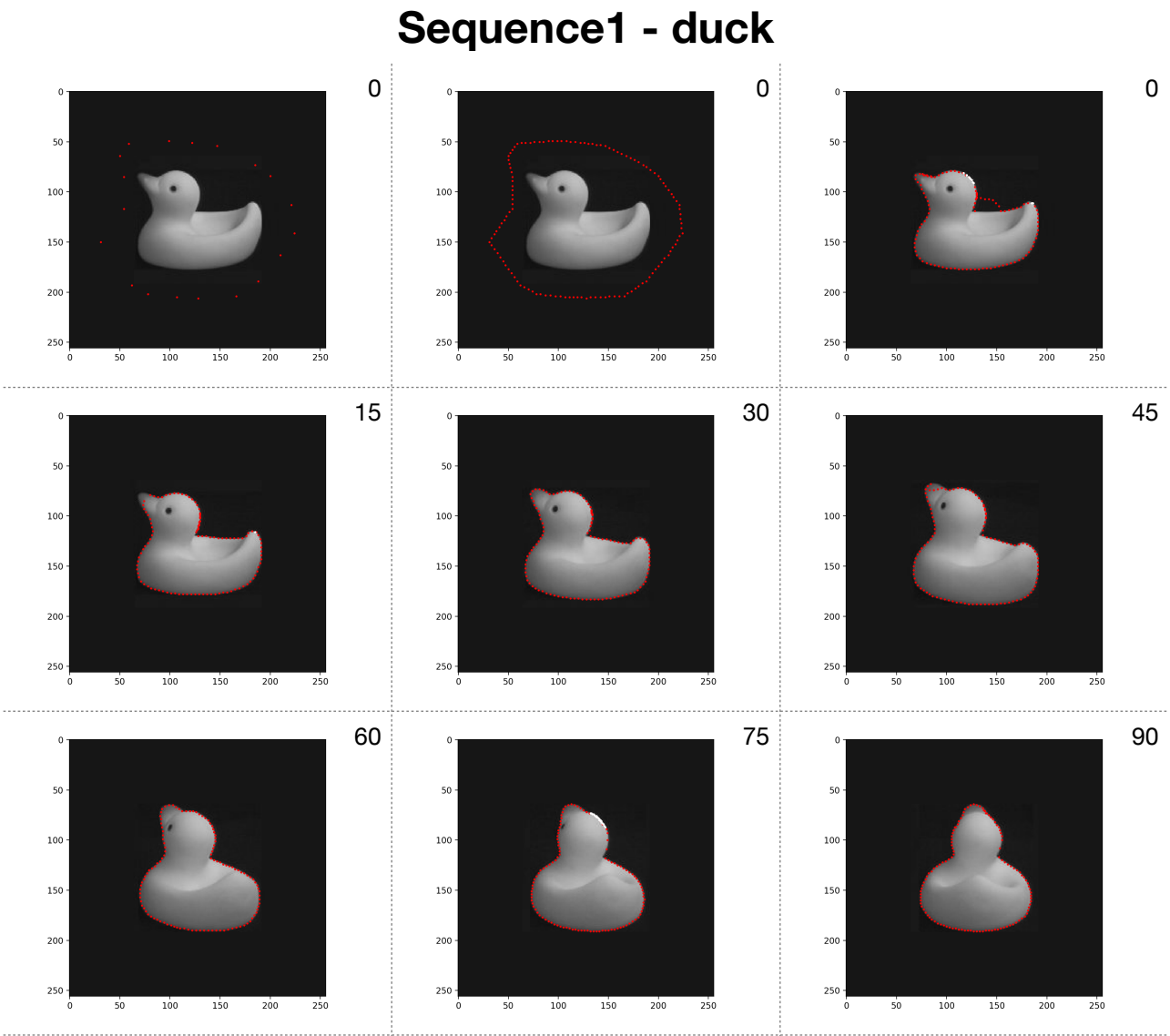
The fraction threshold determines when we terminate the loop. As a result, we set a low threshold, the loop runs longer. If we set a high threshold, the loop would terminate earlier and the contour is less likely to converge to the edge. We can see from the output, when the fraction is 1%, we can have a good contour. But with a fraction of 30%, the contour is not what we want because it stoped before it reach the edge.

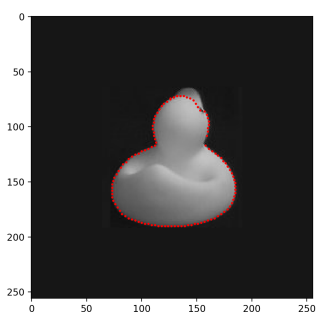
Image Sequences (10%)

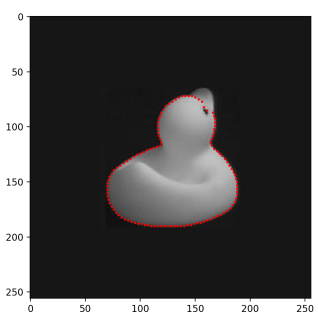
Use the images under the "Sequence1" and "Sequence2" folders to test your code on moving images.

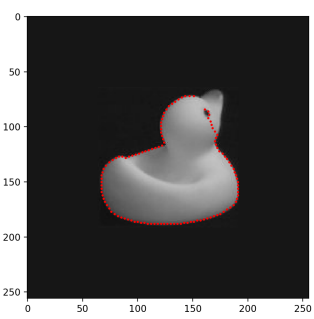
First, run your code on the first image of a sequence by manually entering the initial contour. Then, run your code on the rest of the images in that sequence by using the final contour you obtain from the previous step as the initial contour for the next image. Show the final contour you obtain for each image of each sequence.

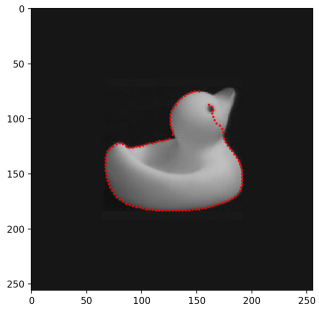
My outputs are listed below:

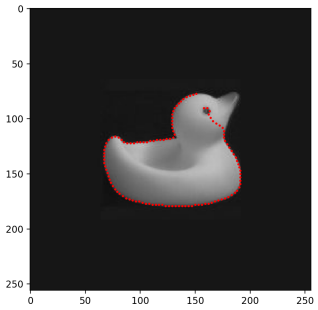


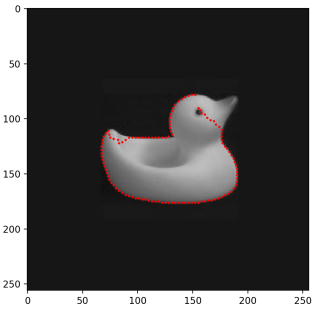

105

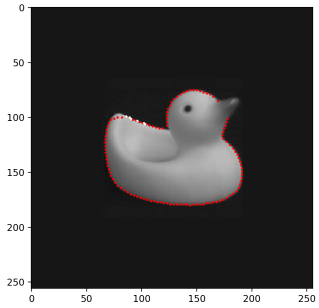

120

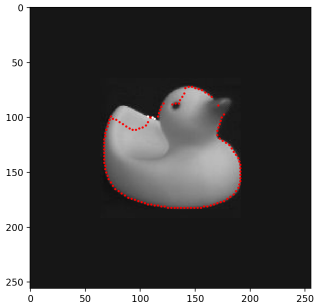

135

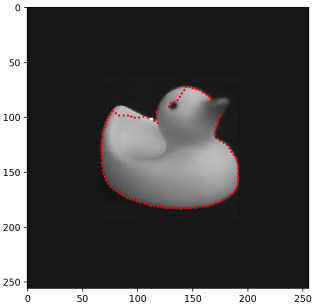

150

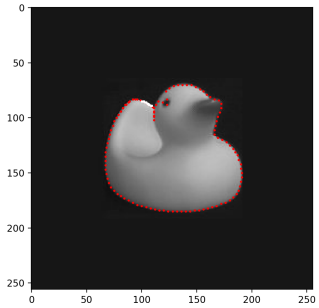

165

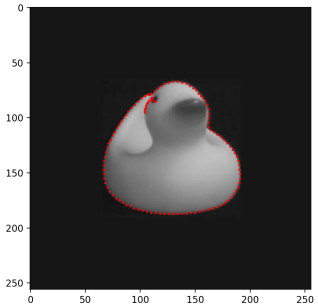

180

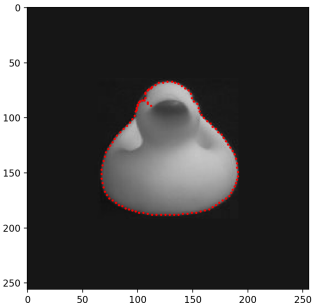

195

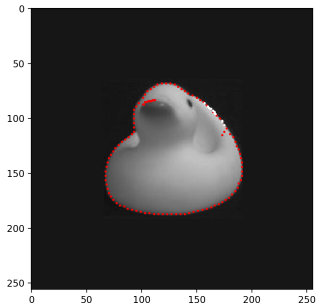

210


225


240


255


270


285

Parameters:

THRESHOLD1: 0.5

THRESHOLD2: 50

THRESHOLD3: 10%

SIGMA:1

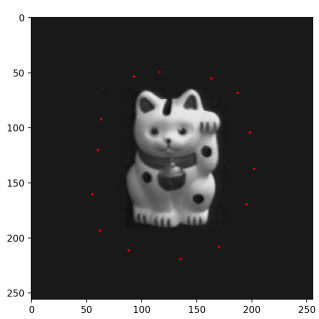
SIZE OF NEIGHBORHOOD: 25

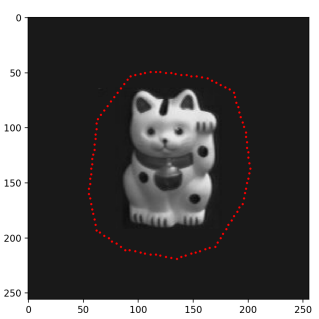
ALPHA: 1

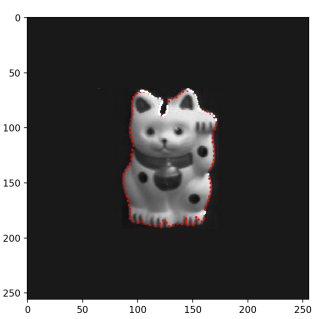
BETA: 1

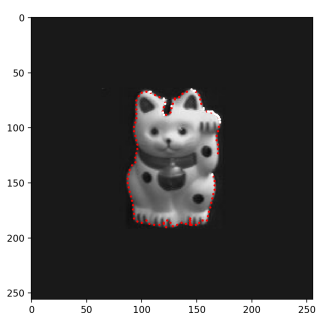
GAMMA: 1

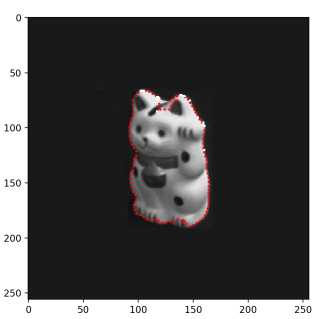
Sequence2 - cat

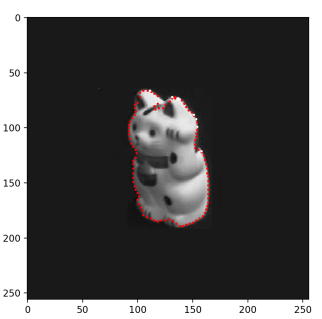


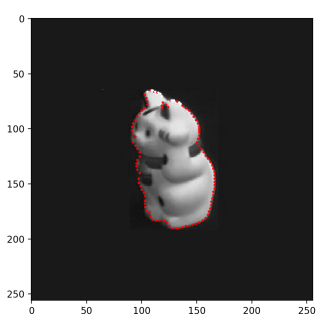


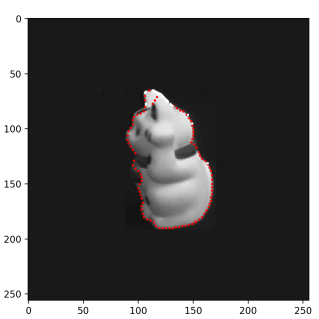


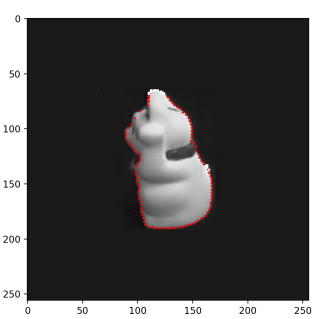


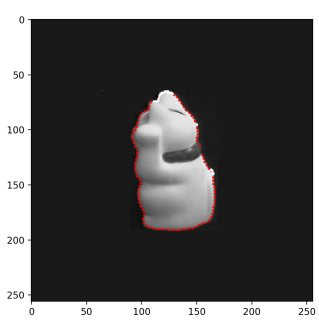


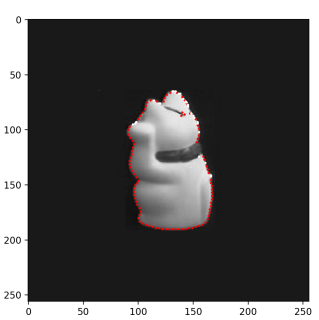


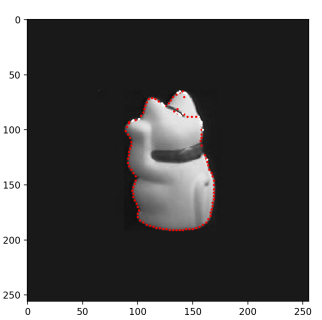


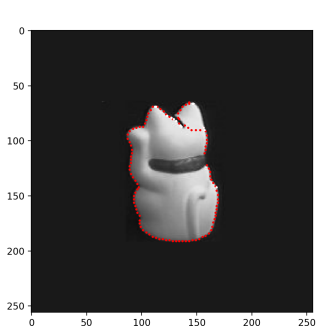


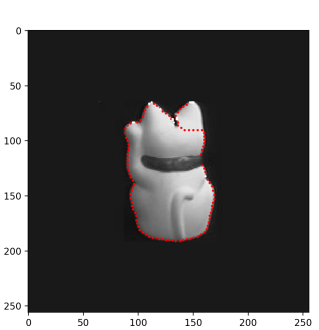












Parameters:

THRESHOLD1: 0.4
THRESHOLD2: 50
THRESHOLD3: 10%
SIGMA:1
SIZE OF NEIGHBORHOOD: 25
ALPHA: 1
BETA: 1
GAMMA: 1