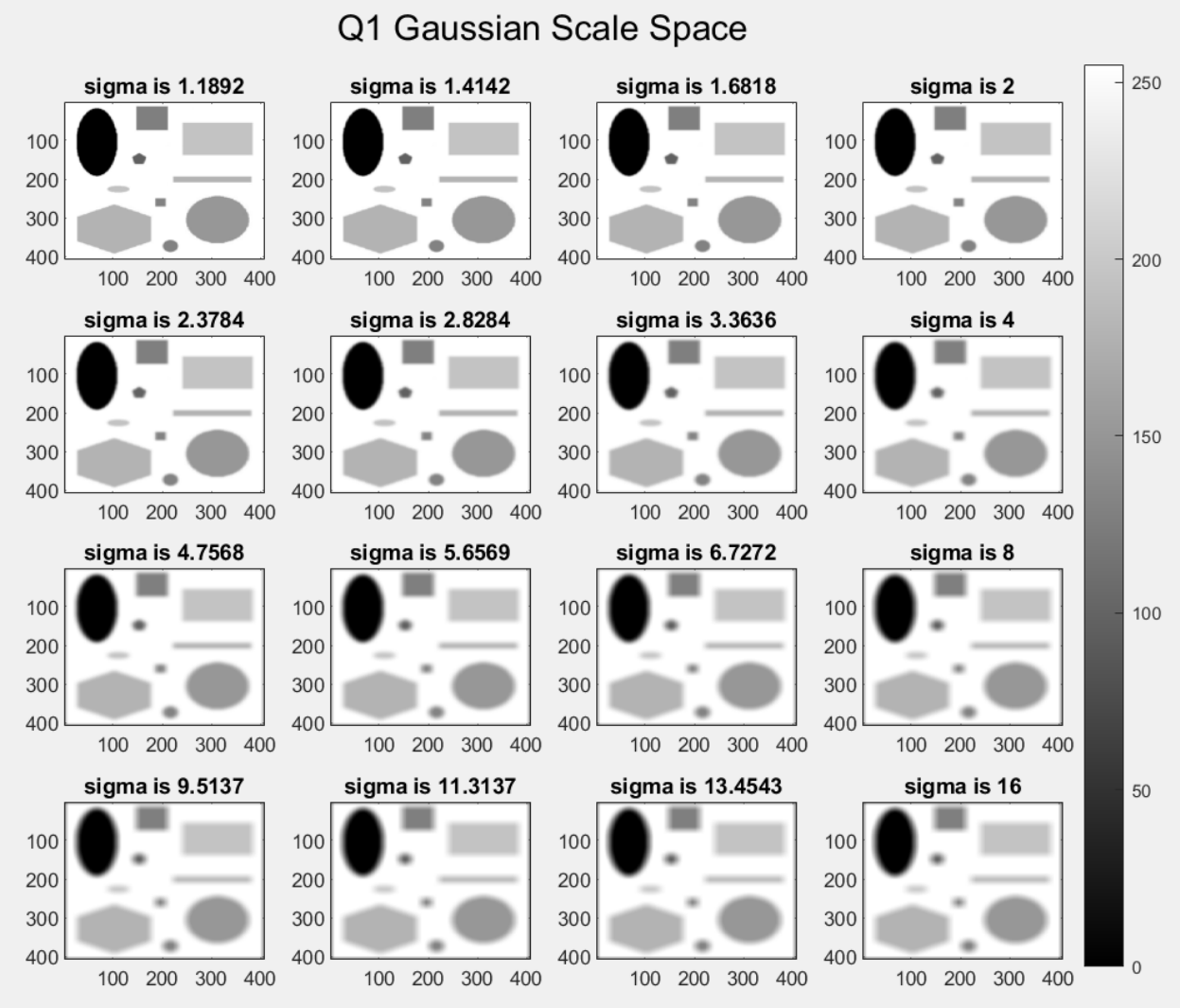
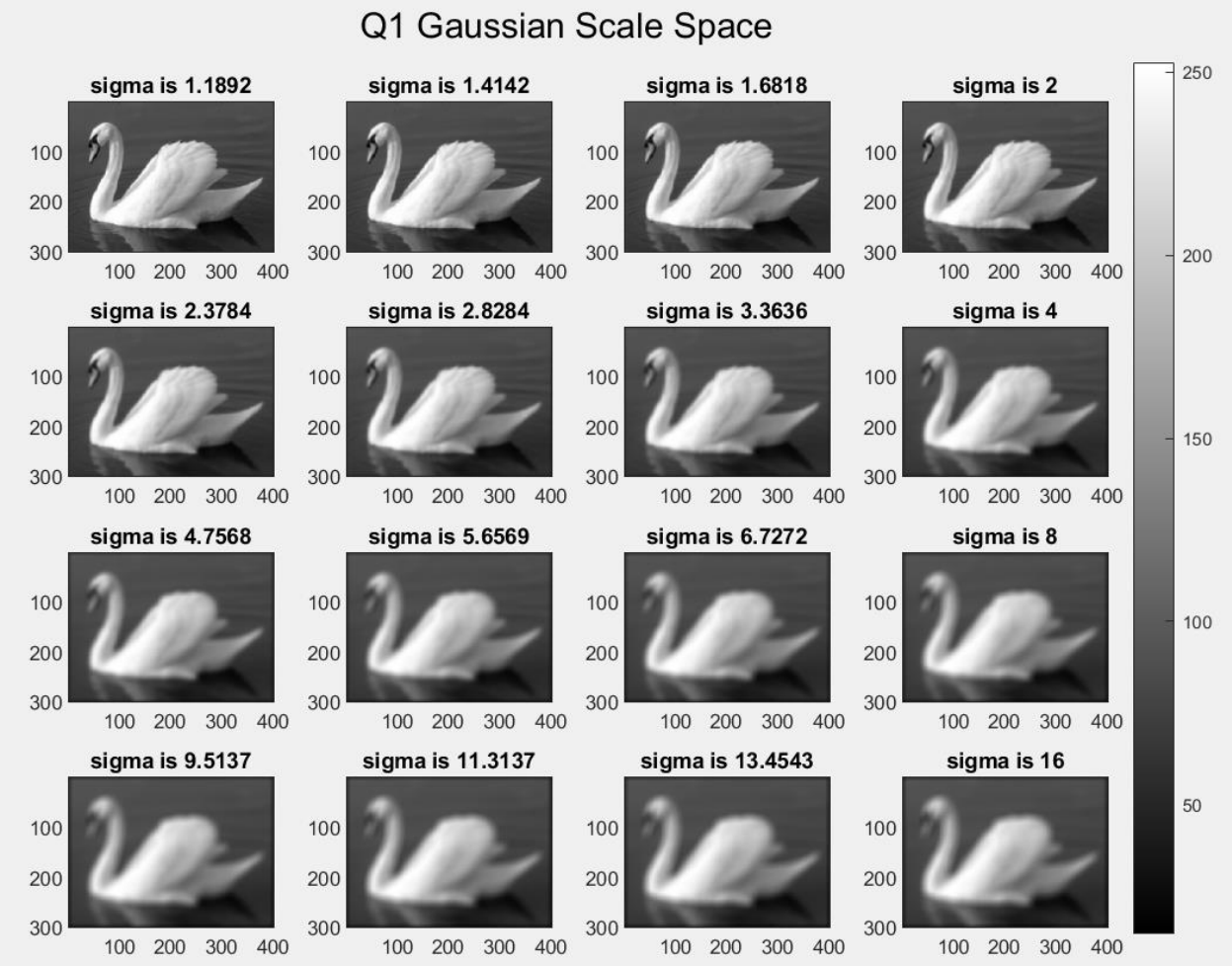
Q1

Slices 2 to 17:

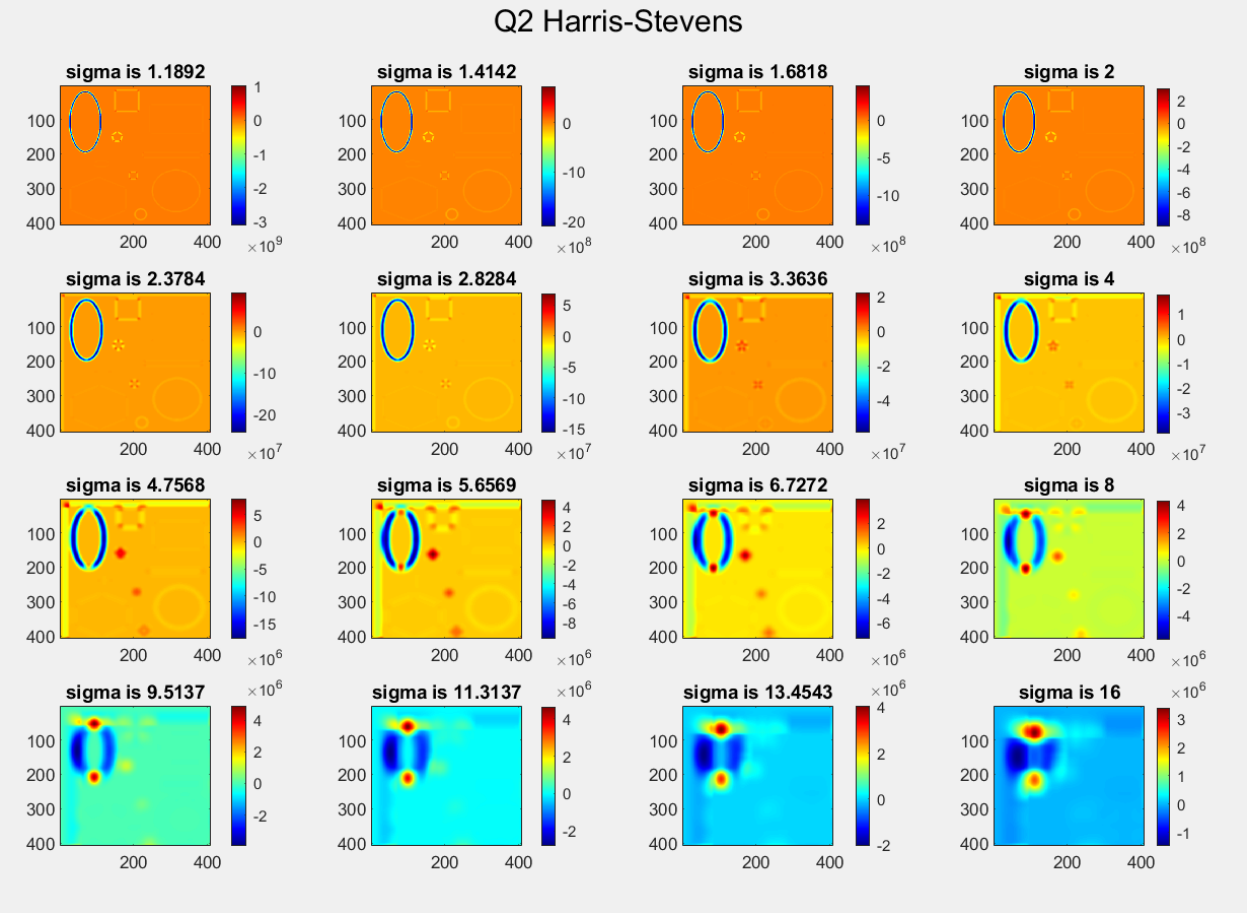


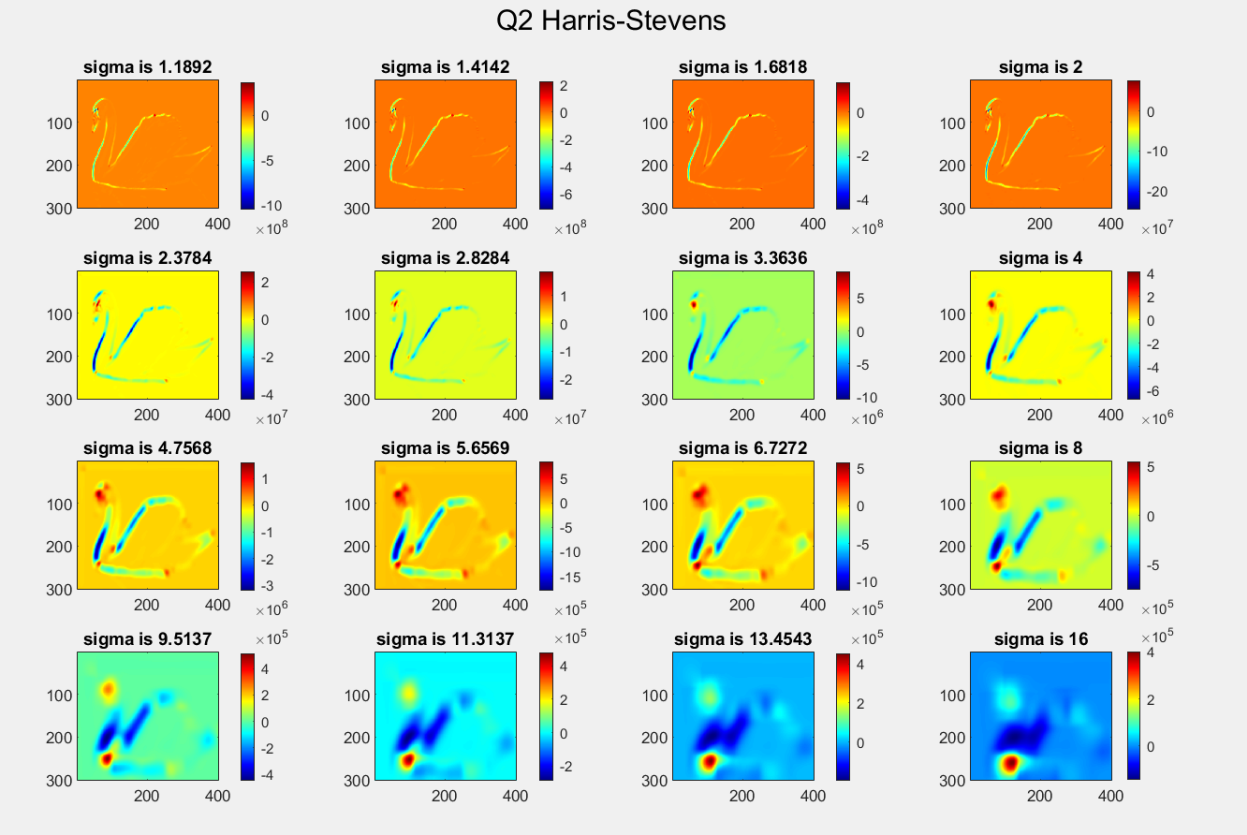


It gets blurrier as expected.

Q2

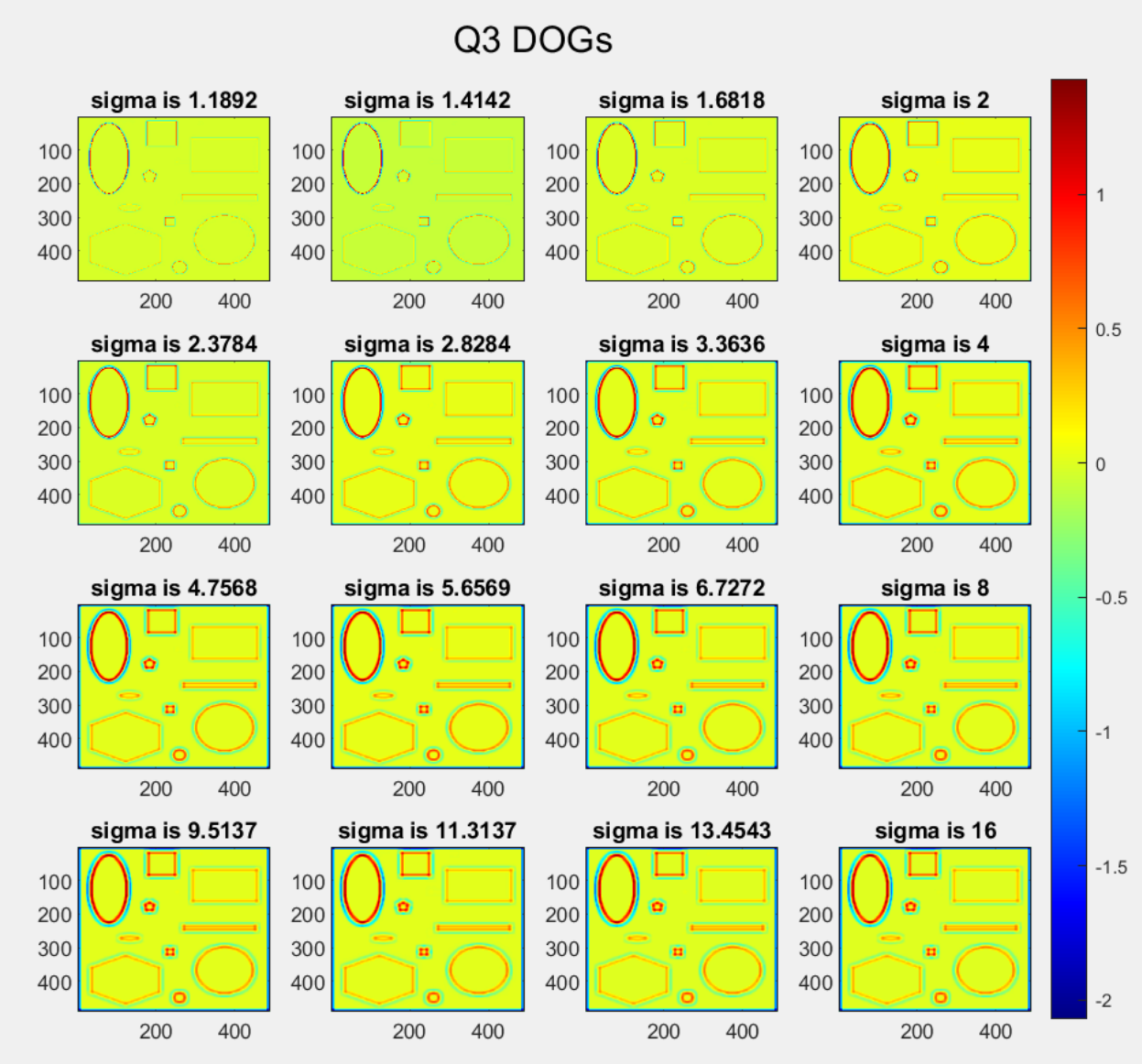
Result of Harris-Stevens is:

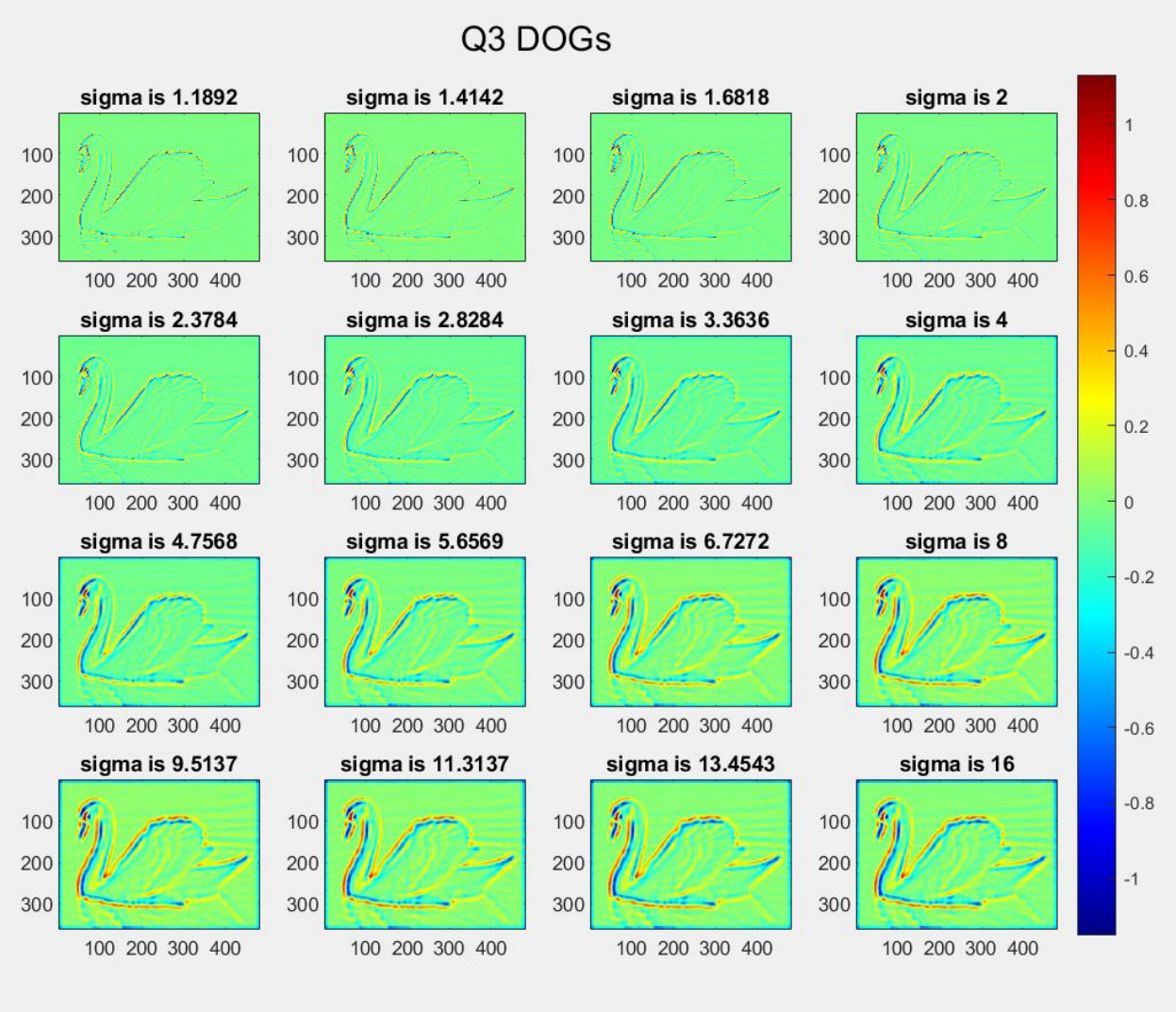




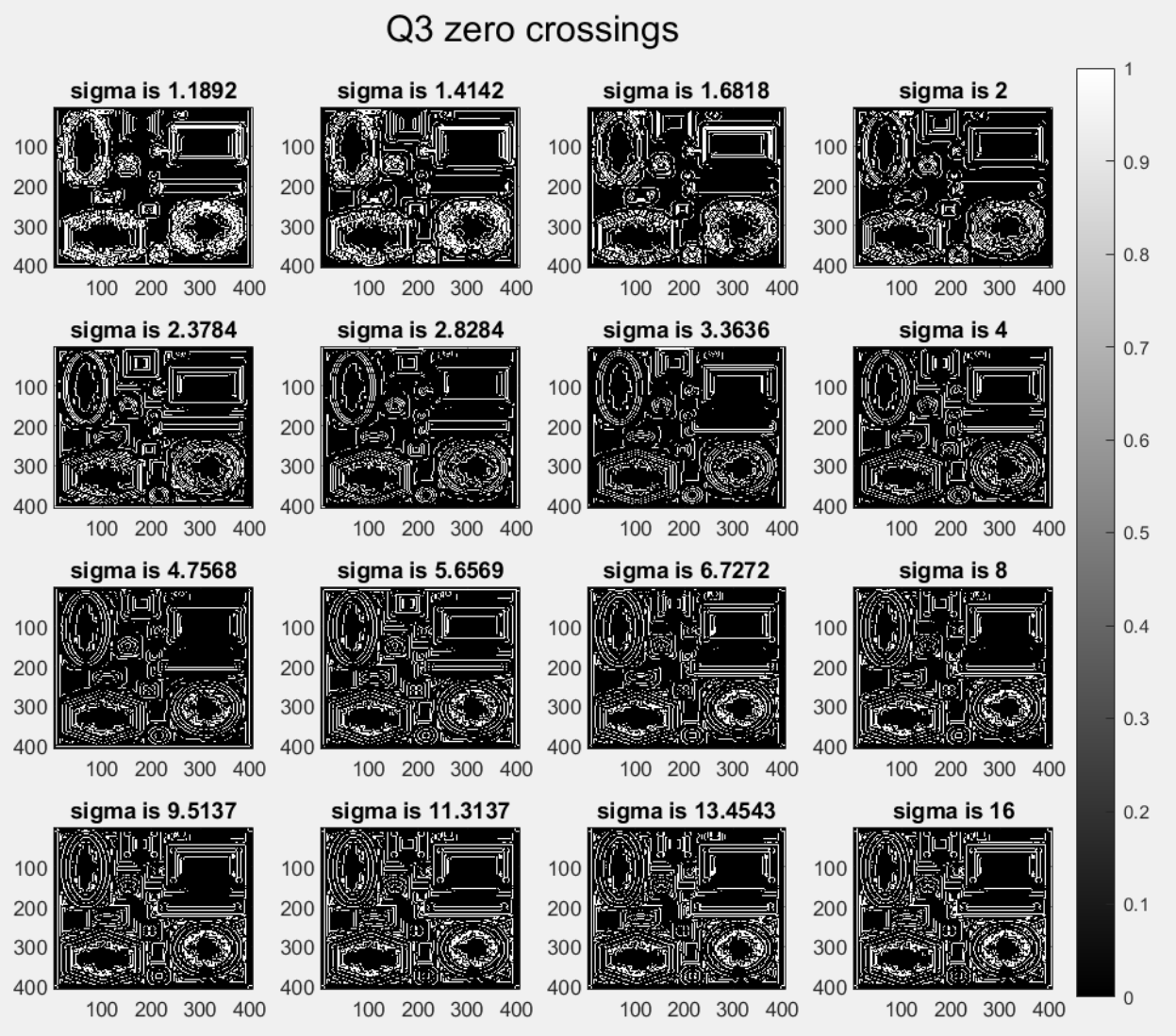
The operator does highlight locally distinctive points, but many of them are invisible because the picture is small, and the Harris-steven value may be high only at several pixels. For the visible ones, the detected key points are getting bigger and blurry as sigma increases, but some of them get suppressed as the scale increases; in the first picture, if we zoom in, we see that corners in this scale are detected, like the corner of the square; in the last picture, with sigma the largest, it successfully shows the locally distinctive points in its scale, like the bottom left part of swan. As scale increases, locally distinctive points are shown according to the scale. Other than that, it is hard to tell if the position of locally distinctive points detected have changed because each picture is small, and the size of the point increases quickly, but generally it tends to shift towards the higher intensity area as sigma increases.

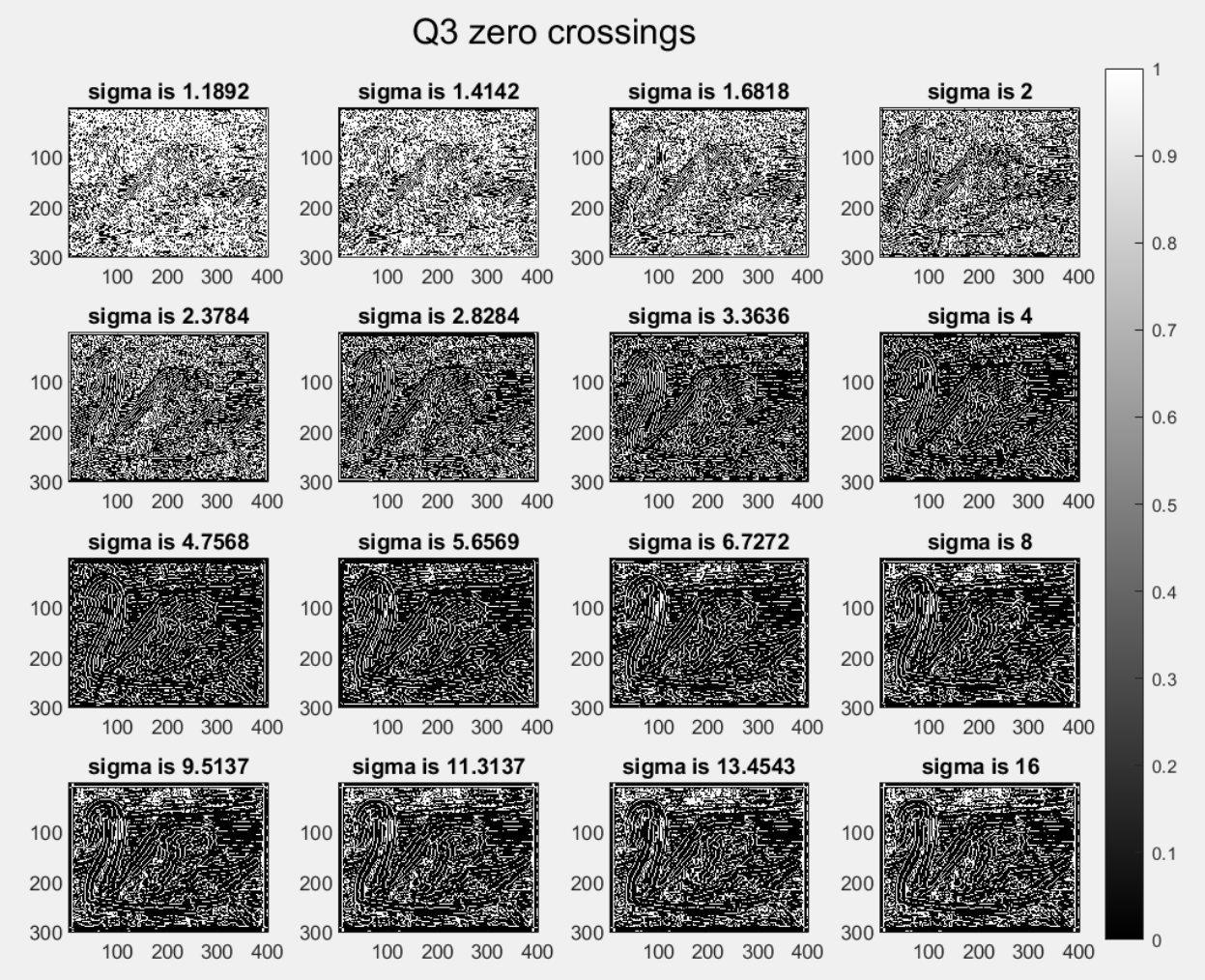
Q3


Q3 DOGs
sigma is 1.1892
sigma is 1.4142
sigma is 1.6818
sigma is 2
sigma is 2.3784
sigma is 2.8284
sigma is 3.3636
sigma is 4
sigma is 4.7568
sigma is 5.6569
sigma is 6.7272
sigma is 8
sigma is 9.5137
sigma is 11.3137
sigma is 13.4543
sigma is 16
100
200
300
400
200
400
1
0.5
0
-0.5
-1
-1.5
-2


Q3 DOGs
sigma is 1.1892
sigma is 1.4142
sigma is 1.6818
sigma is 2
sigma is 2.3784
sigma is 2.8284
sigma is 3.3636
sigma is 4
sigma is 4.7568
sigma is 5.6569
sigma is 6.7272
sigma is 8
sigma is 9.5137
sigma is 11.3137
sigma is 13.4543
sigma is 16
100 200 300 400
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1

Zero crossings

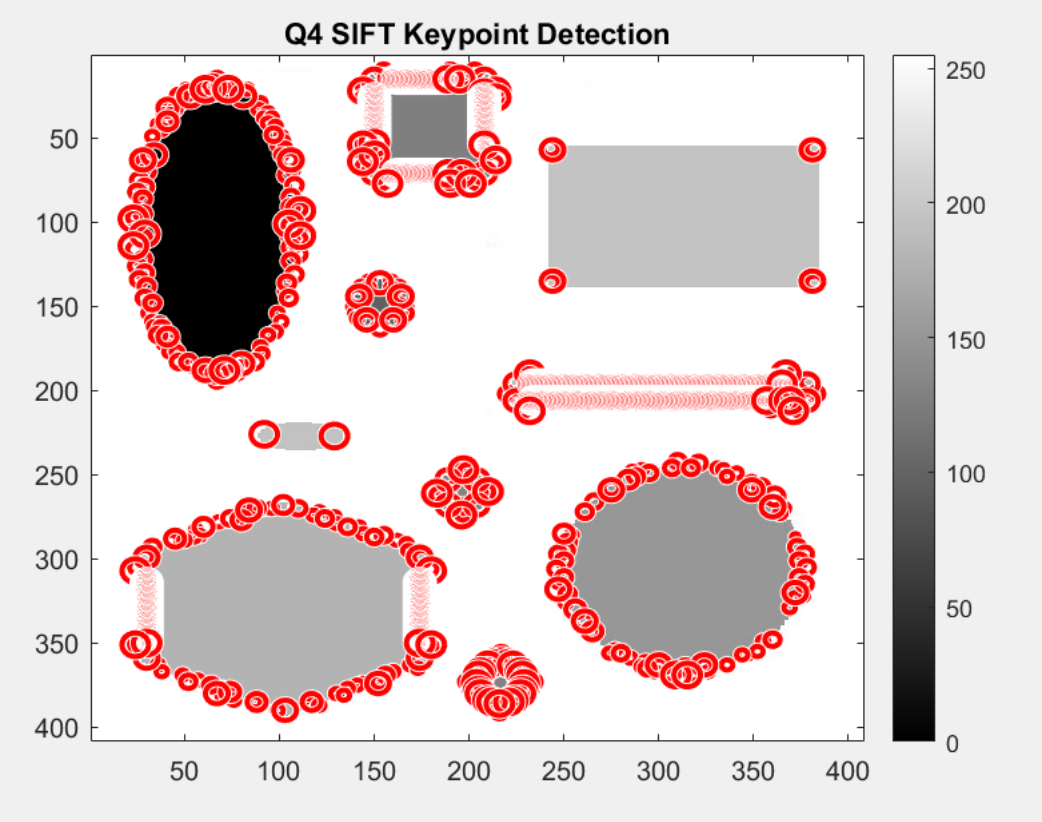


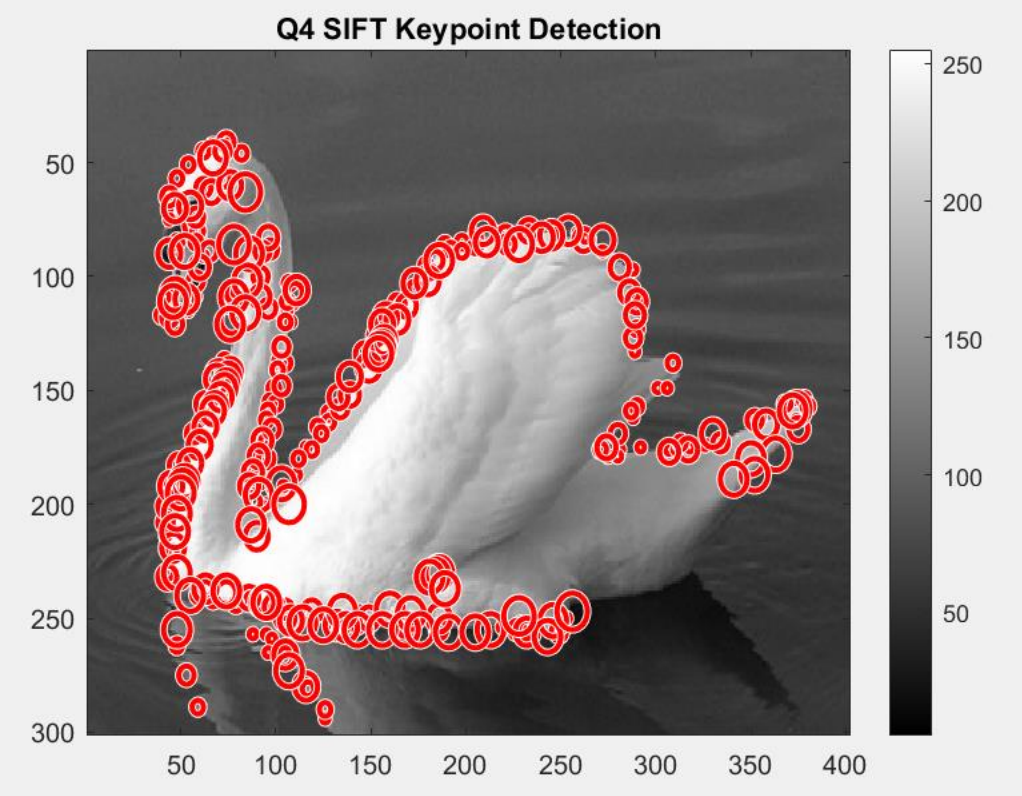


We can see from the first subplot that the zero crossings show the edges but are blurred, and the position of zero crossings does not change from the edge itself generally. As sigma increases, there are more smoothing going on and the edges get more influences from further neighbors, and some small edges are suppressed.

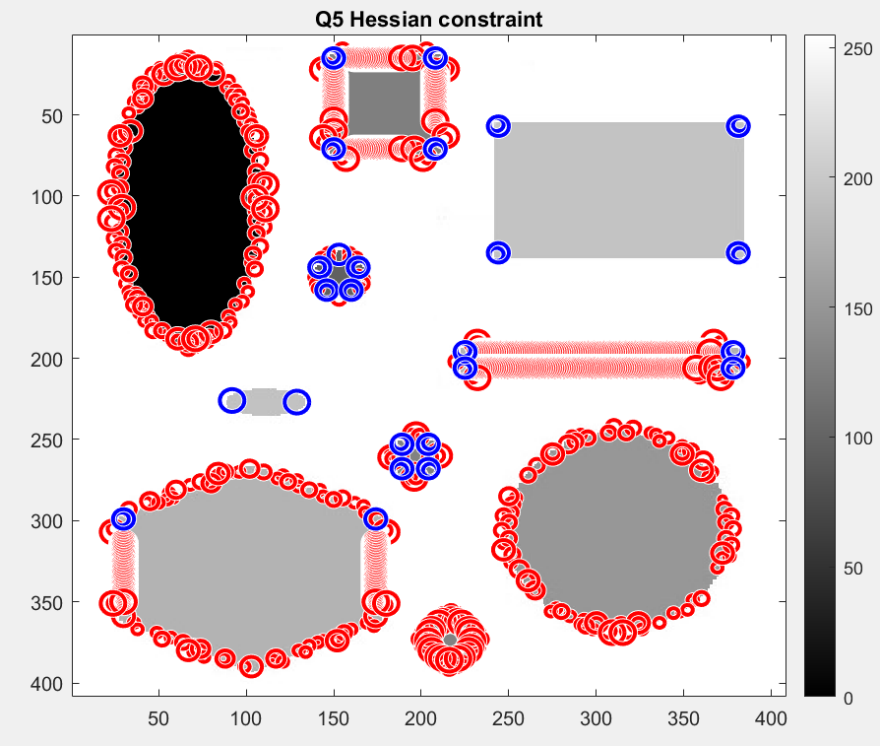
In the synthetized image we can say that there is no noise, so more sigma makes the image further from the original image, but as for the natural image, the scene is depicted better because there are many tiny changes in intensity in natural environment, so the zero crossing in first subplot is awful to understand the scene; as image get smoothed more, the scene is clearer, because some “noises” are suppressed.

Q4

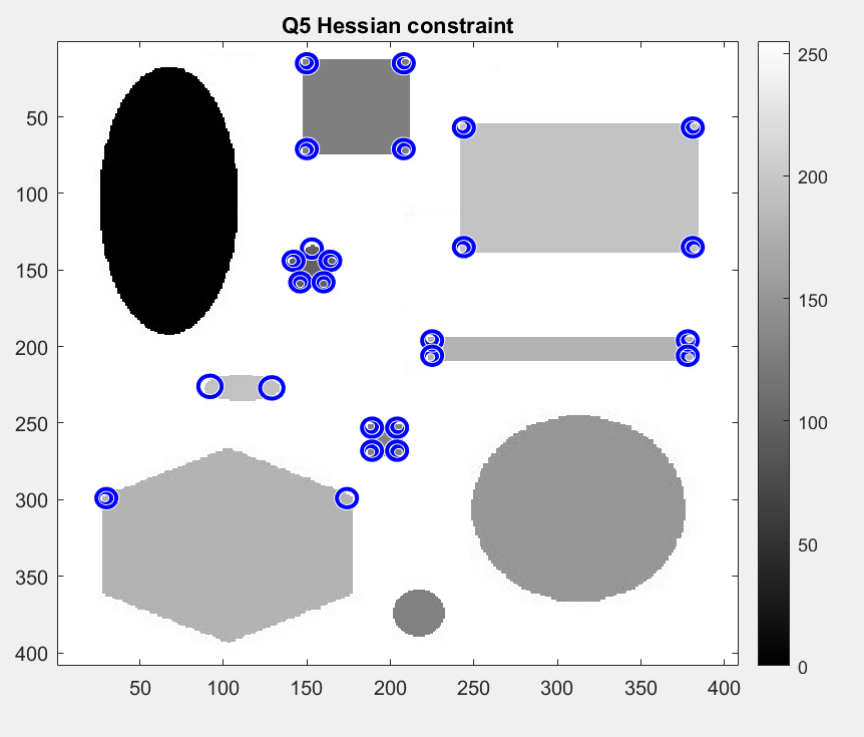

Q4 SIFT Keypoint Detection
50
100
150
200
250
300
350
400
250
200
150
100
50
0

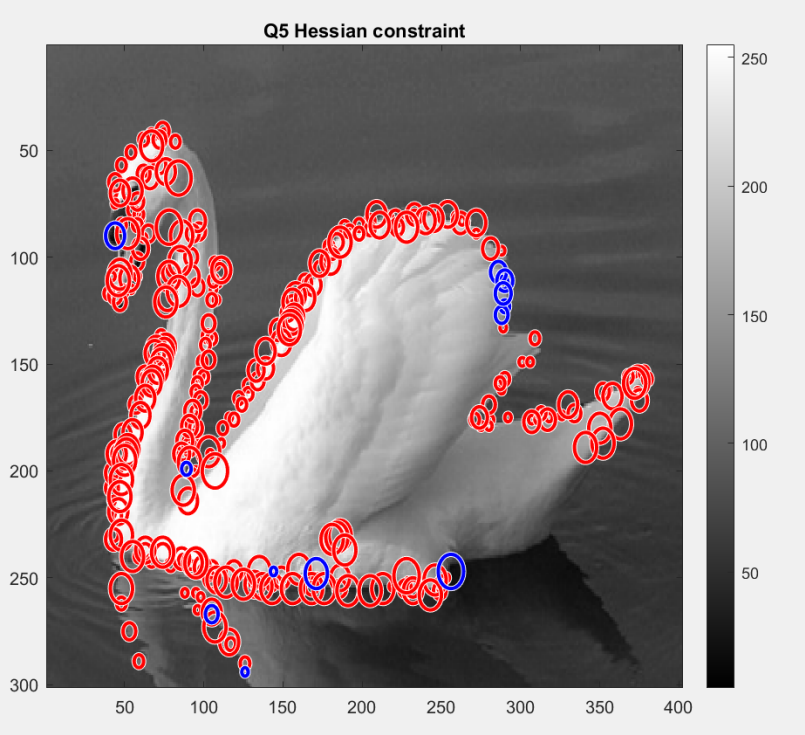

Q4 SIFT Keypoint Detection
50
100
150
200
250
300
350
400
250
200
150
100
50

Q5

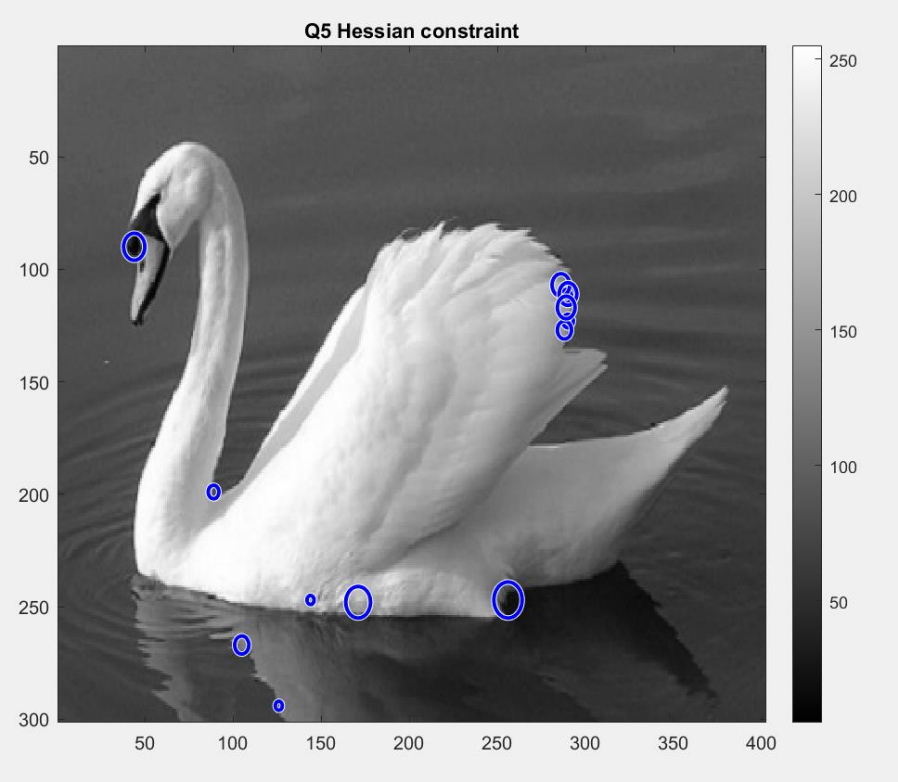


In case you need only blue points:





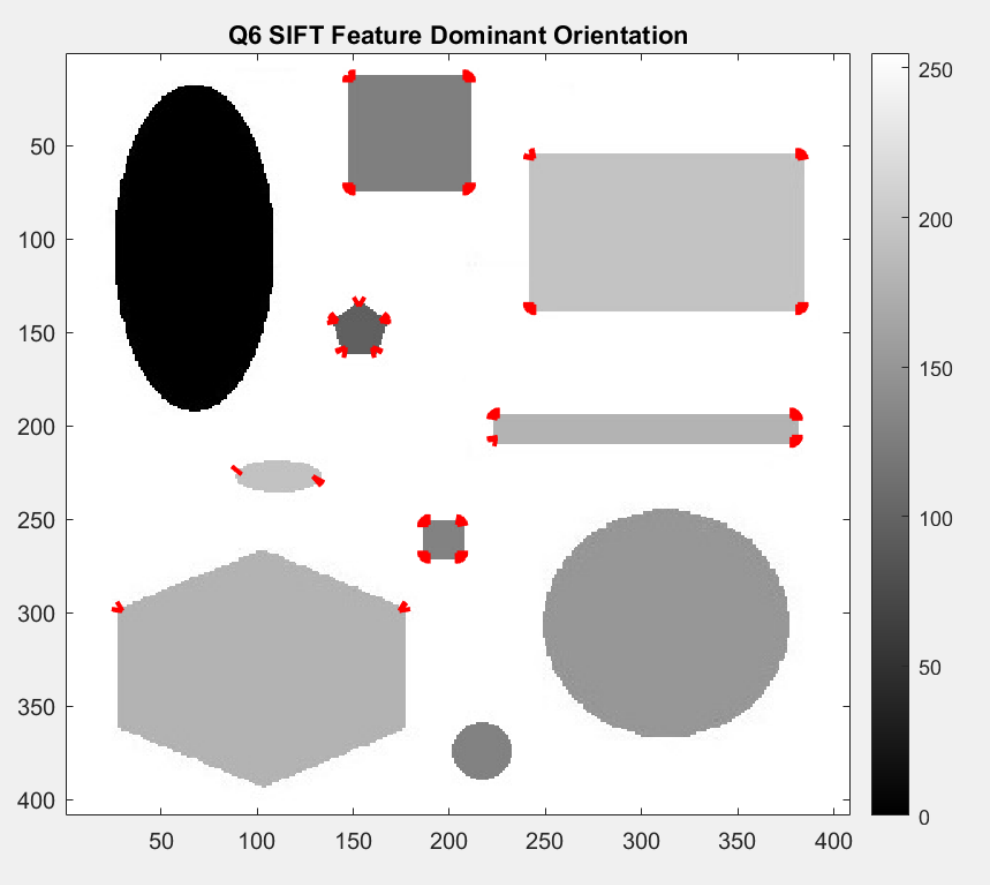
In case you only need the blue ones:



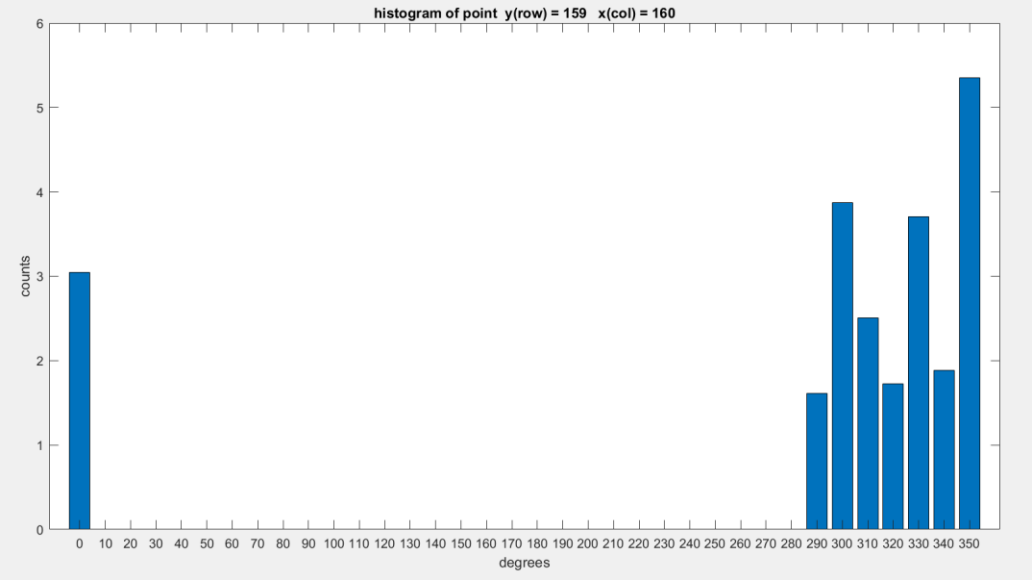
Approximation for Hessian:

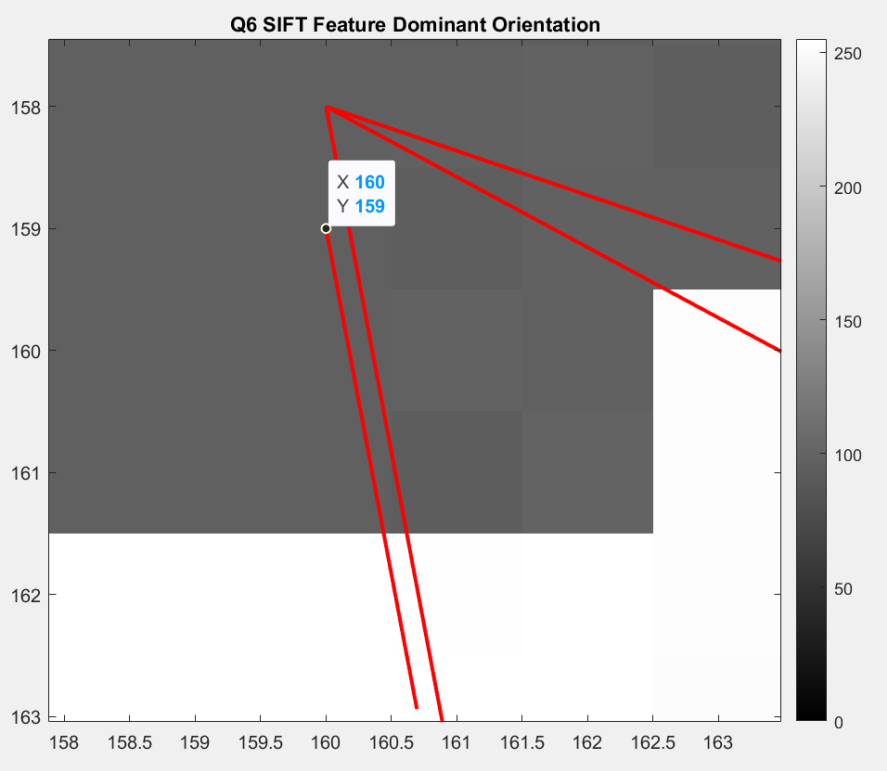
First, we get the intensities of the image in the neighborhood of the keypoint (x0,y0) with size 5. (so the keypoint is the (3,3)th element, in the center of the neighborhood). And then, we filter the neighborhood intensities by x derivative filter and y derivative filter with valid(x derivative filter is [1 0 -1; 1 0 -1; 1 0 -1]/6, and transpose it for y derivative), so we get two 3*3 derivative matrix for x and y; similarly, we calculate the second derivative by first derivative in the same way(filter the first derivative by the derivative filter): we get second derivative of x by taking derivative of first derivative of x(xxdr); we get derivative of x and y by taking derivative with respect to y of first derivative of x(xydr=yxdr); we get second derivative of y by taking derivative of first derivative of y(yydr). With these, hessian matrix is finished by [xxdr, xydr; xydr, yydr].

Q6



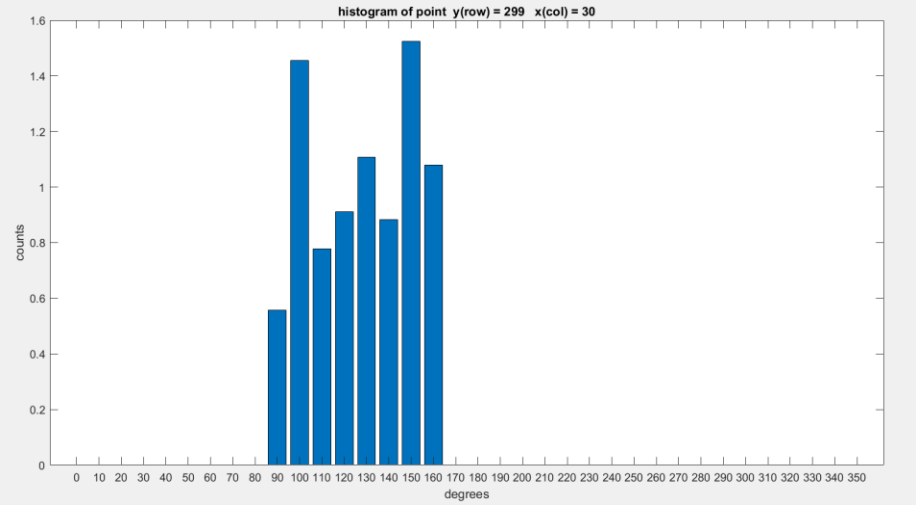
Histograms:



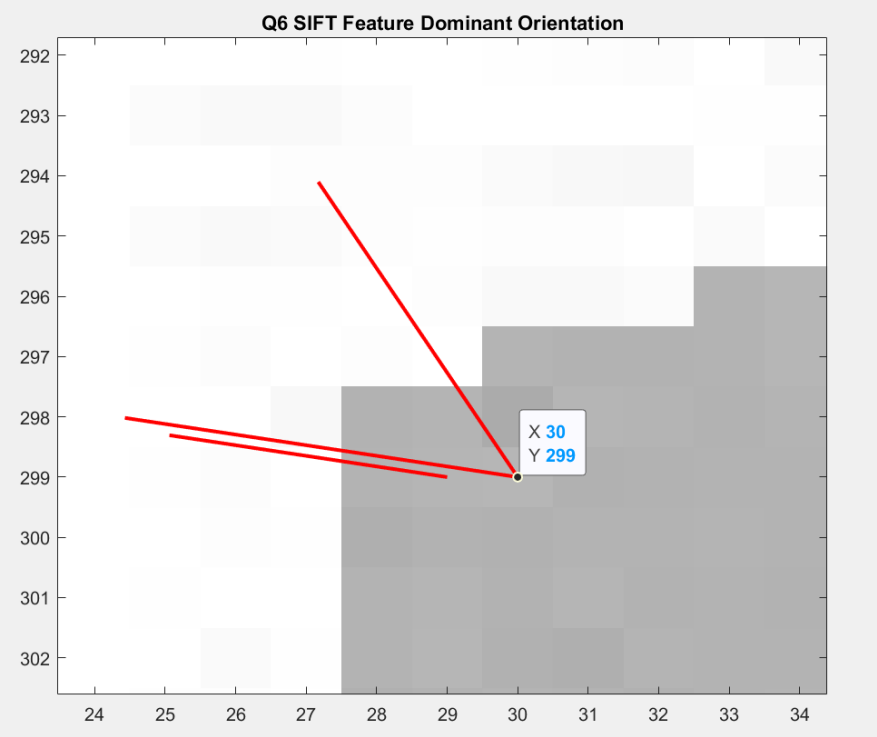


The angle is set to be 0 downward(positive y direction in matlab), and go clockwise. As a result, we see that the dominant orientation is 350 degree shown.

Another point:



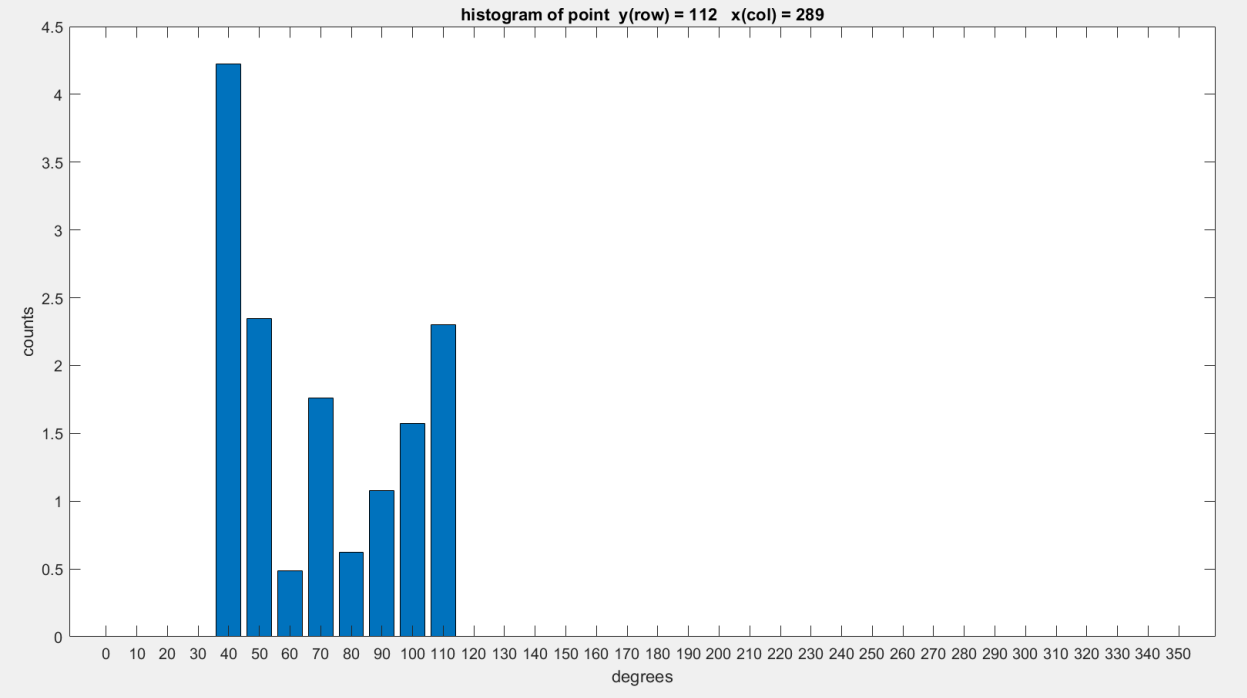
The histogram suggests that there are two dominant orientations, degree=100 and degree=160.

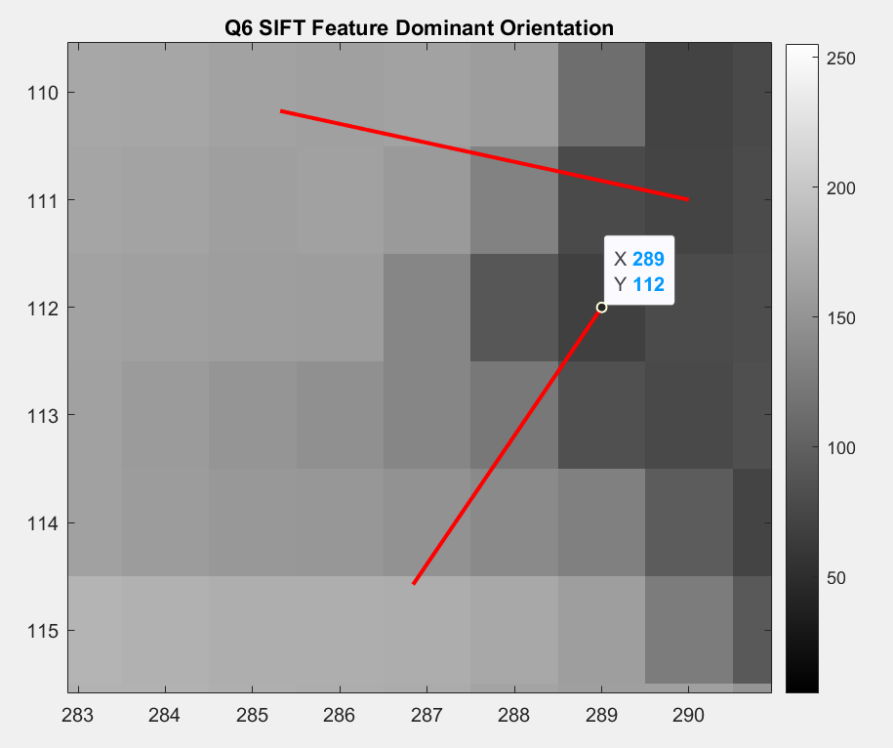


Similarly, 0 degree is downward, and angles go clockwise, so we see that the dominant orientation shown are indeed 100 degree and 160 degree.

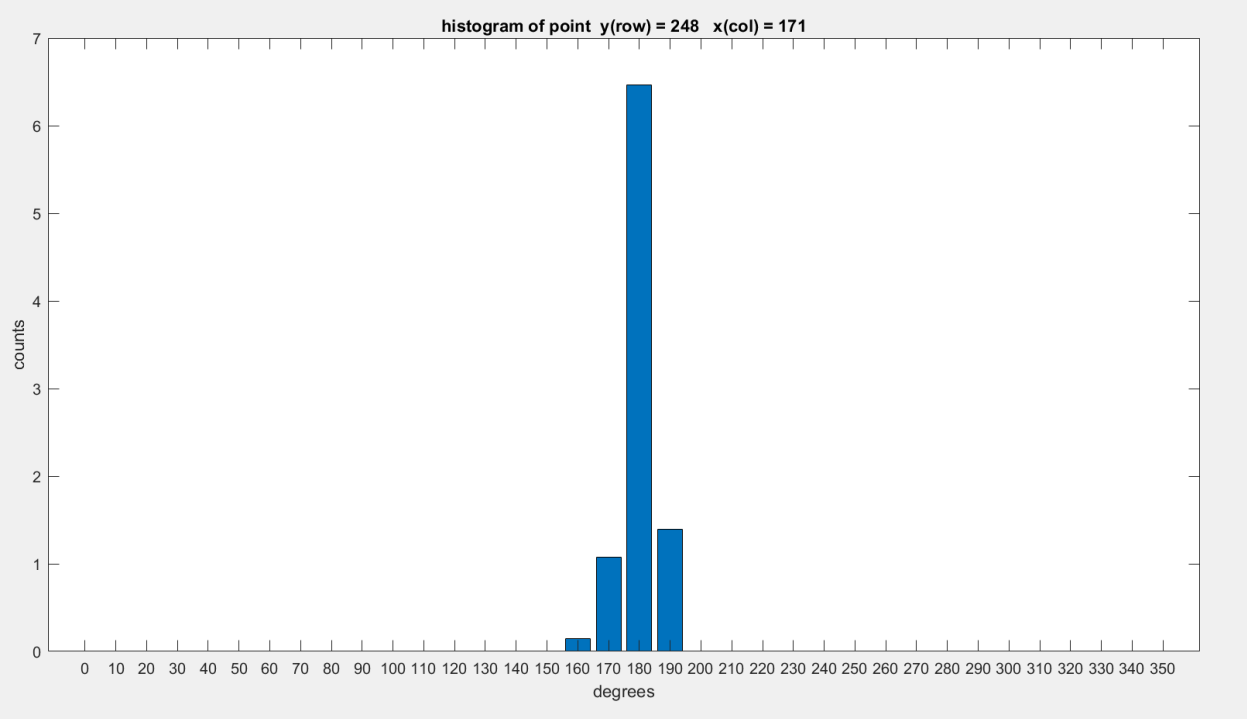
Natural scene:

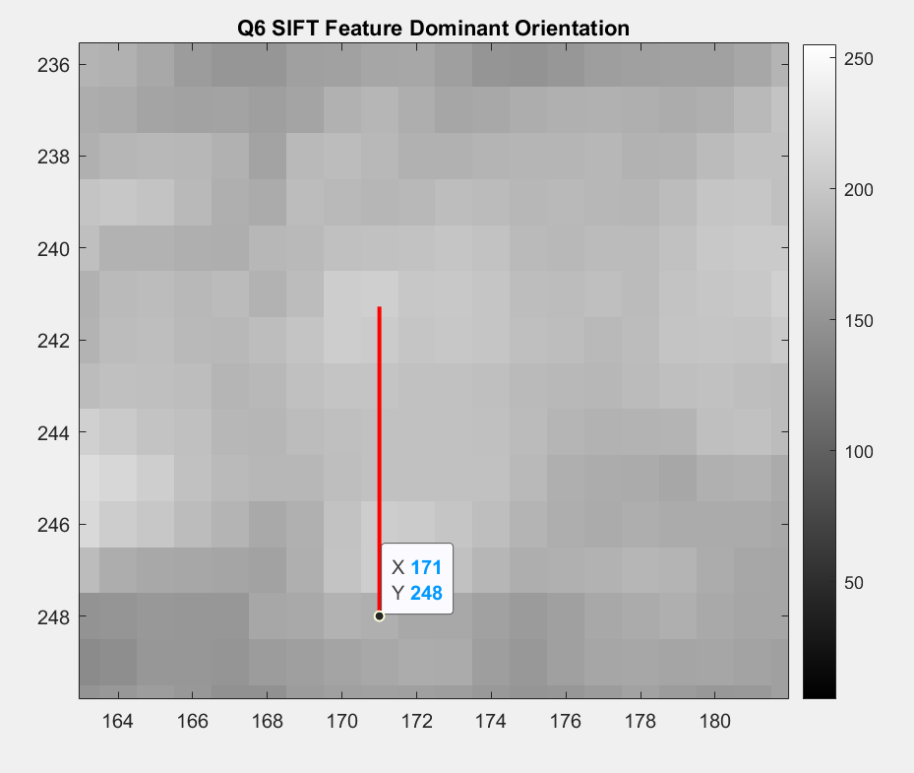
Histograms:





The histogram suggests there is 1 dominant orientation, 40 degree. We see that with the standard matlab orientation, the direction is correct.

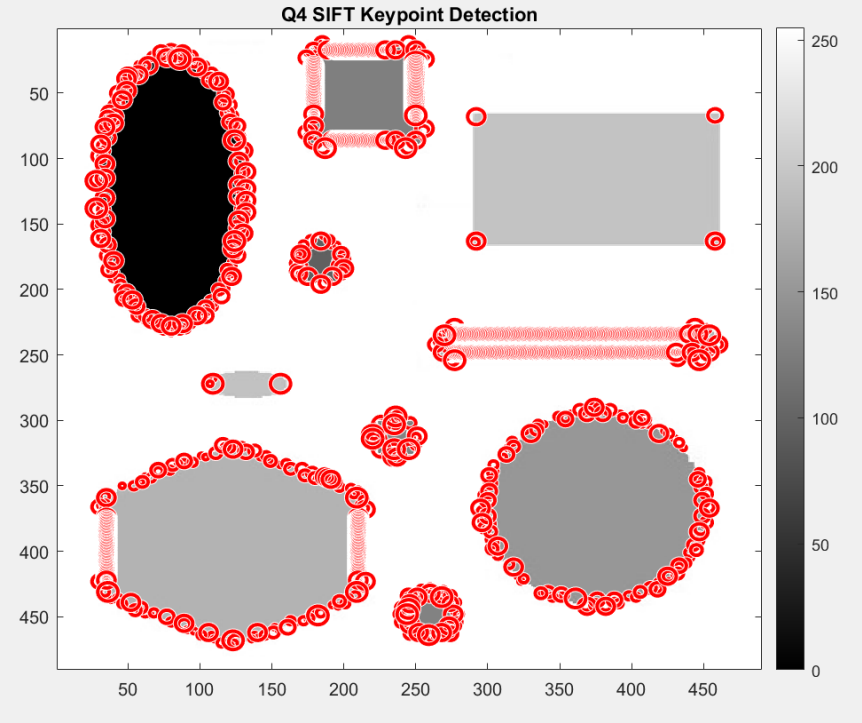


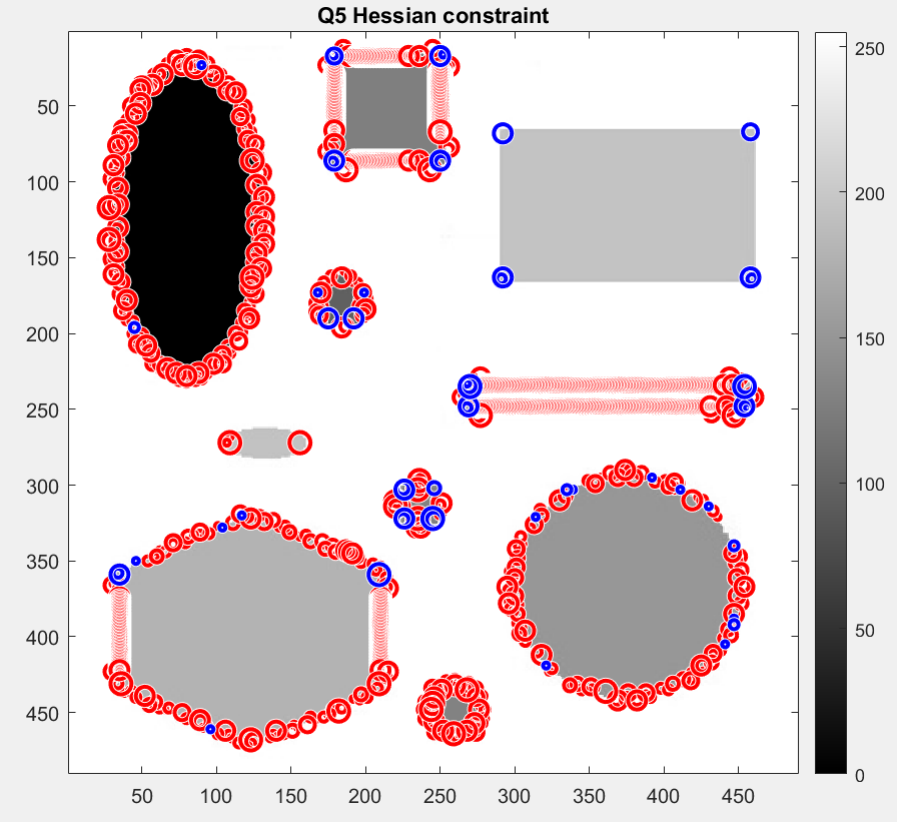


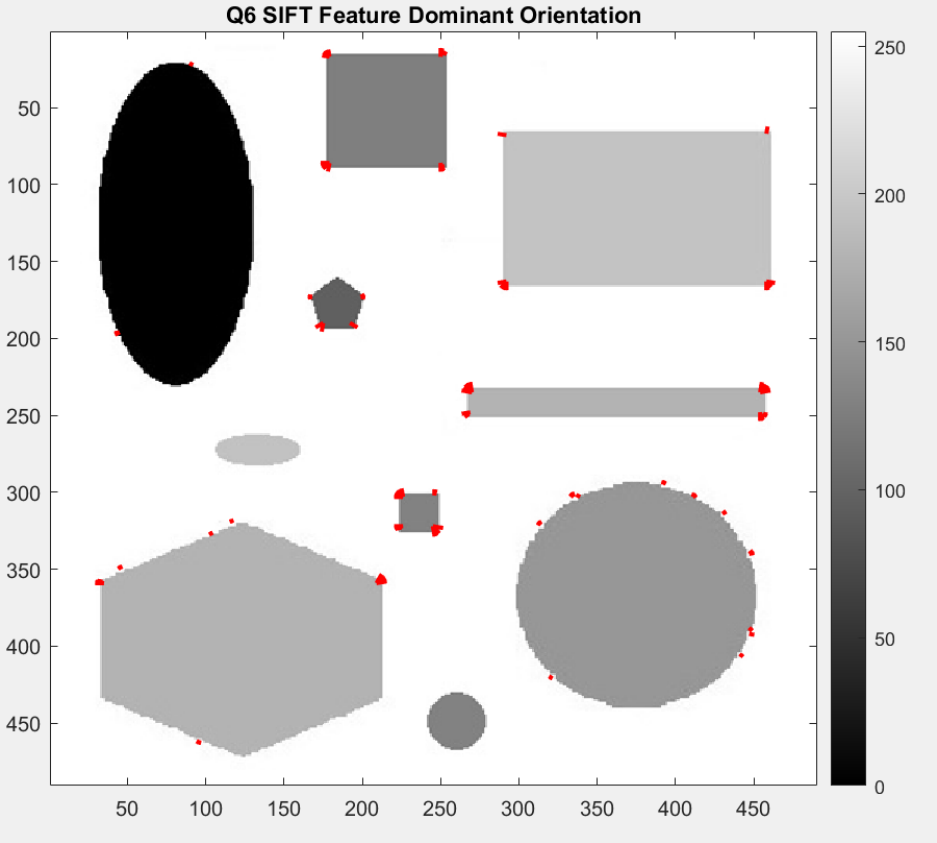
The histogram suggests there is 1 dominant orientation, 180 degree. We see that with the standard matlab orientation, the direction is correct.

Q7

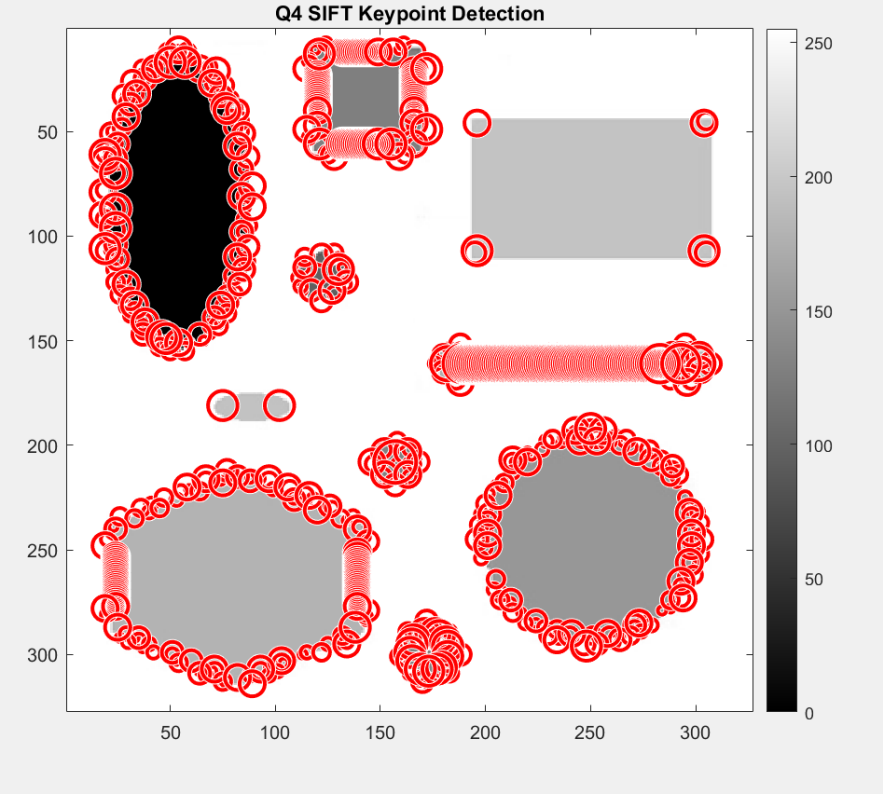
Rescale to 1.2:

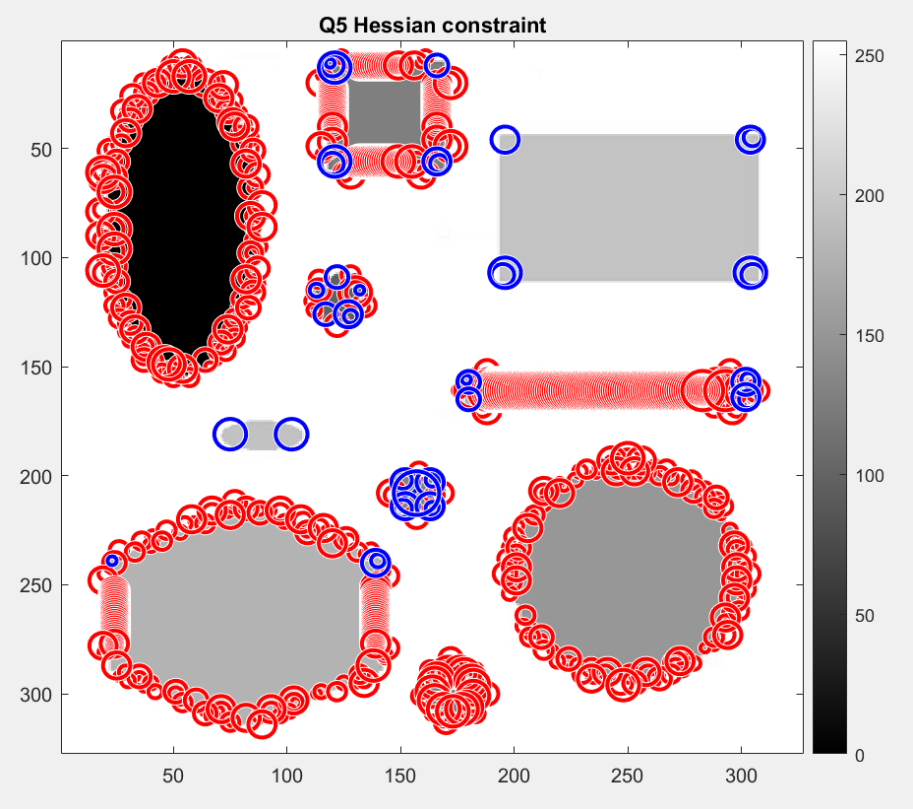


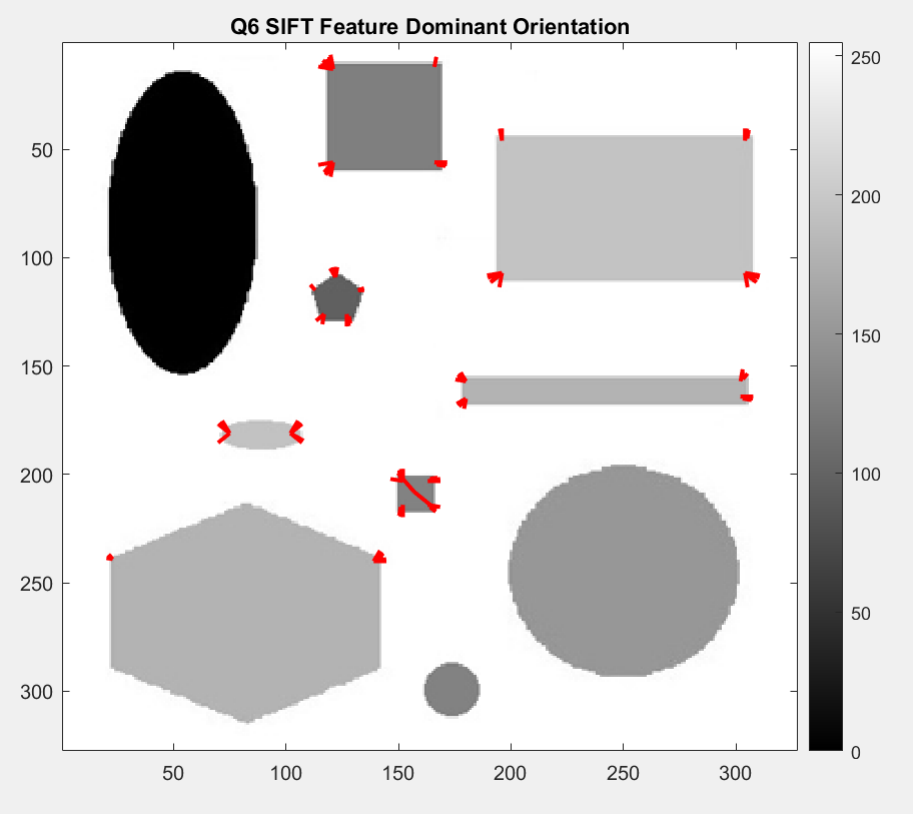



Q6 SIFT Feature Dominant Orientation
50
100
150
200
250
300
350
400
450
250
200
150
100
50
0

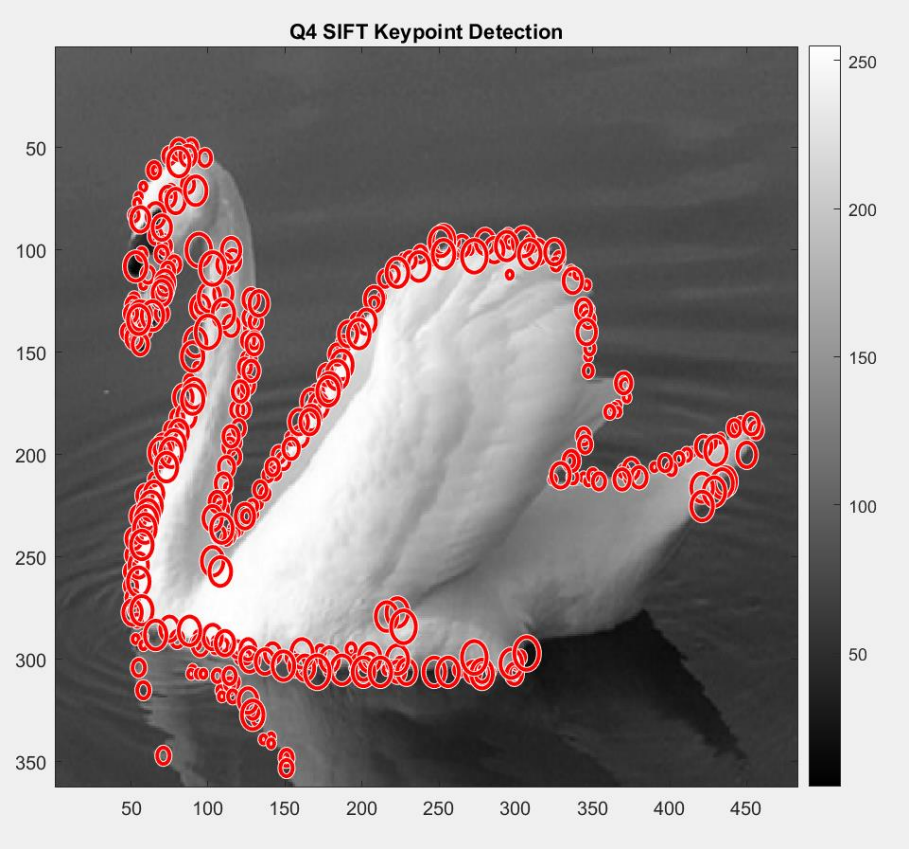
Resize to 0.8

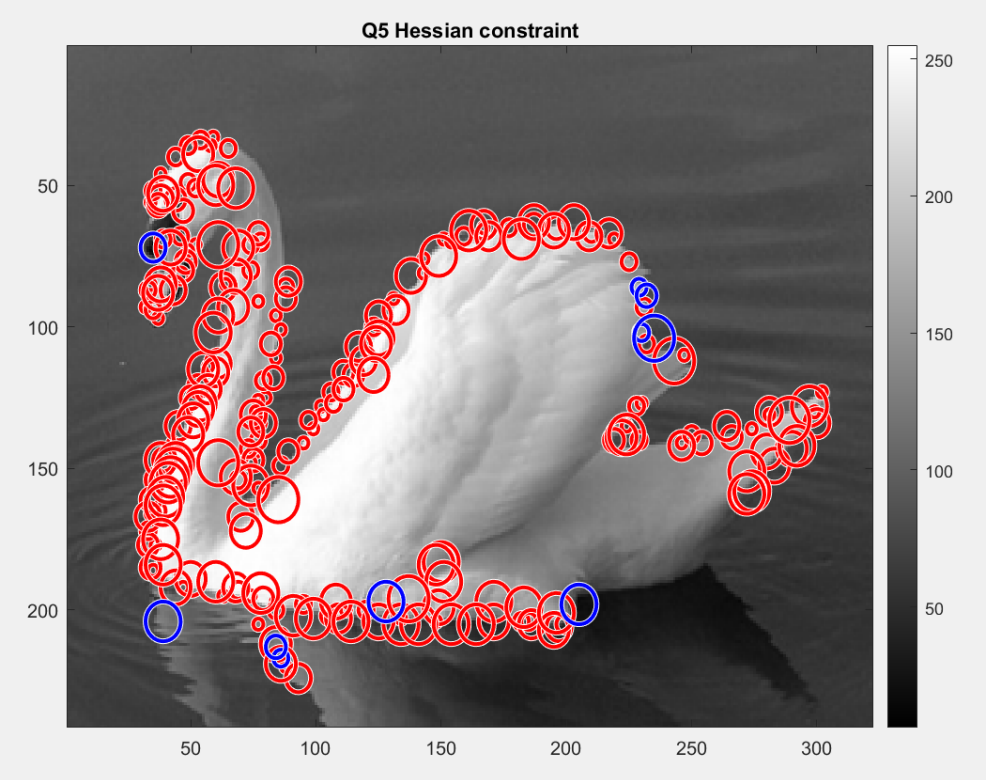

Q4 SIFT Keypoint Detection
50
100
150
200
250
300
250
200
150
100
50
0

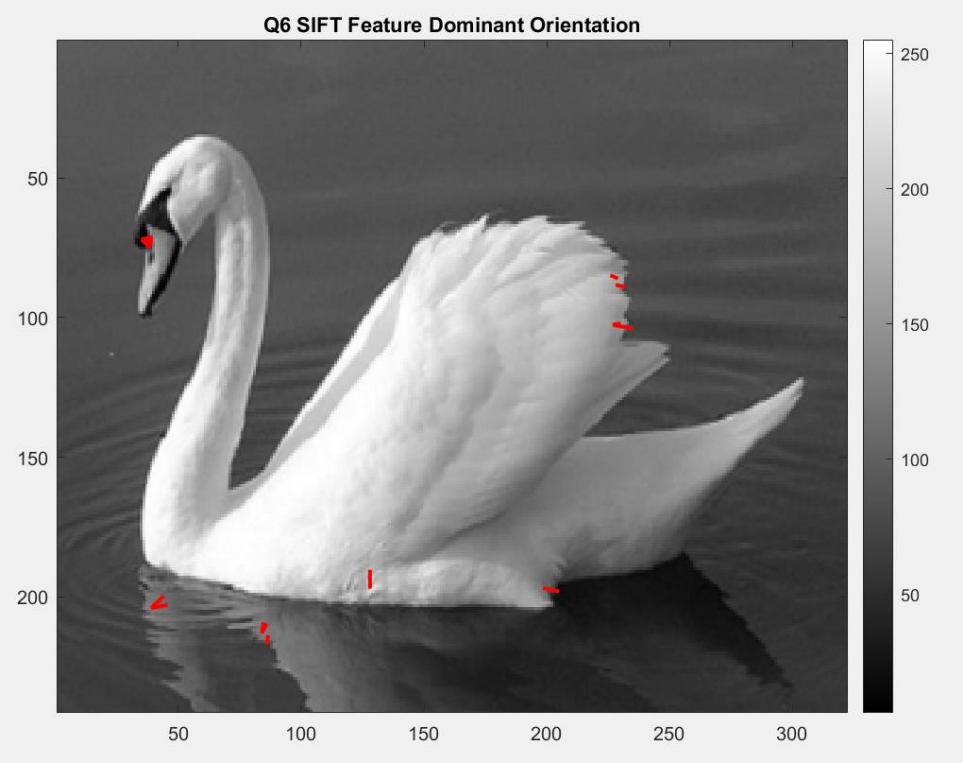

Q5 Hessian constraint
50
100
150
200
250
300
250
200
150
100
50
0


Q6 SIFT Feature Dominant Orientation
50
100
150
200
250
300
250
200
150
100
50
0

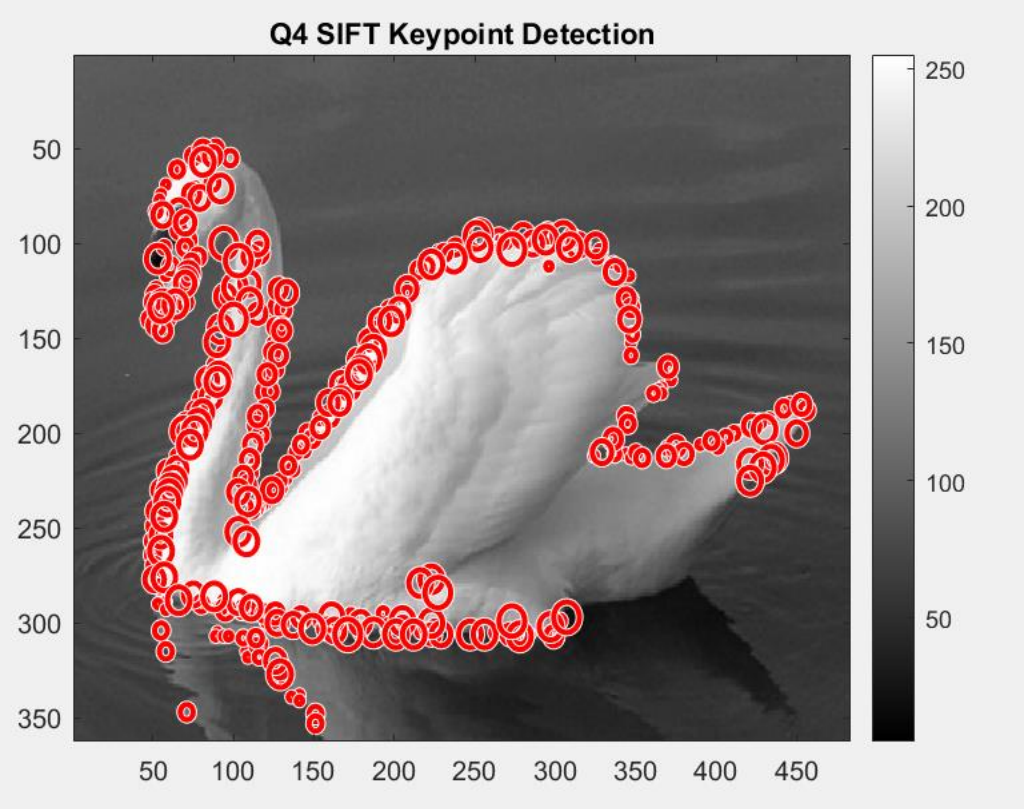
Nature picture: rescale to 0.8

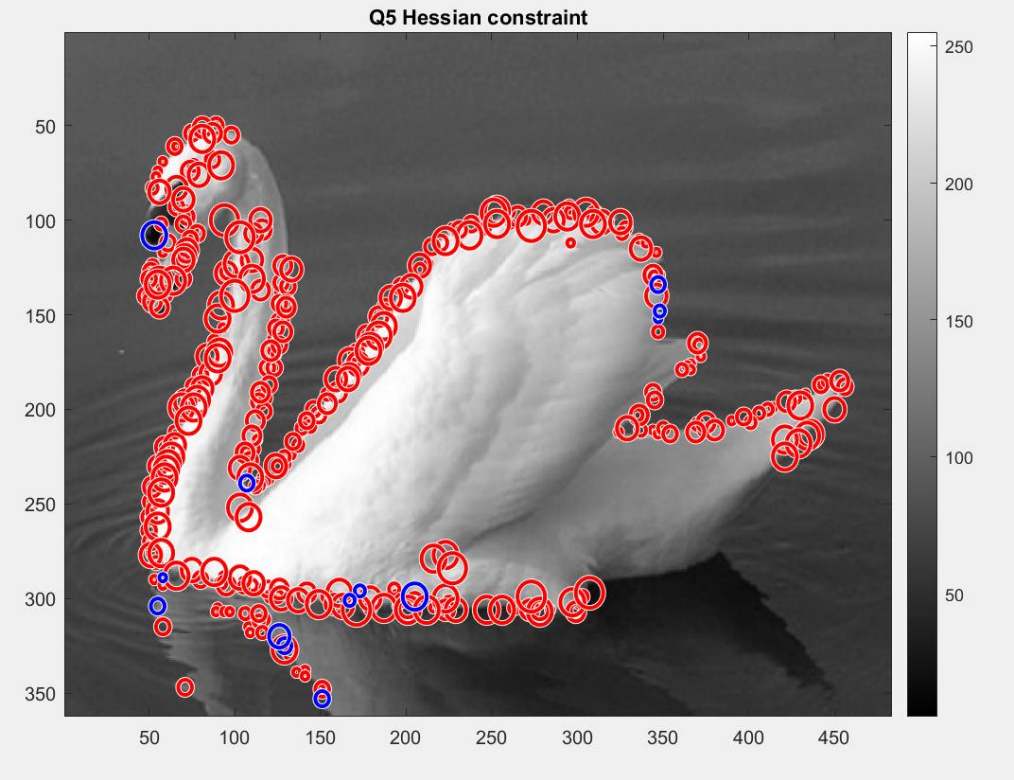


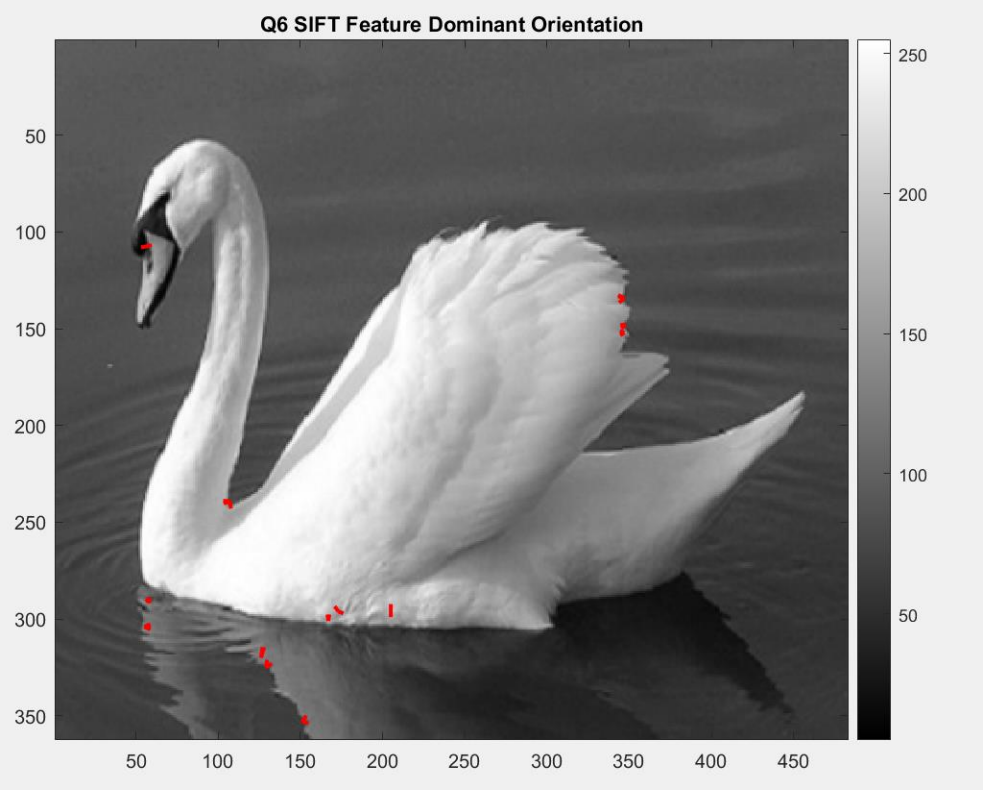



Q6 SIFT Feature Dominant Orientation
50
100
150
200
50
100
150
200
250
300
250
200
150
100
50

Rescale to 1.2







The keypoints are generally scale invariant; there are some key points emerging in scaling up and down, but that may due to the choice of threshold. For the keypoints that are found in all scales, their position and orientation does not change, so that the newly rescaled image correspond in the expected way with the key points in the original image. (it is difficult to check if scale is the same, because the size of picture changes, so circles in the same size will appear smaller in scaled up pictures, but in general, scale is invariant to imresize)