Barış Can 16/12/2019

21501886

HW3

I started the code by getting all images with a for loop. I used slicomex function since it looked smoother and I initially used 500 superpixels, but after some trials, I decided to use 500 superpixels since it made the end result better. After that, I went on with getting RGB values of all pixels in the superpixels. After getting values, I calculated their mean value dynamically. For this part, I only used the image as my parameter and my output was a cell that includes RGB mean values of superpixels. It has the number of superpixels as its rows and 1 as its column. In each column, there are RGB mean values. Then, I took the transpose of the cell so that I can normalize mean values easily. After taking the transpose, my cell had 3 rows (RGB) and the number of superpixels as the columns. I also convert my cells to an array and normalize the values. After normalizing the values, I calculated the colour distance for two superpixels as $D = \sqrt{((R1 - R2)^2 + (G1 - G2)^2 + (B1 - B2)^2}$. I put my threshold as 0.11 after doing many trials. It was not optimized for all pictures but this was the one of best I can get. The output was a label array that I later create a mask with it.

After calculating the colour labels, I went on with Gabor filters. I converted my image to grayscale and create a cell with four rows and four columns. I used 2, 3.3, 4.6 and 6 as my scales. I first used 2, 4, 6 and 8 but then I realized scale 8 was very big and it blocks nearly all details, so I changed my scales. Also, I used 45, 90, 135 and 180 as my orientation. I applied `imgaborfilt` filter and get the magnitudes. In figure 1, you can see all Gabor filters for all images.

Figure 1: Bus Image for Scale 2 and Orientations 45-90-135-180

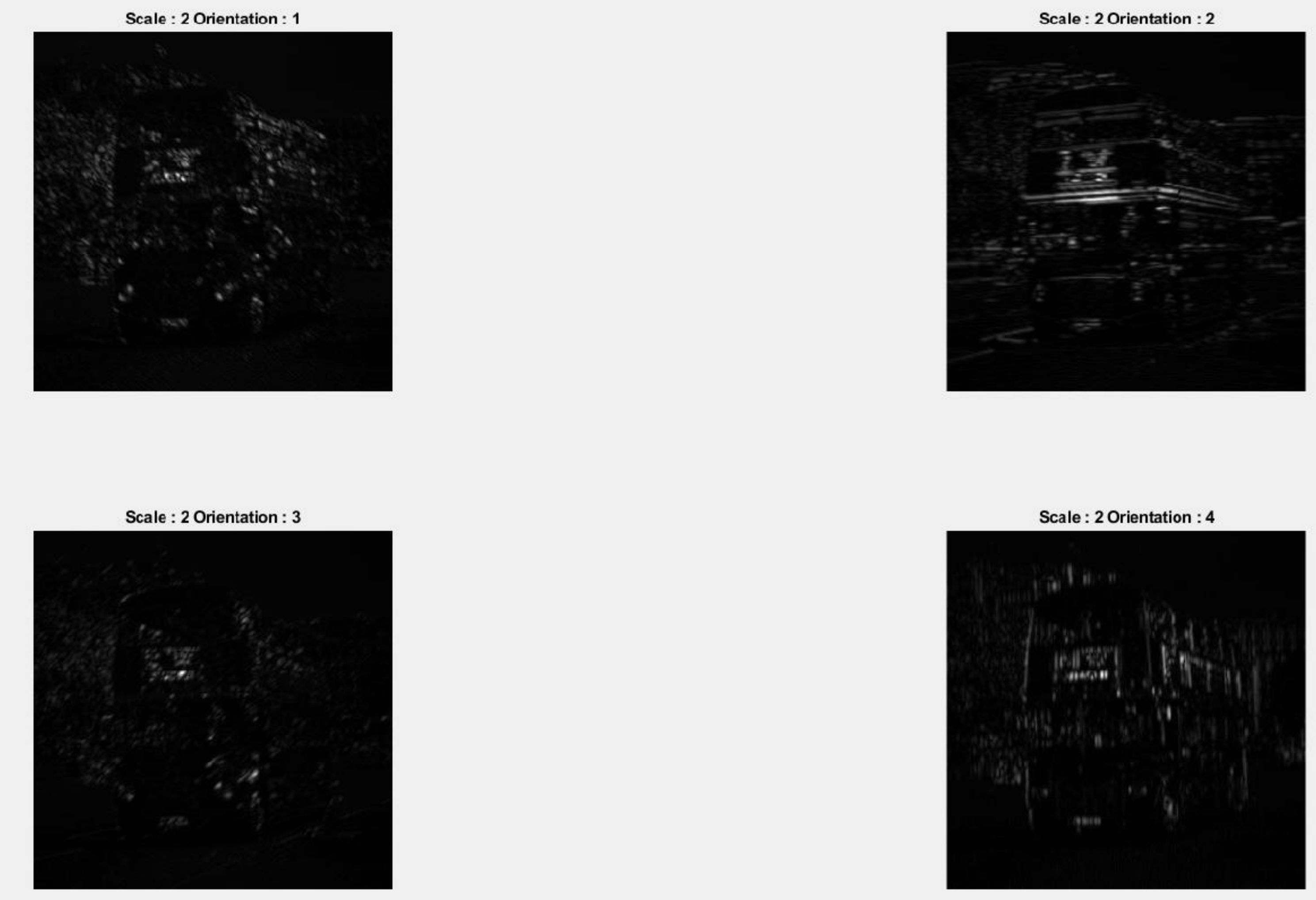

Figure 2: Bus Image for Scale 3.3 and Orientations 45-90-135-180

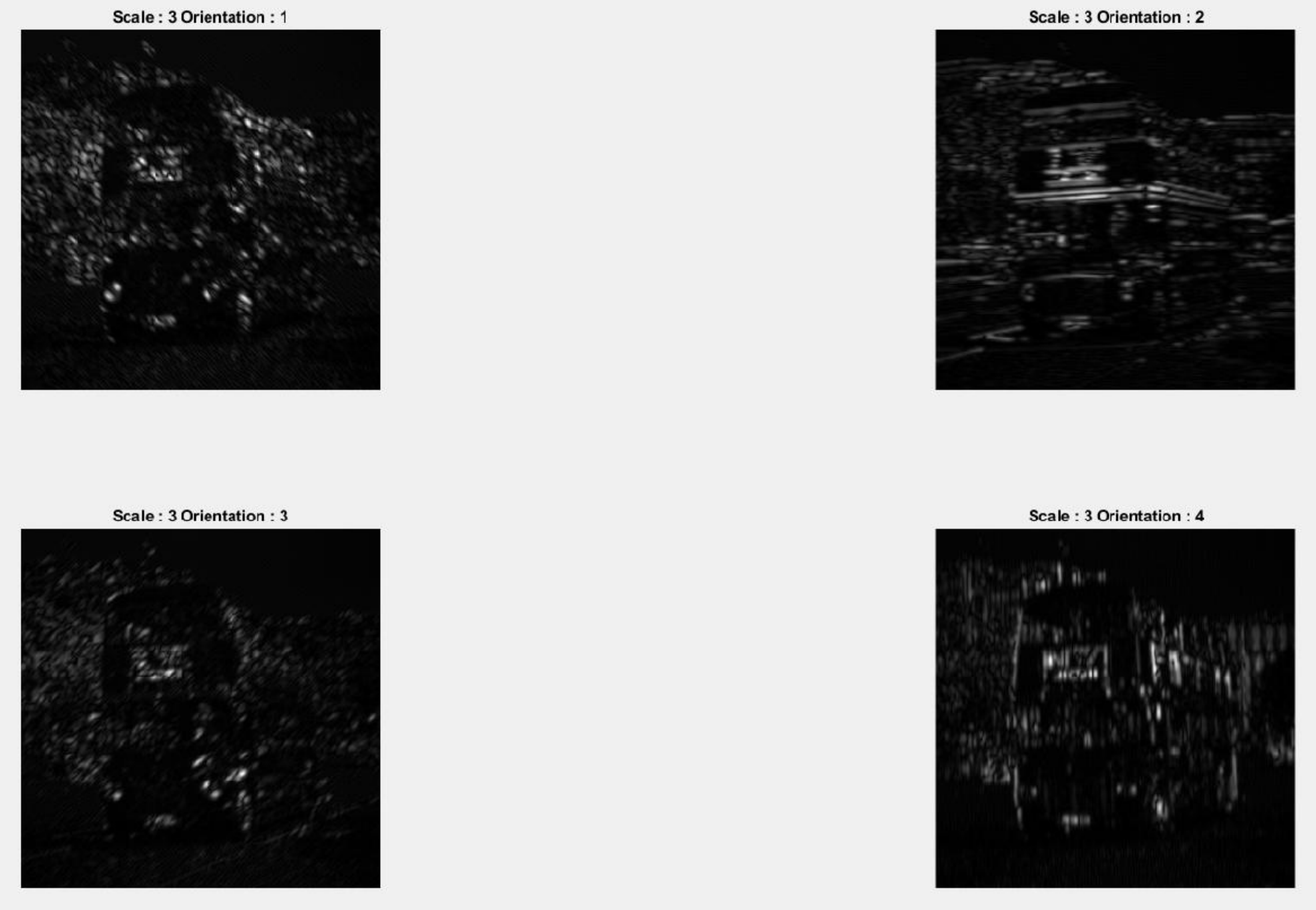

Figure 3: Bus Image for Scale 4.6 and Orientations 45-90-135-180

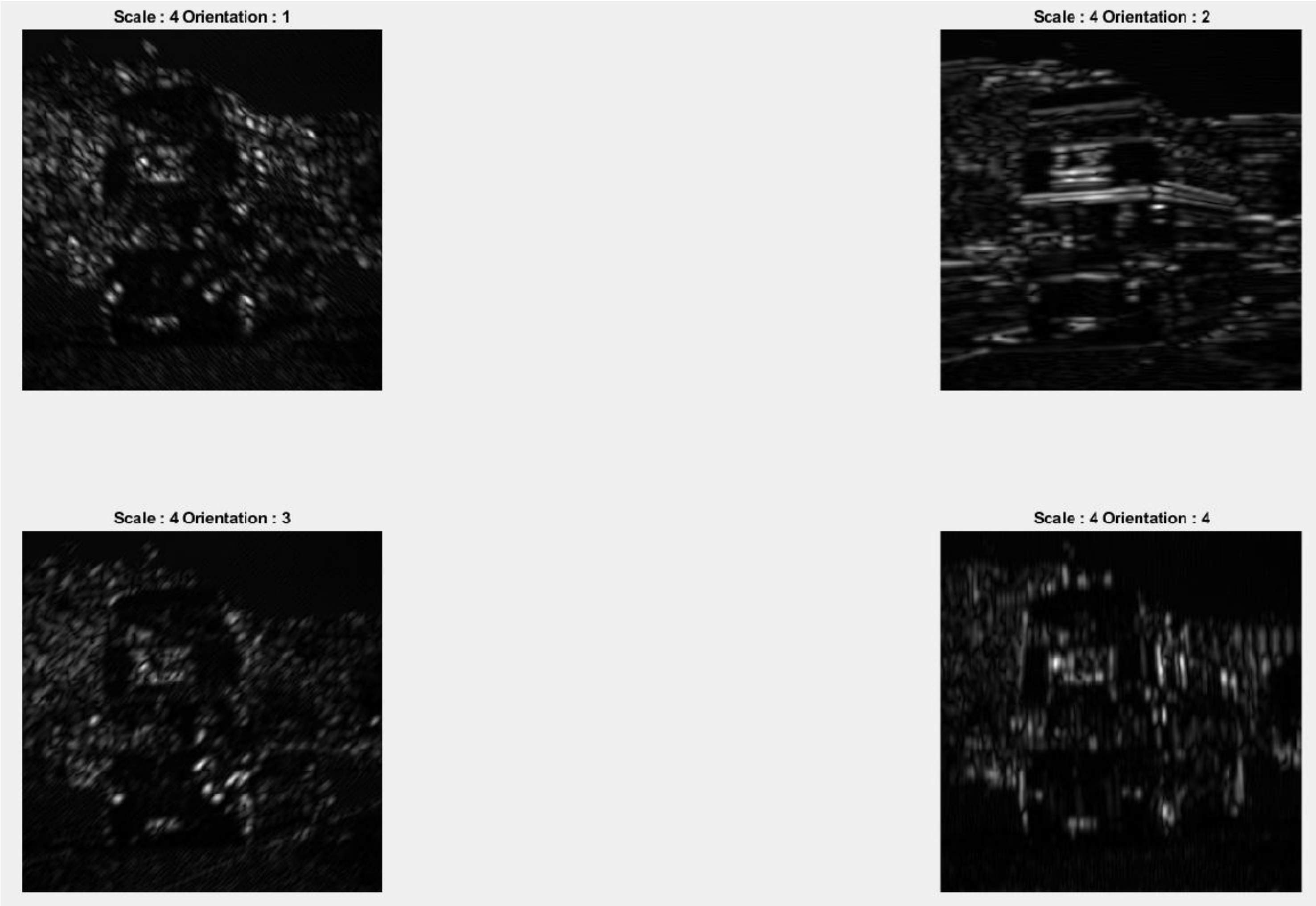

Figure 4: Bus Image for Scale 6 and Orientations 45-90-135-180

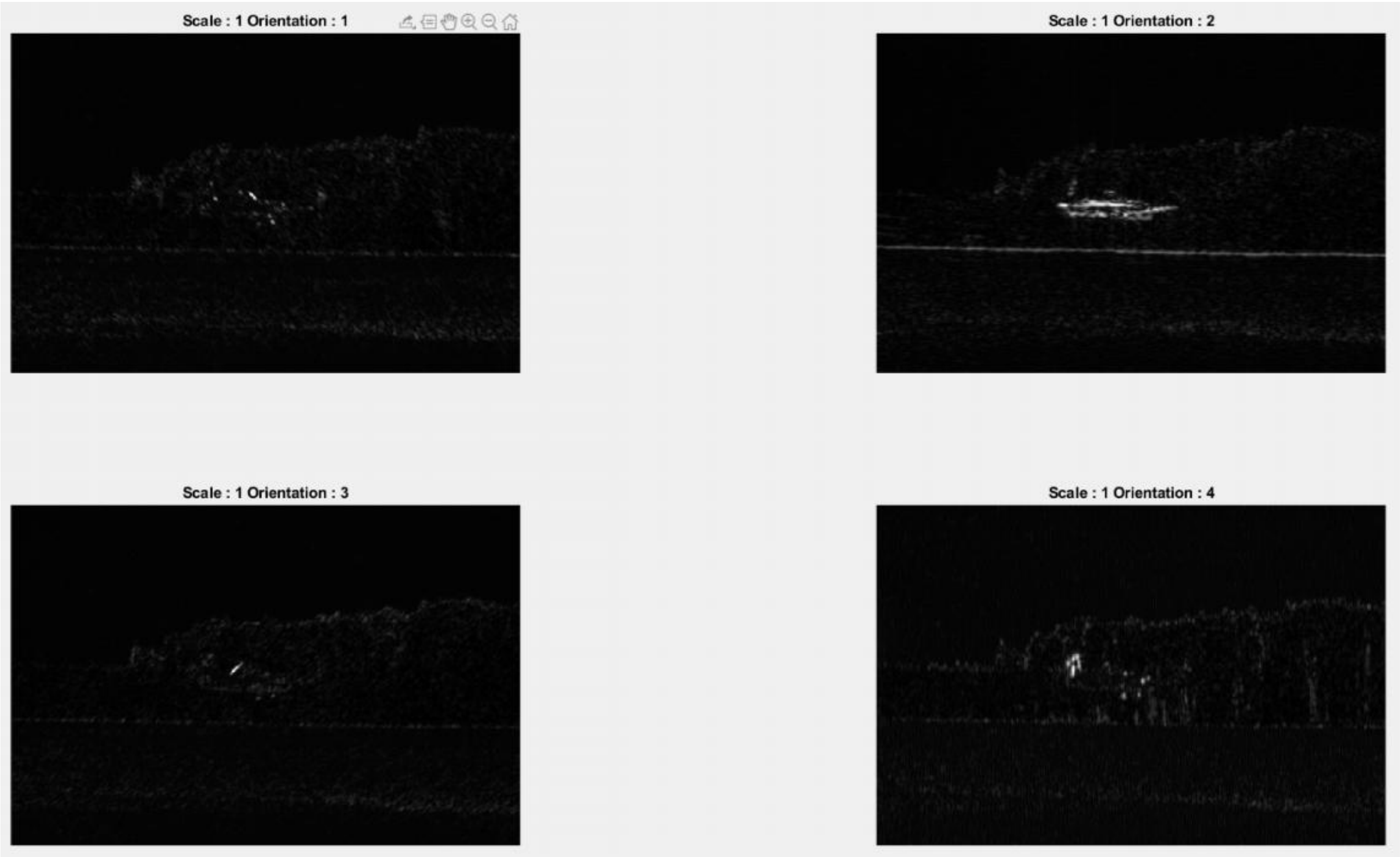

Figure 5: Plane Image for Scale 2 and Orientations 45-90-135-180

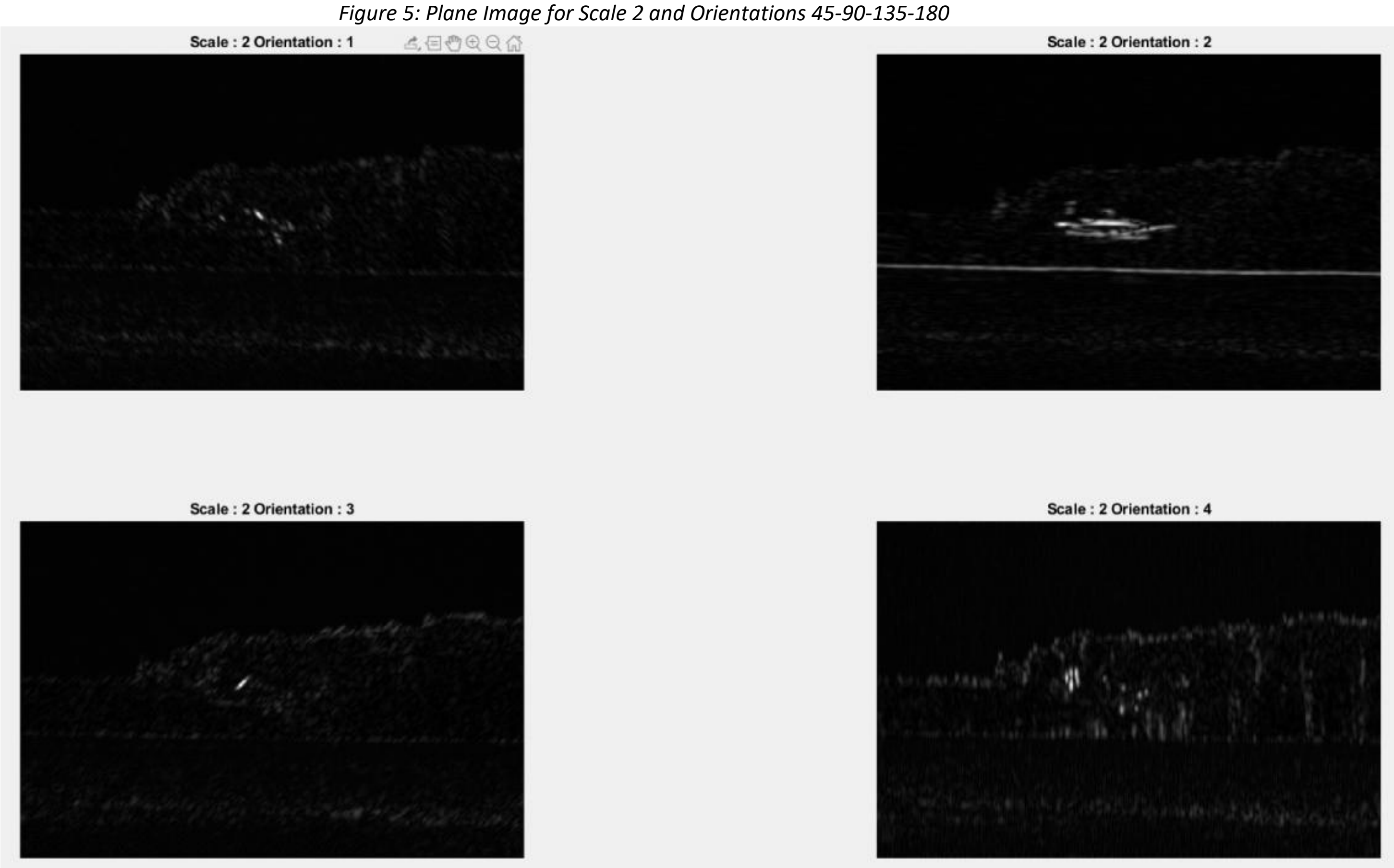

Figure 6: Plane Image for Scale 3.3 and Orientations 45-90-135-180

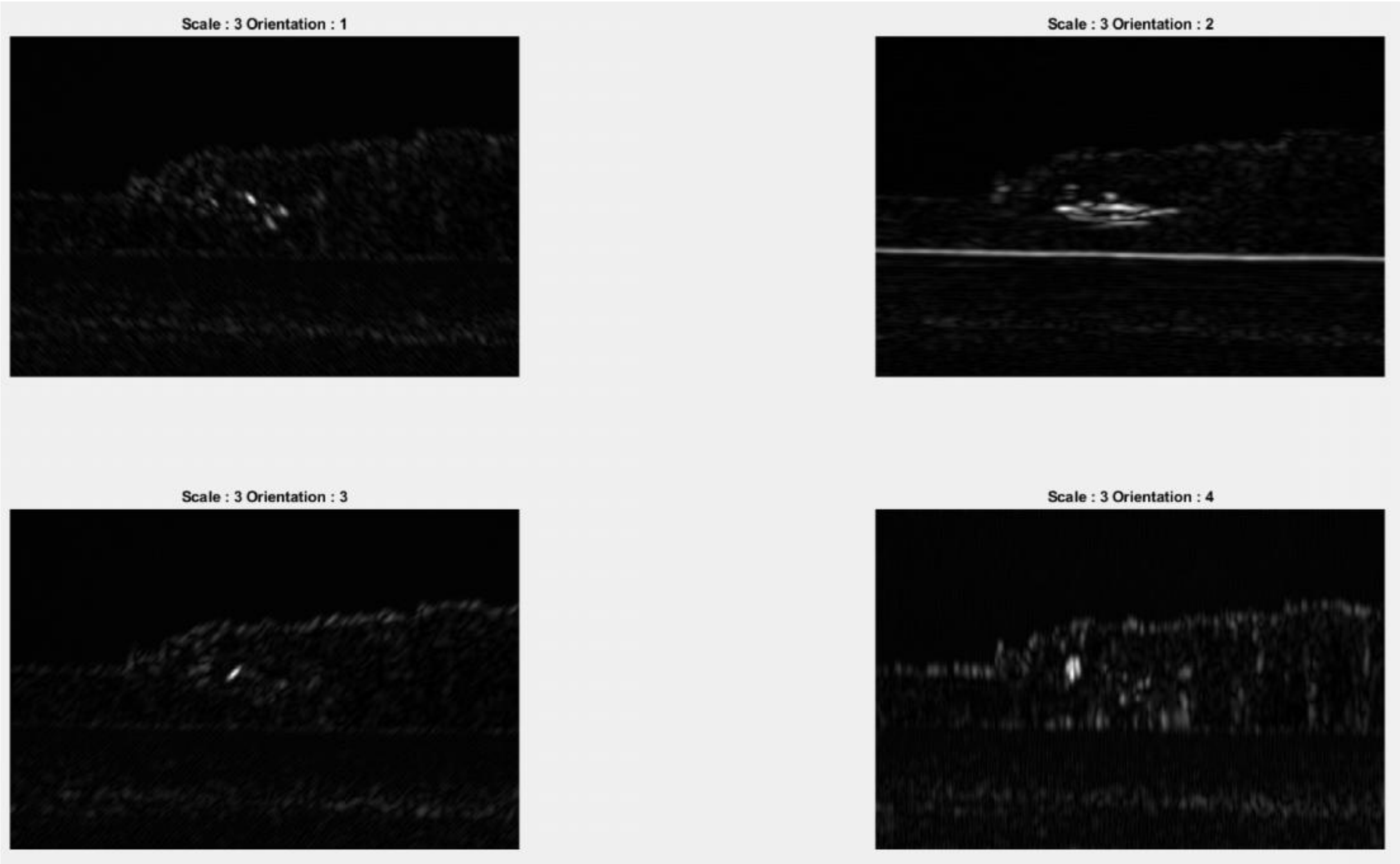

Figure 7: Plane Image for Scale 4.6 and Orientations 45-90-135-180

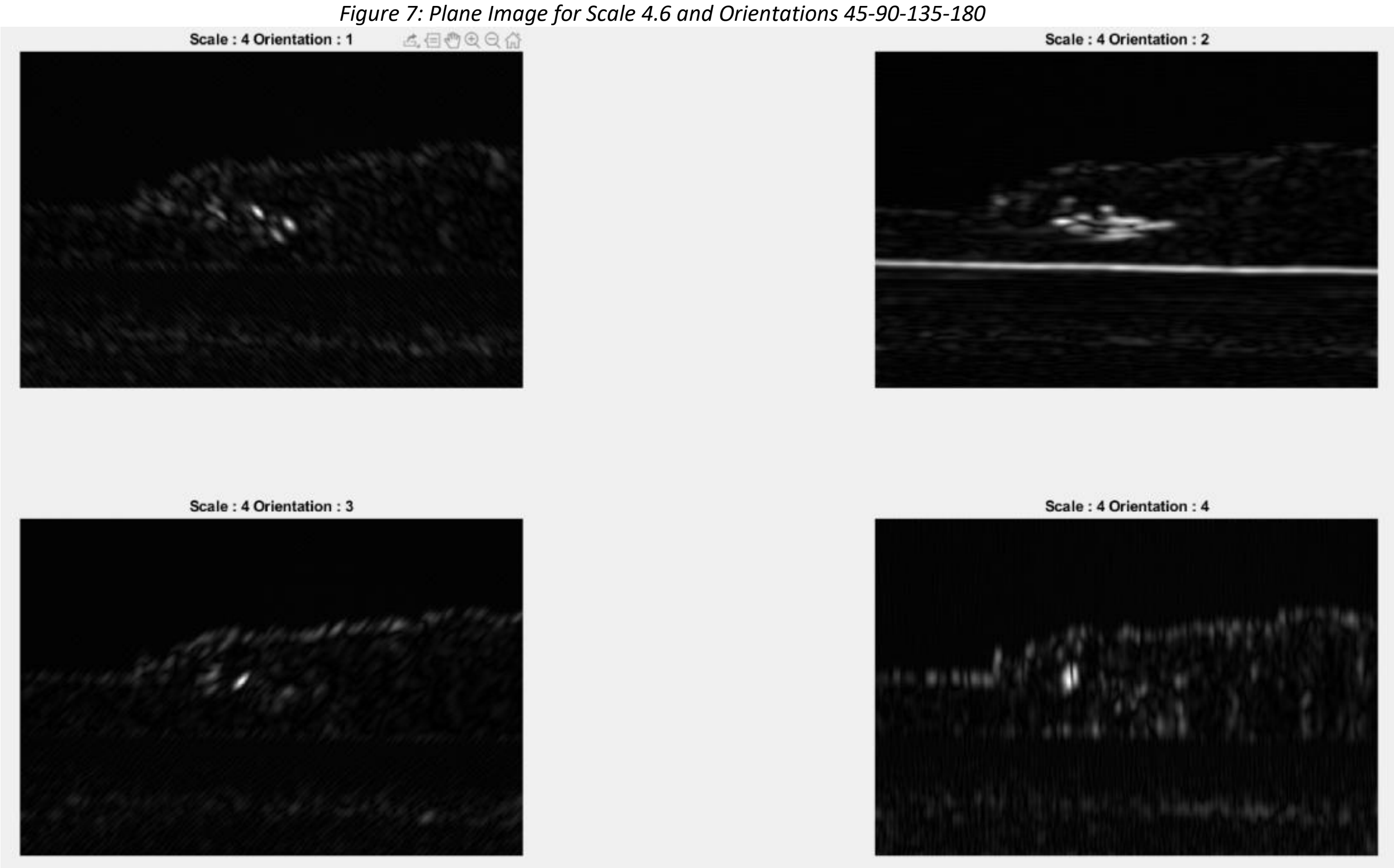

Figure 8: Plane Image for Scale 6 and Orientations 45-90-135-180

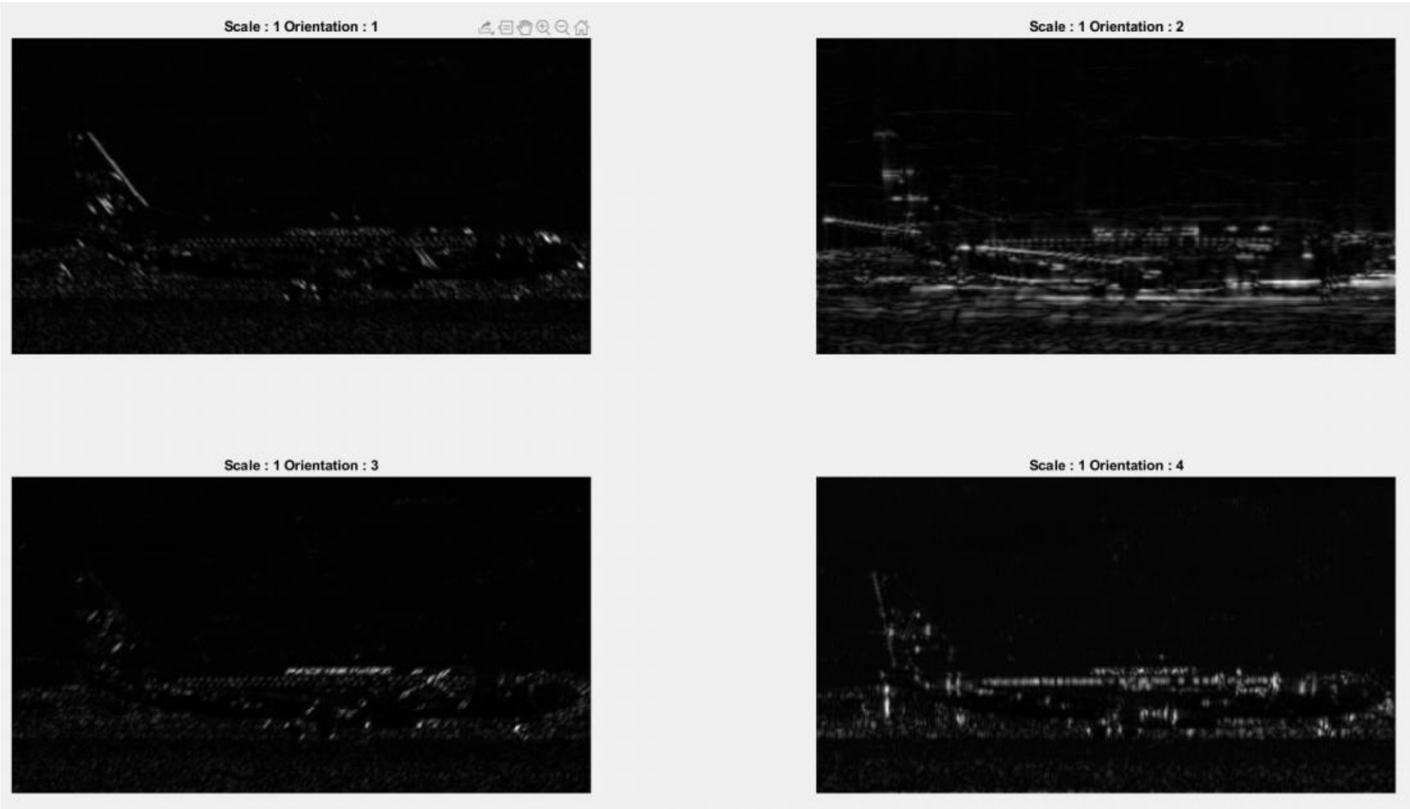

Figure 9: Plane Image for Scale 2 and Orientations 45-90-135-180

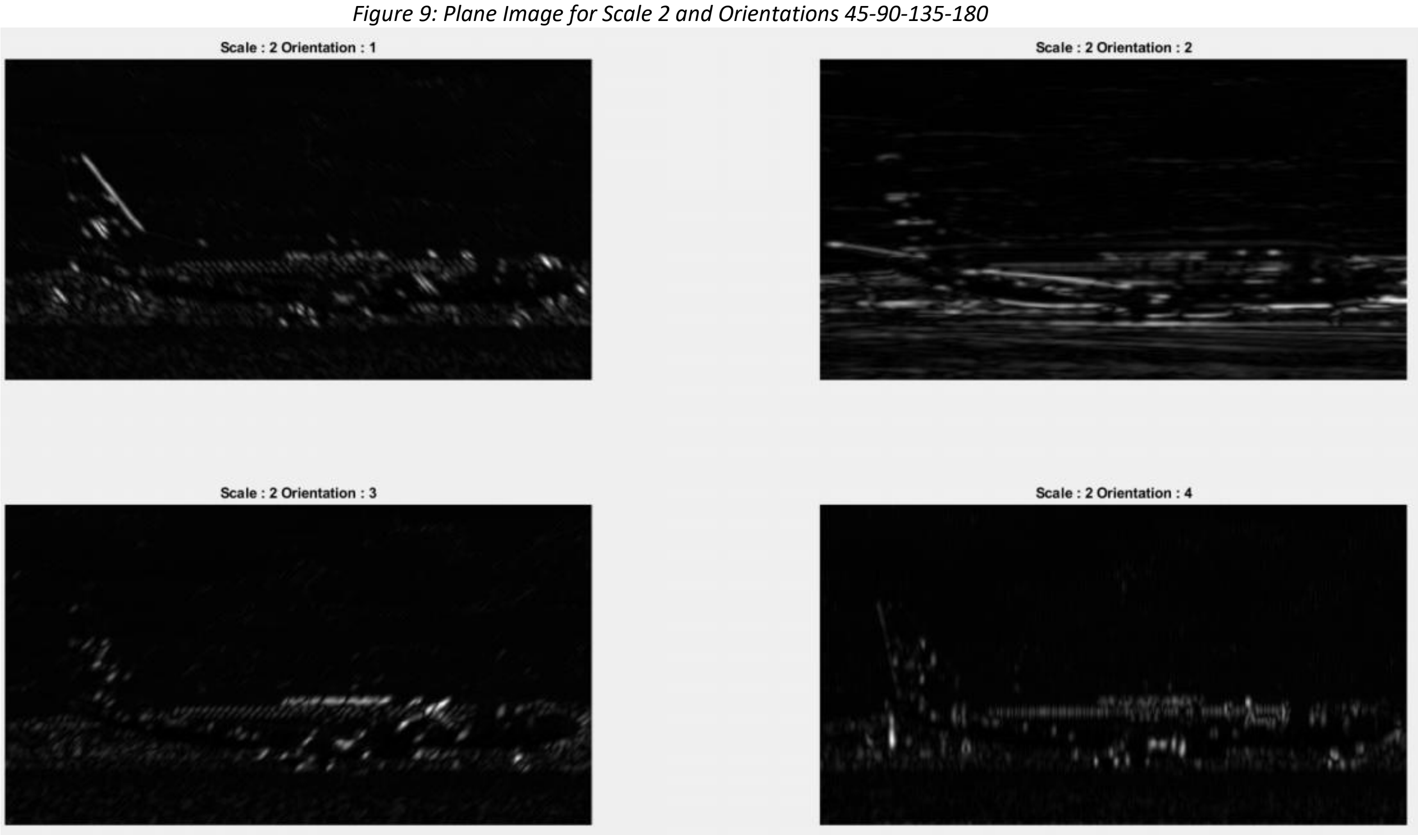

Figure 10: Plane Image for Scale 3.3 and Orientations 45-90-135-180

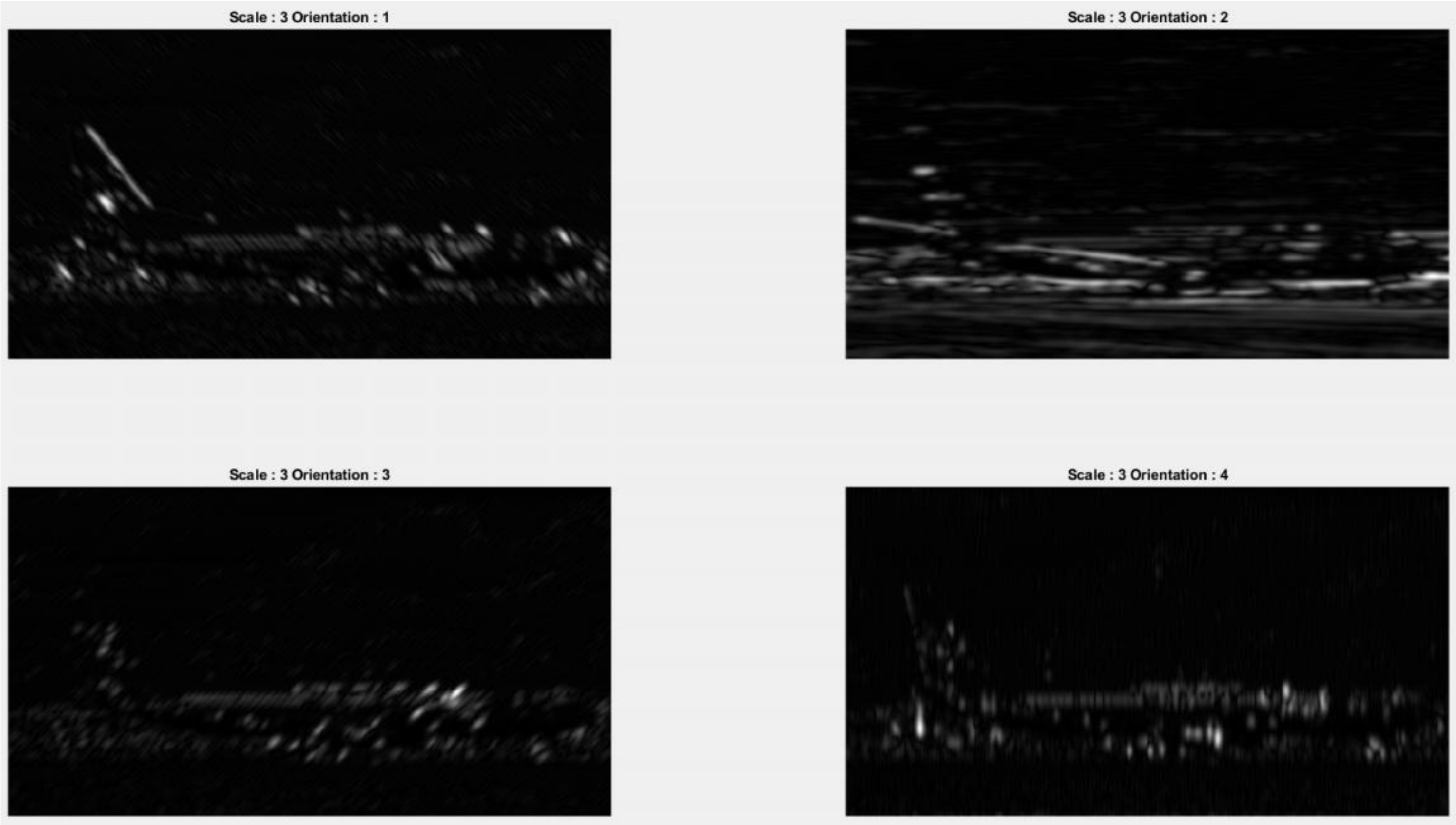

Figure 11: Plane Image for Scale 4.6 and Orientations 45-90-135-180

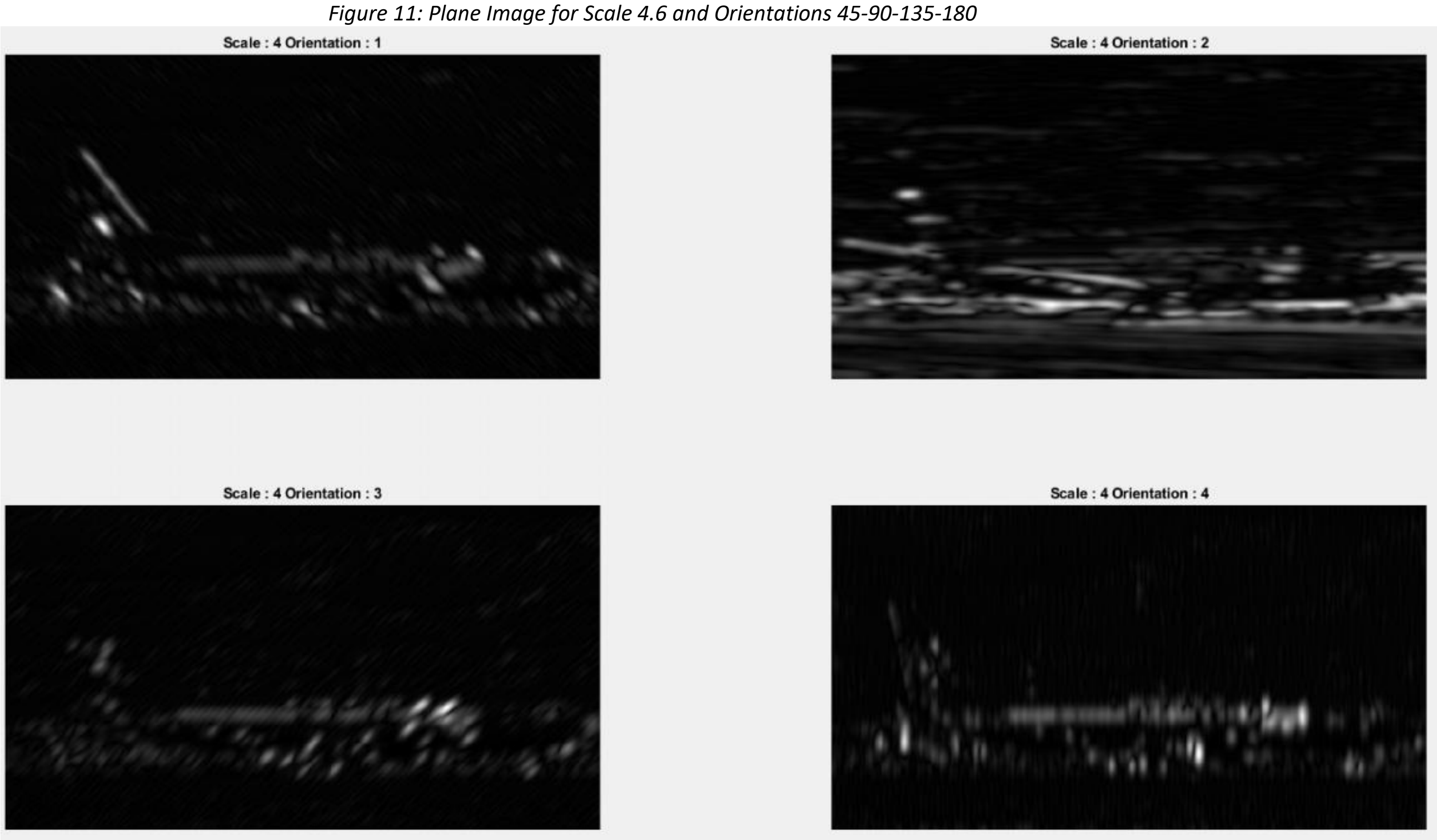

Figure 12: Plane Image for Scale 6 and Orientations 45-90-135-180

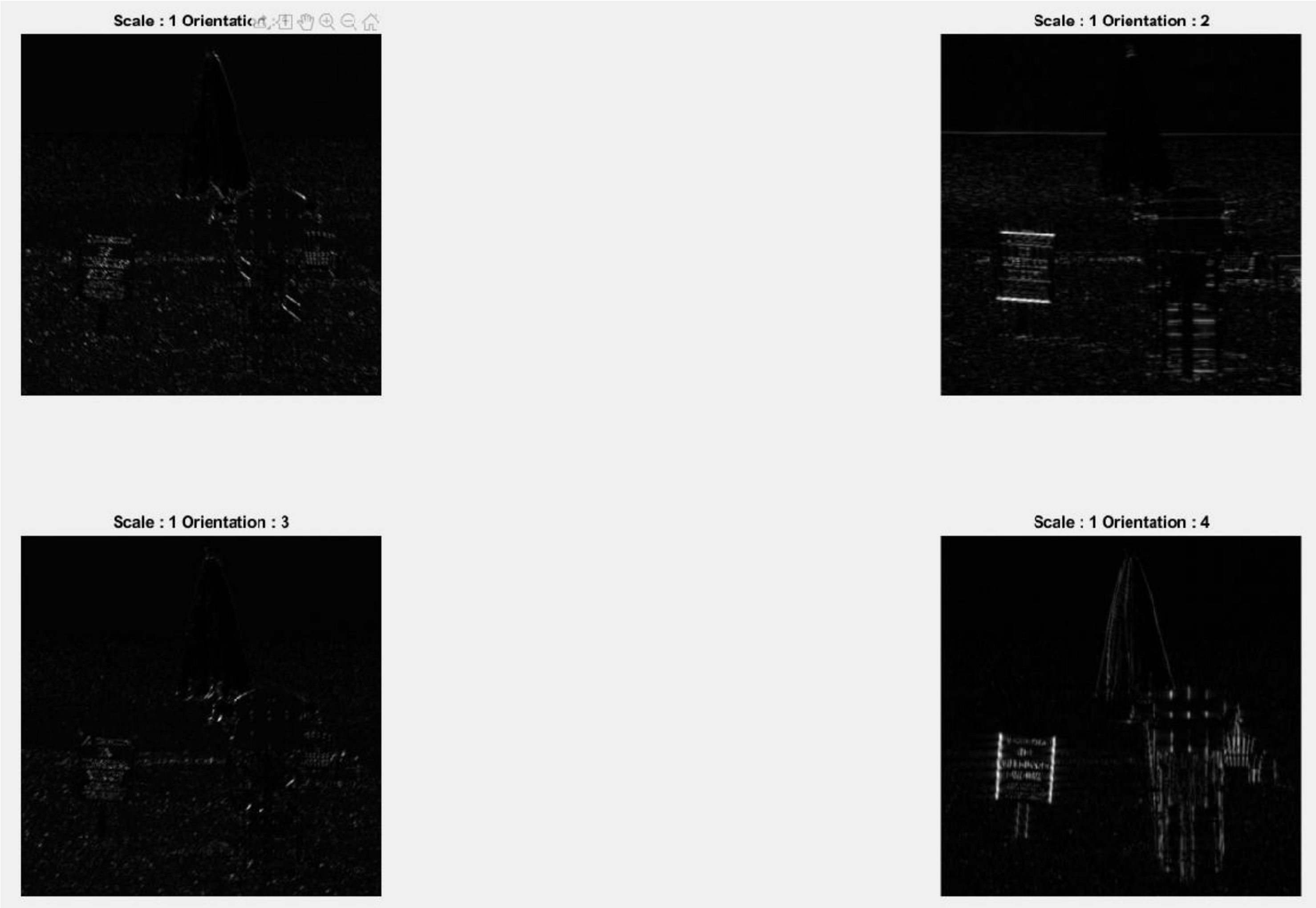

Figure 13: Beach Image for Scale 2 and Orientations 45-90-135-180

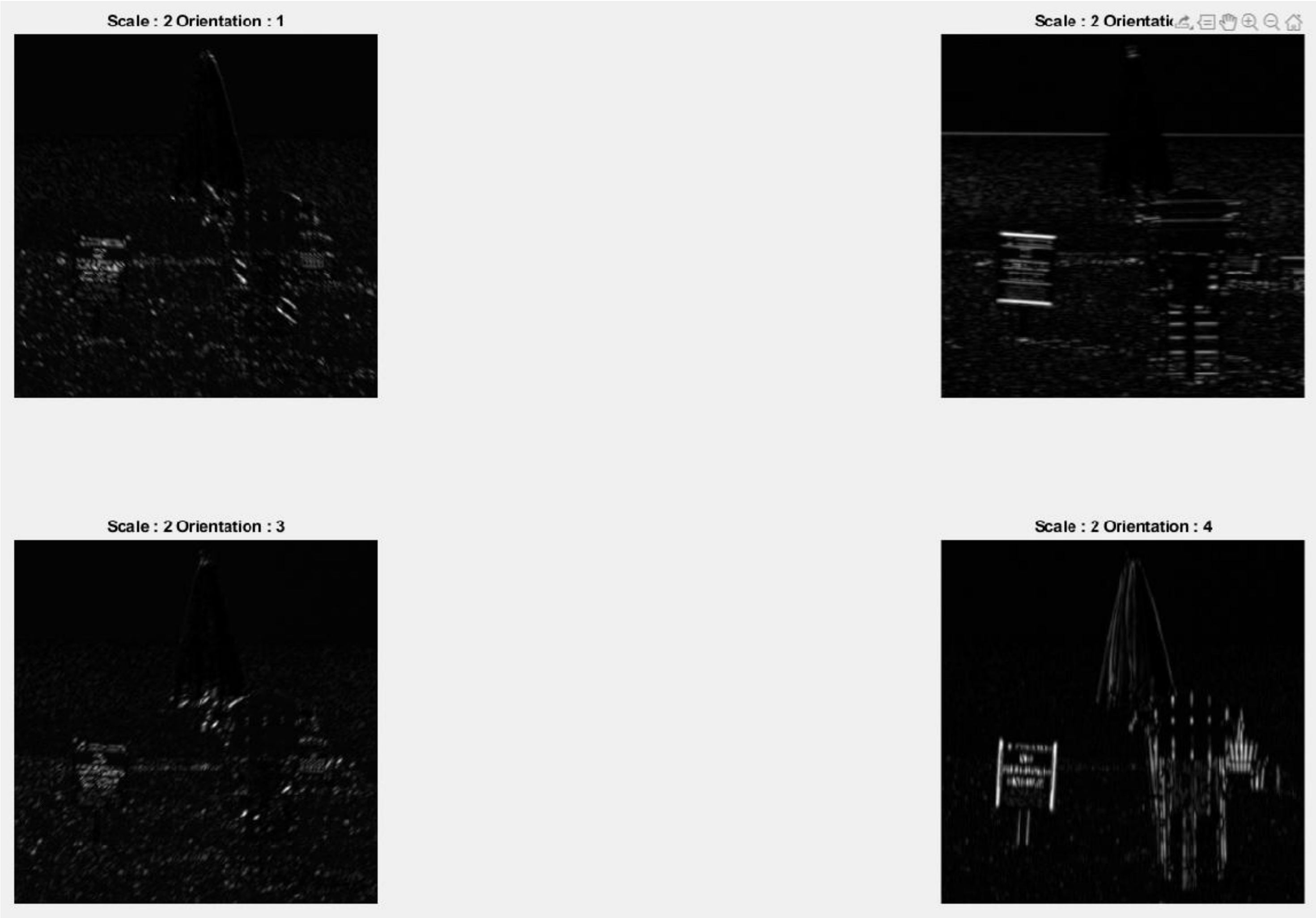

Figure 14: Beach Image for Scale 3.3 and Orientations 45-90-135-180

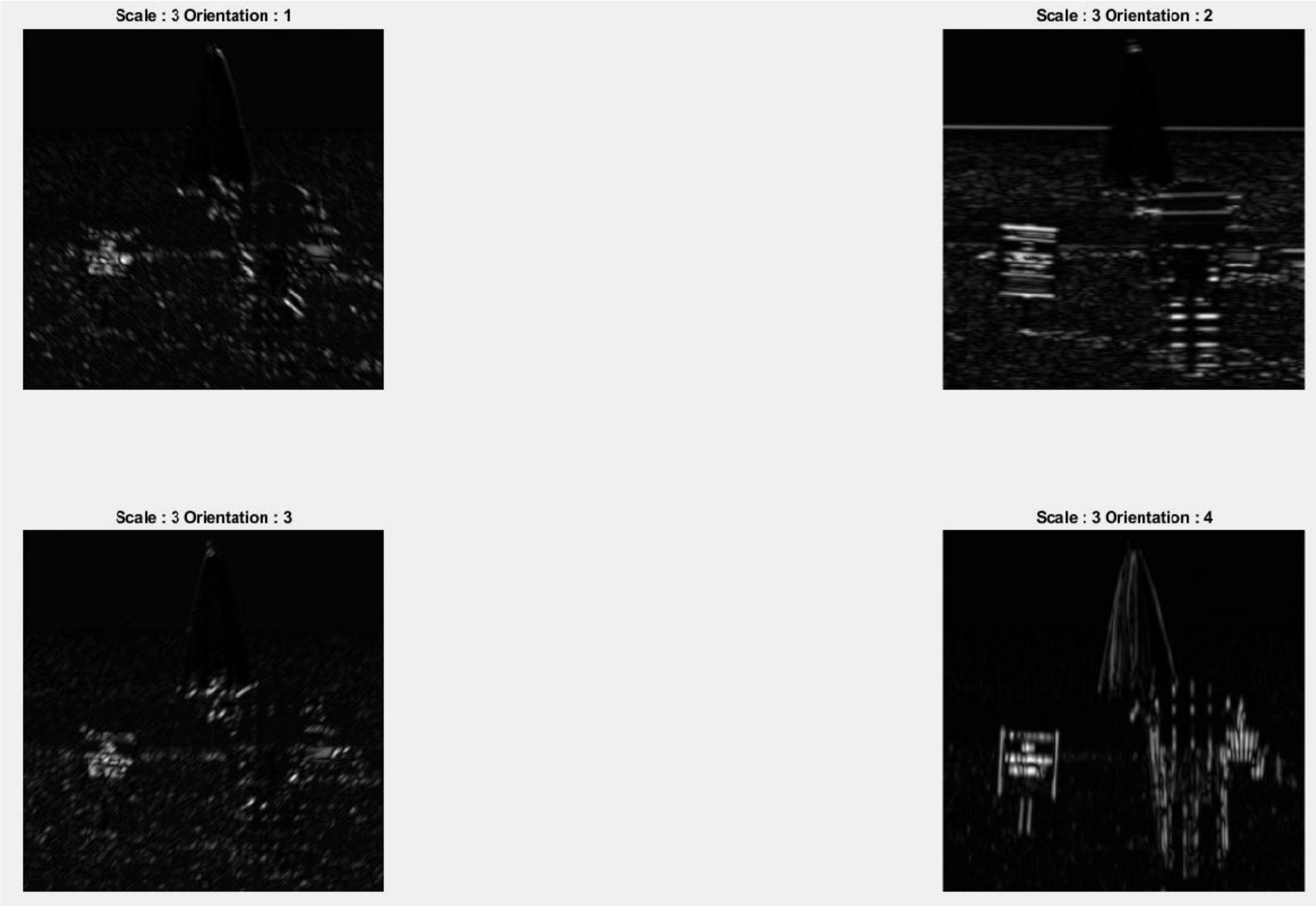

Figure 15: Beach Image for Scale 4.6 and Orientations 45-90-135-180

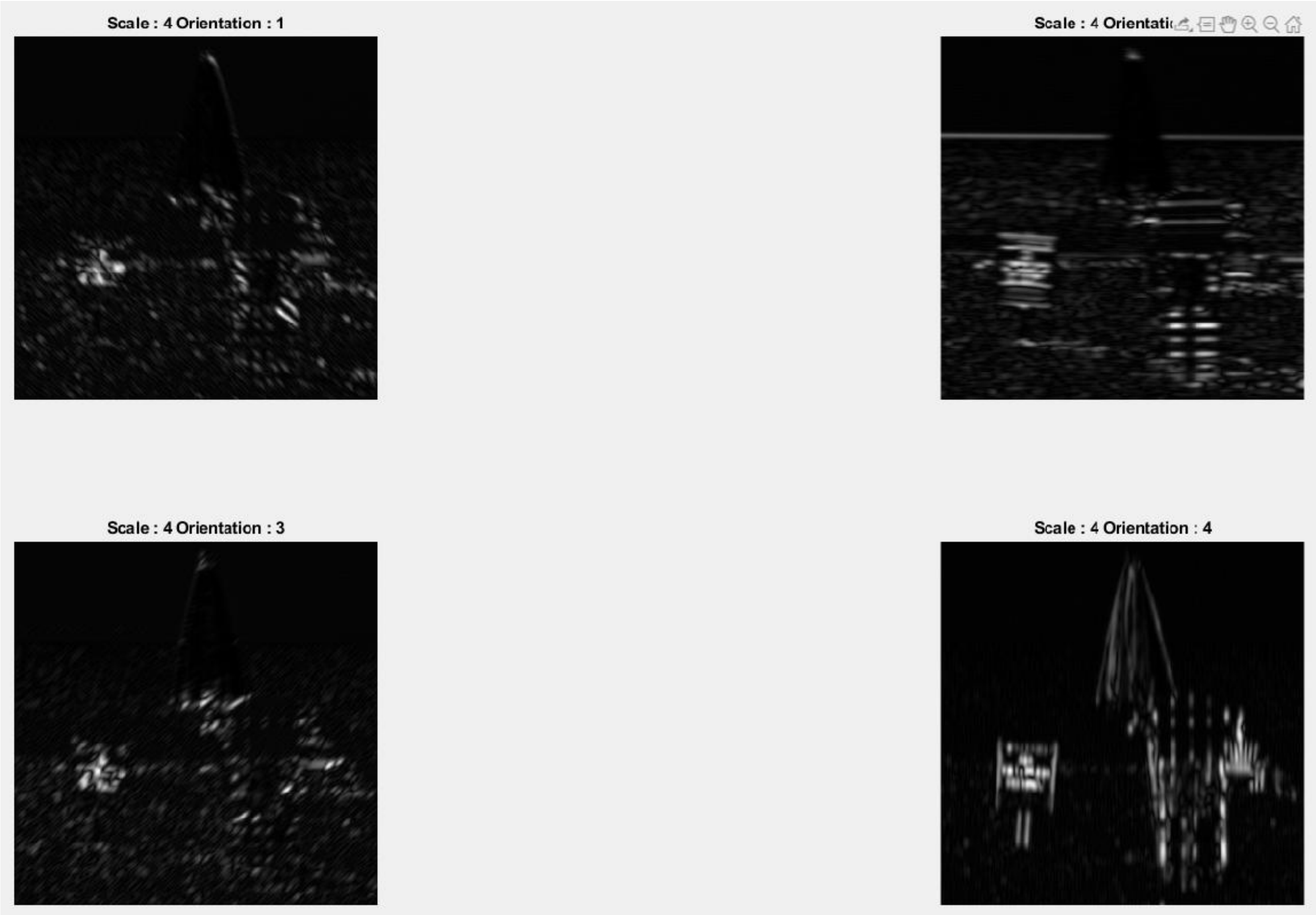

Figure 16: Beach Image for Scale 6 and Orientations 45-90-135-180

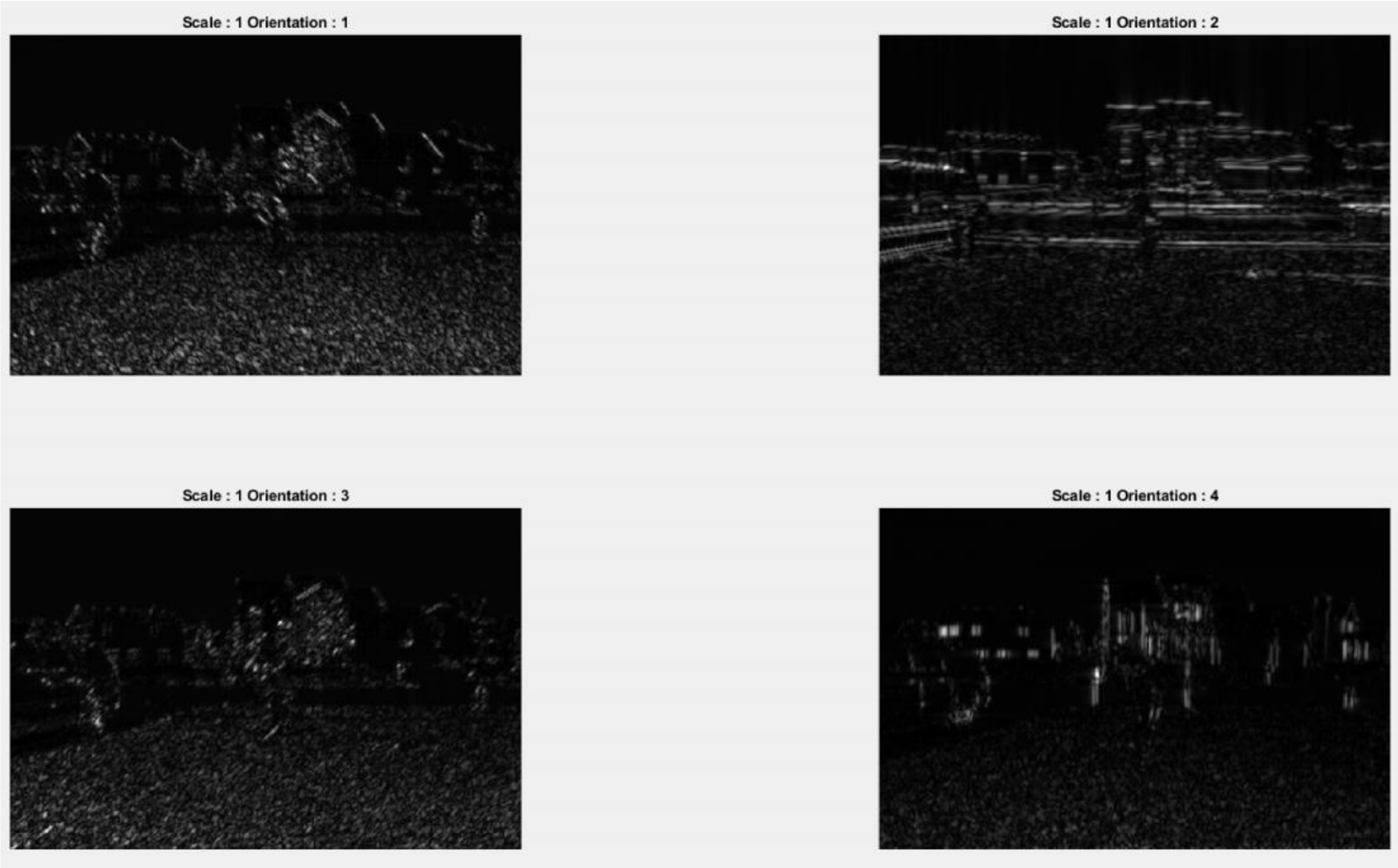

Figure 17: Houses Image for Scale 2 and Orientations 45-90-135-180

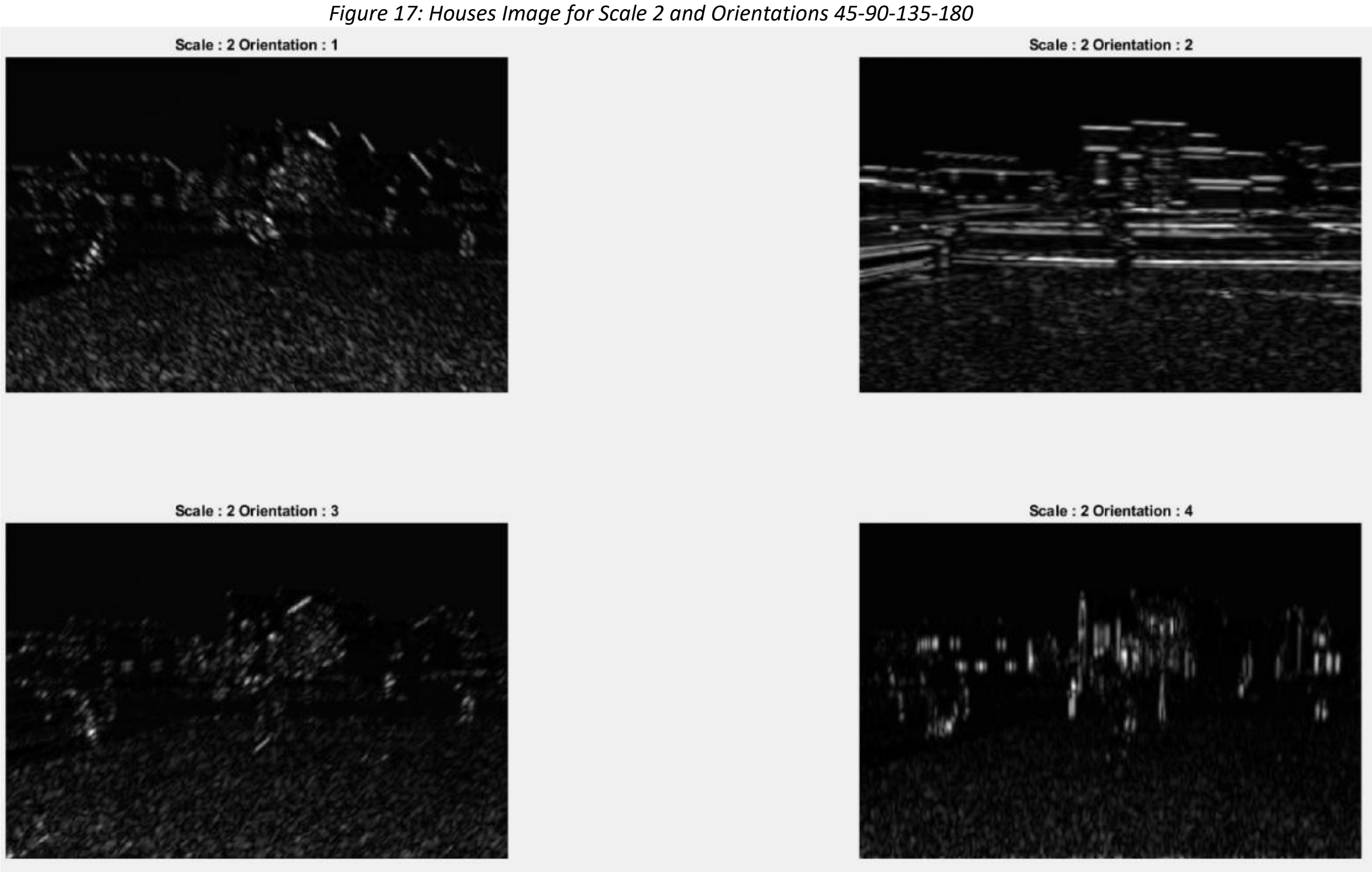

Figure 18: Houses Image for Scale 3.3 and Orientations 45-90-135-180

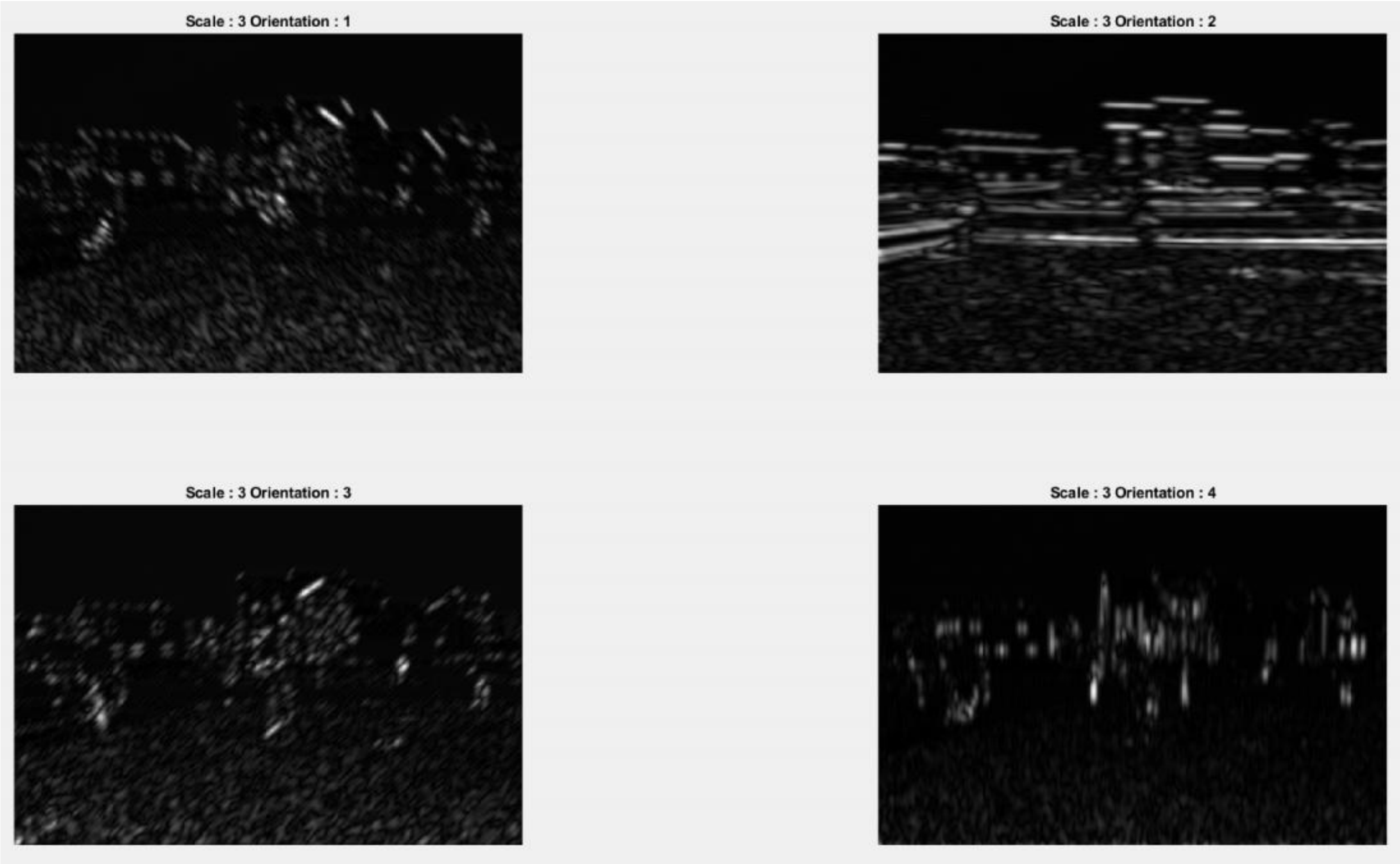

Figure 19: Houses Image for Scale 4.6 and Orientations 45-90-135-180

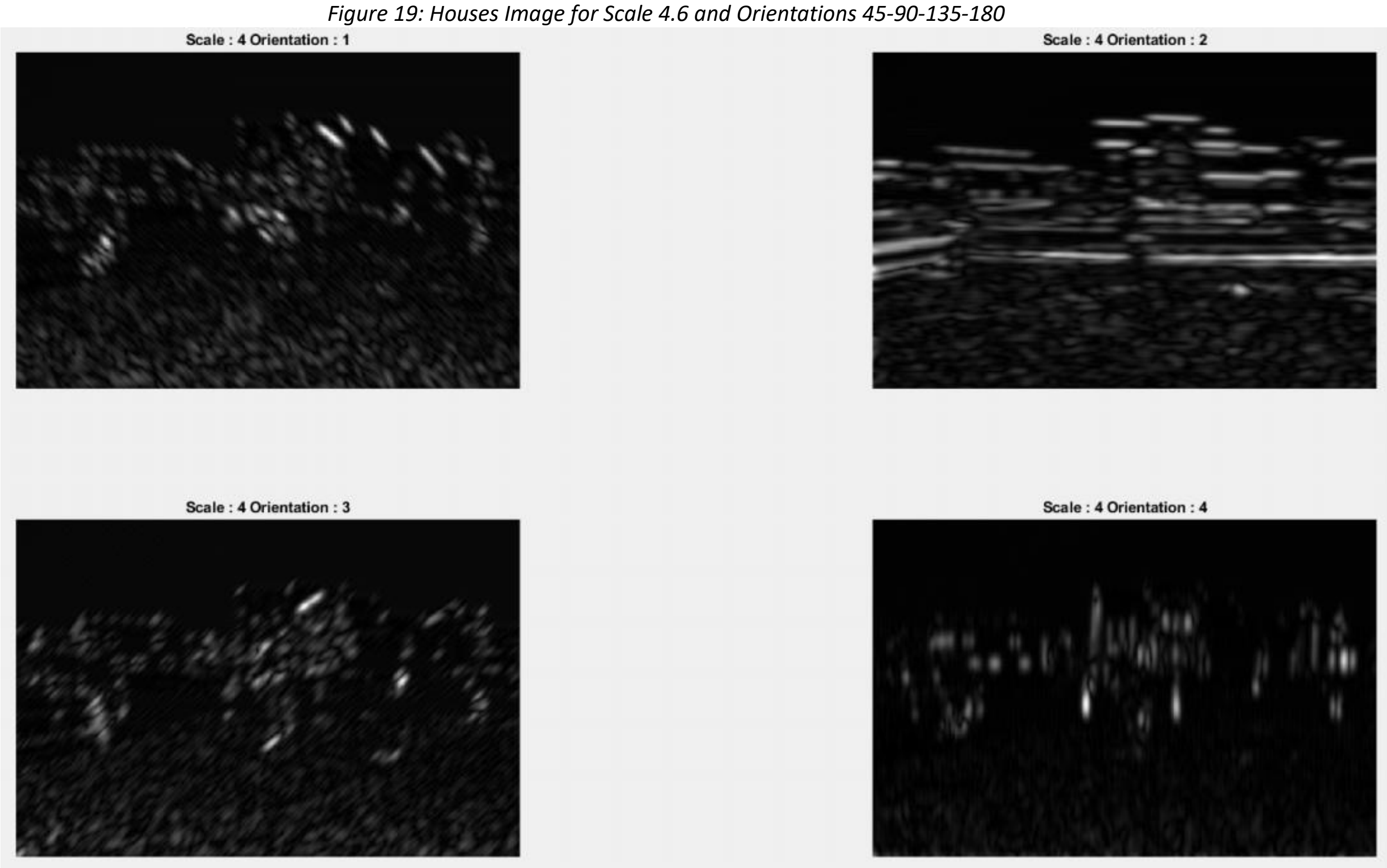

Figure 20: Houses Image for Scale 6 and Orientations 45-90-135-180

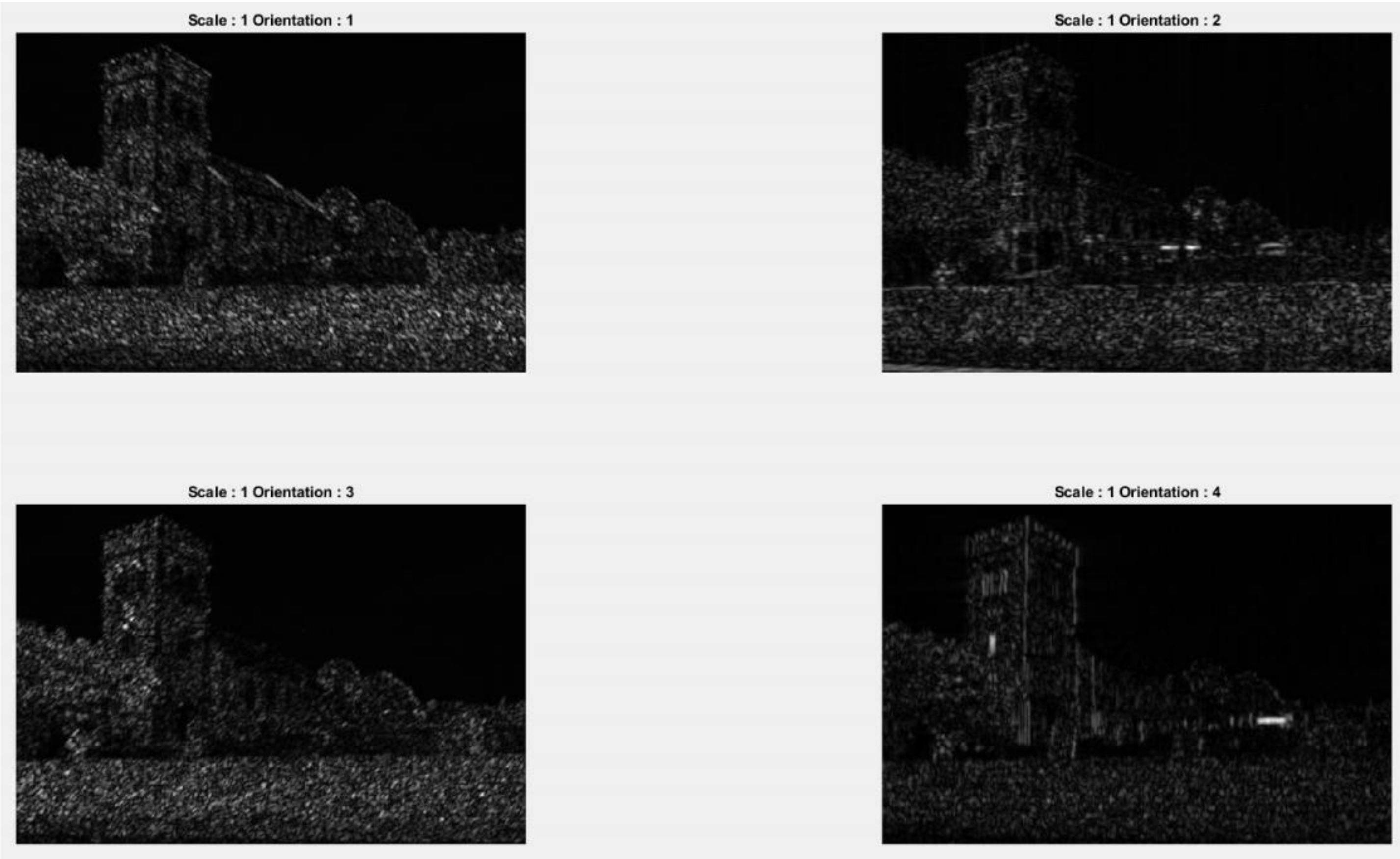

Figure 21: Cemetery Image for Scale 2 and Orientations 45-90-135-180

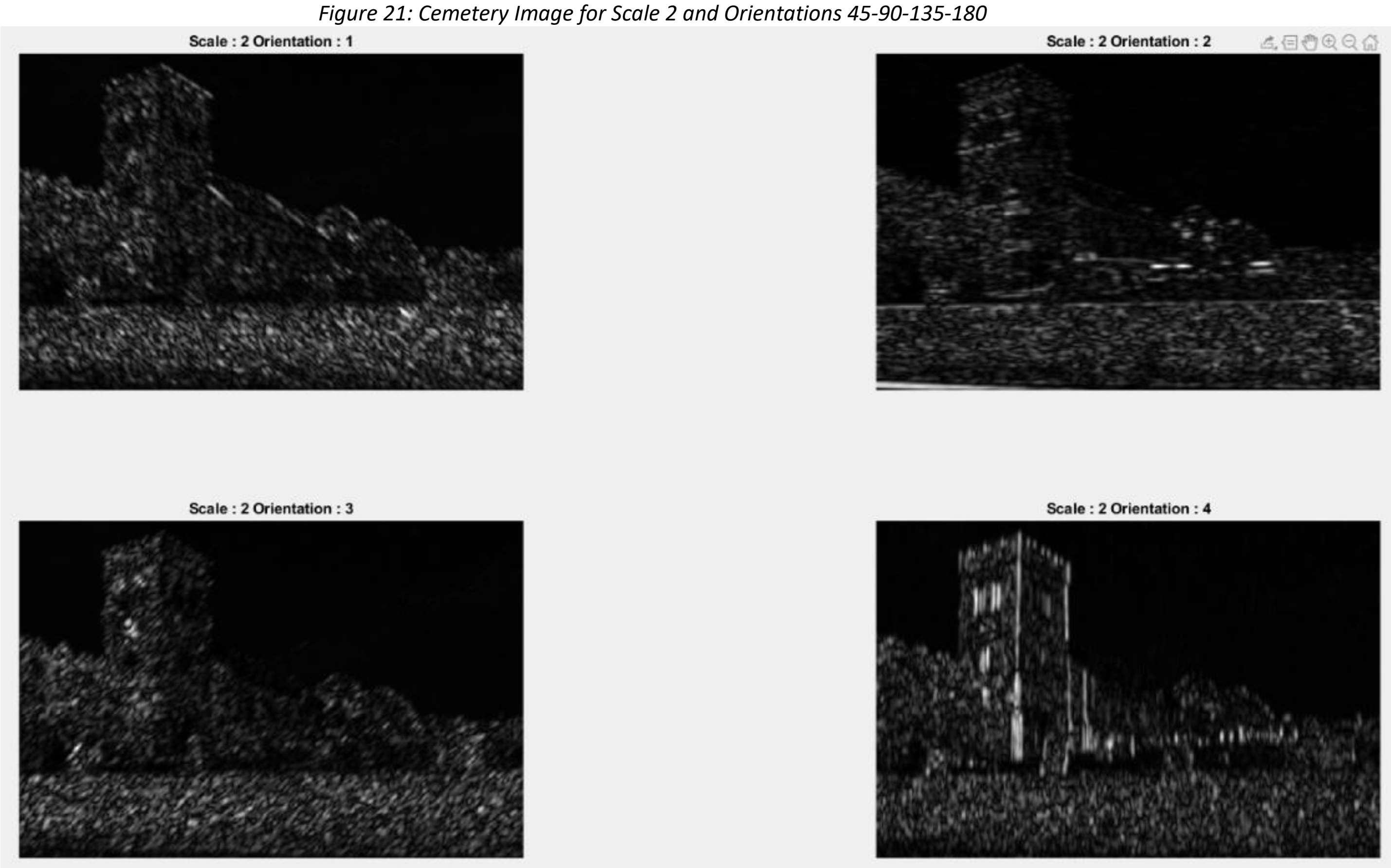

Figure 22: Cemetery Image for Scale 3.3 and Orientations 45-90-135-180

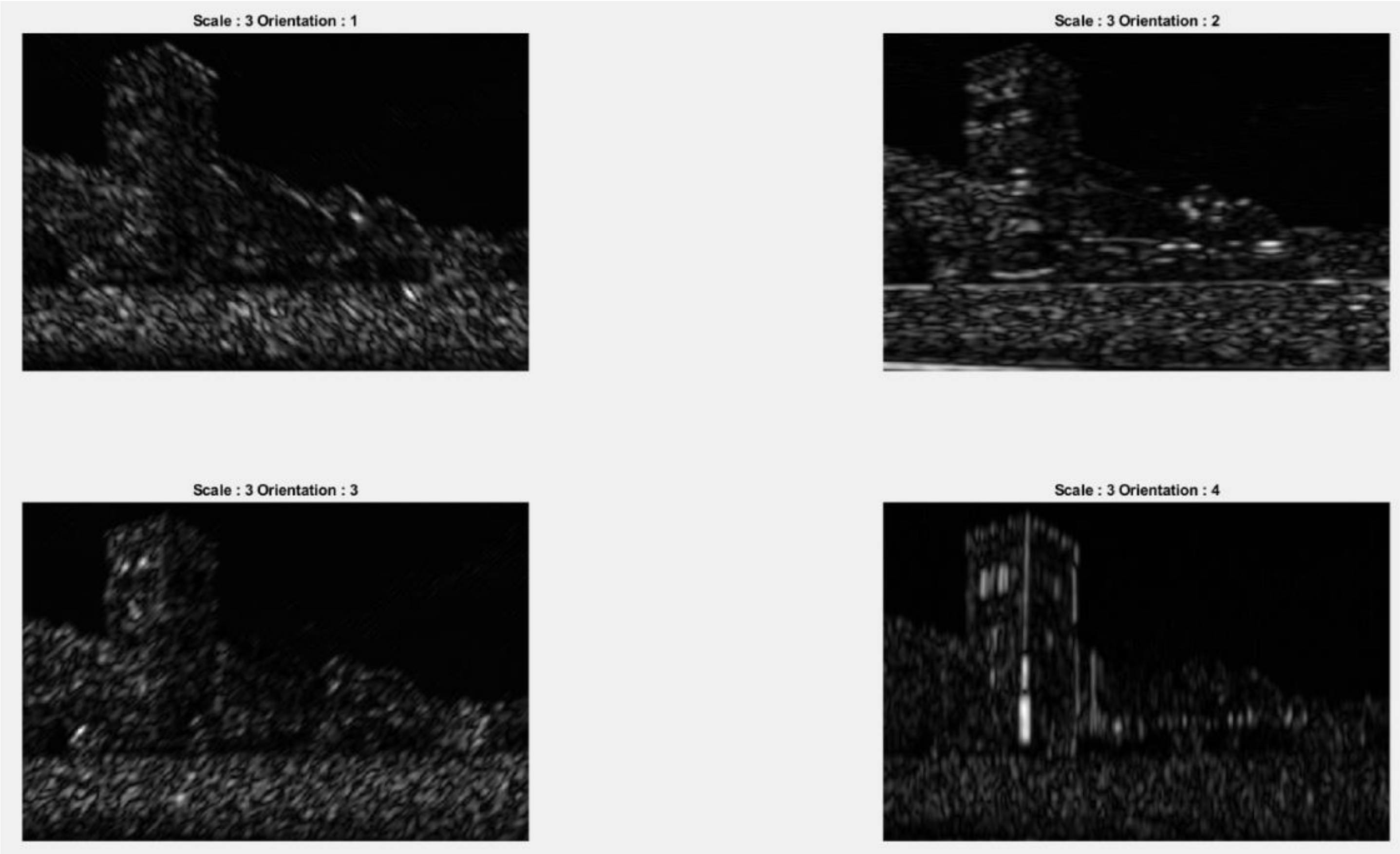

Figure 23: Cemetery Image for Scale 4.6 and Orientations 45-90-135-180

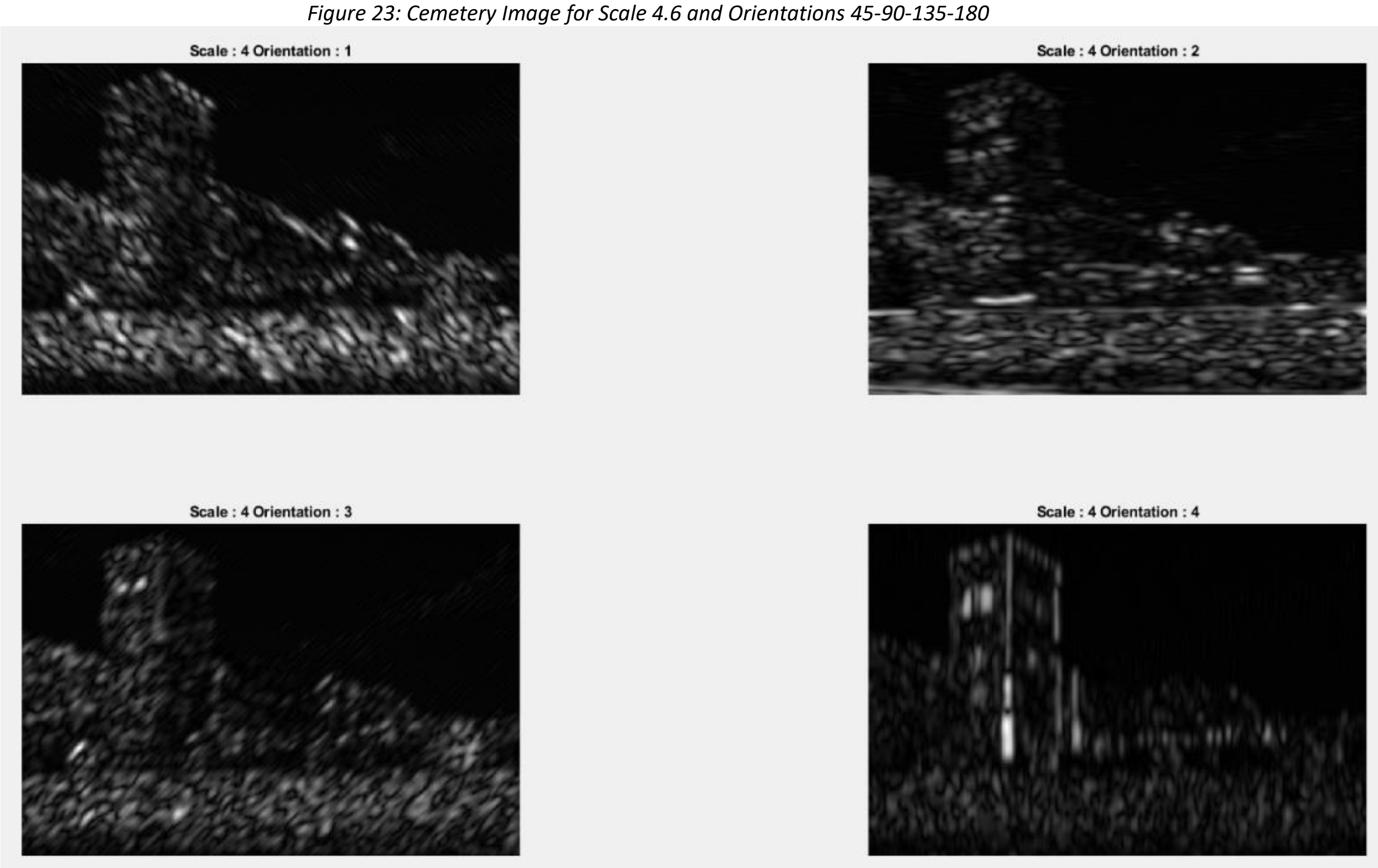

Figure 24: Cemetery Image for Scale 6 and Orientations 45-90-135-180

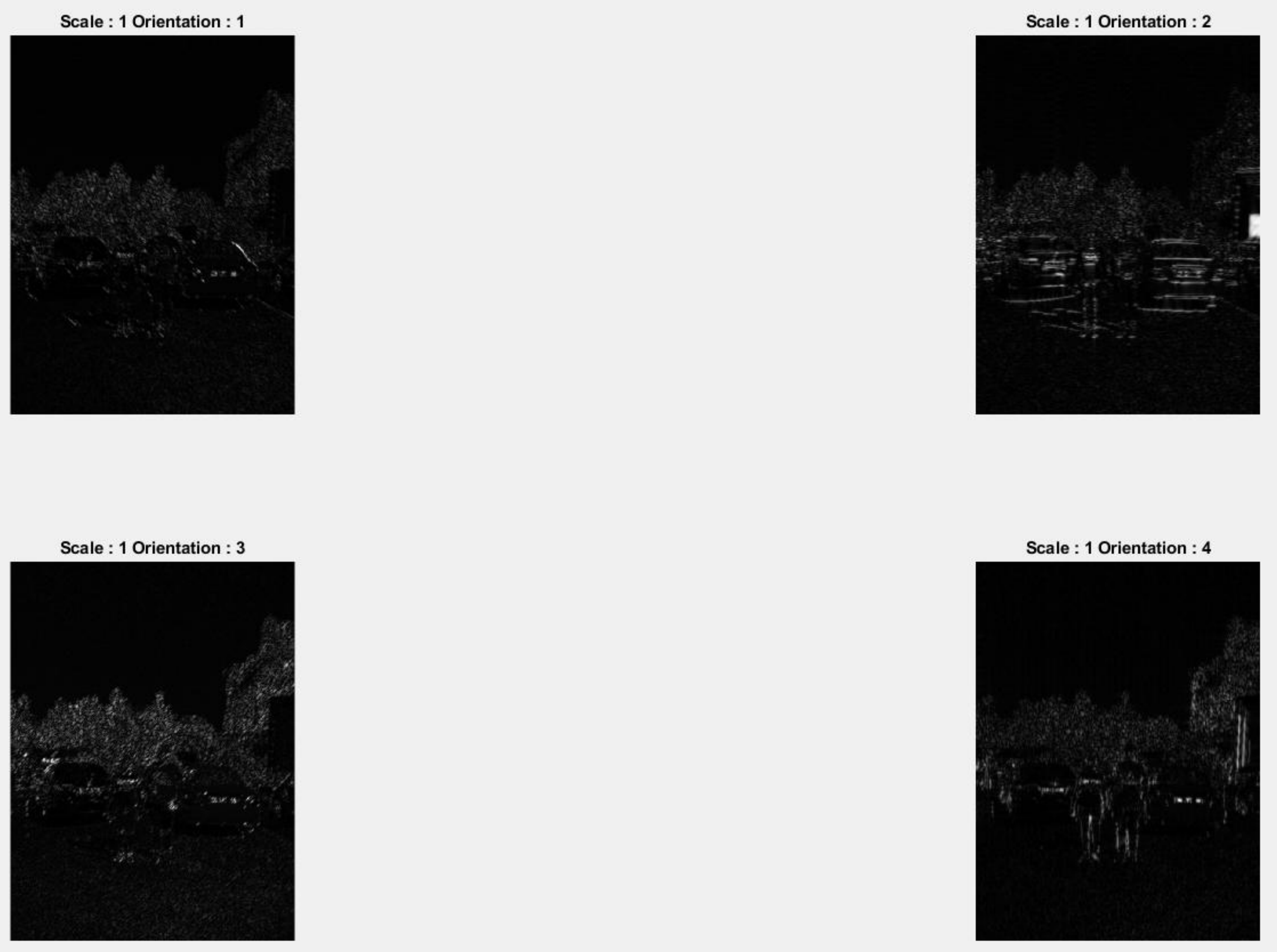

Figure 25: Image I used in my First Homework for Scale 2 and Orientations 45-90-135-180

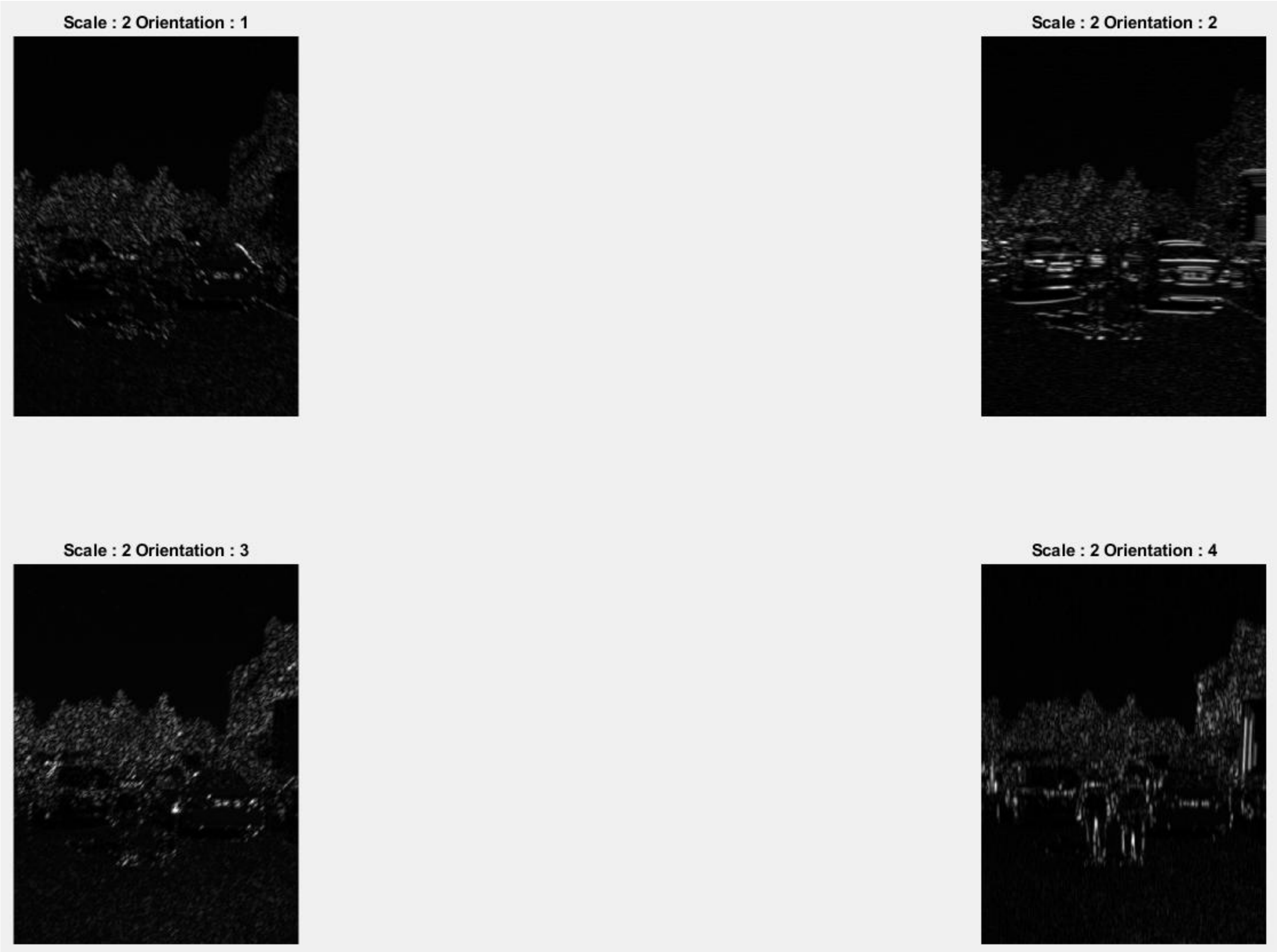

Figure 26: Image I used in my First Homework for Scale 3.3 and Orientations 45-90-135-180

Figure 27: Image I used in my First Homework for Scale 4.6 and Orientations 45-90-135-180

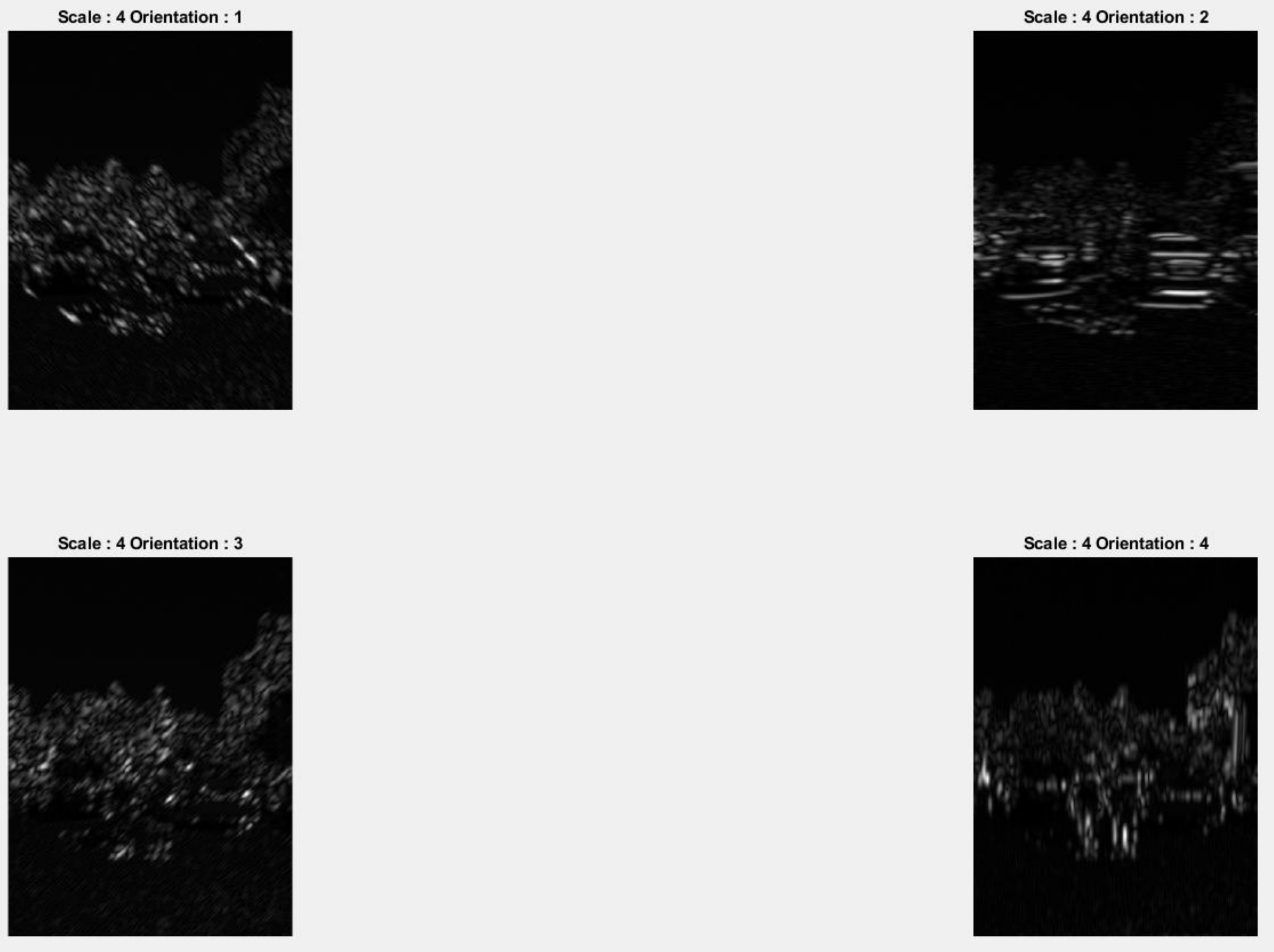

Figure 28: Image I used in my First Homework for Scale 6 and Orientations 45-90-135-180

After getting the Gabor magnitudes, I calculated their means for all superpixels for by four. The only parameter for this part was Gabor magnitudes and in the end, I had one cell with the number of labels rows and 4 for columns. Inside the columns, there were four values for each scale. Then I took the transpose of the array as I did in the colour part, and I had an array with 16 rows and the number of labels columns. I normalized the values for each scale and orientation and calculated the distance. While calculating the distances, I subtract each 16 value from each other. I put my threshold as 0.25, but as in the colour part, I did not get an optimal threshold for all images. I tested my threshold on several images, some of them were good and some of them was bad. While I optimized my threshold for one image, the other became bad, so I tried to find an average threshold for all images. After deciding on the threshold, I found my last Gabor labels.

Finally, I appended my colour and texture features to find the colour-texture labels. Since I normalized them before, I just appended them together and changed my Gabor array and method with three more array space. I changed the threshold as the other parts, but it was not much different from Gabor threshold since there are only three variables between them. The threshold of Gabor and colour was much more different. After completing all of my label arrays. I created masks by using `labeloverlay` created and applied them to my initial image. Below, you can find all of my masks applied all of the images.

Figure 29

Figure 30

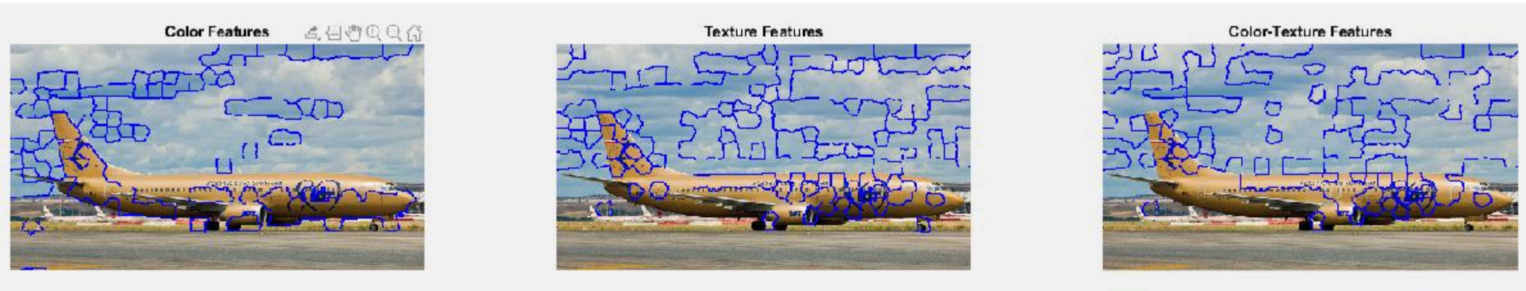

Figure 31

Figure 32

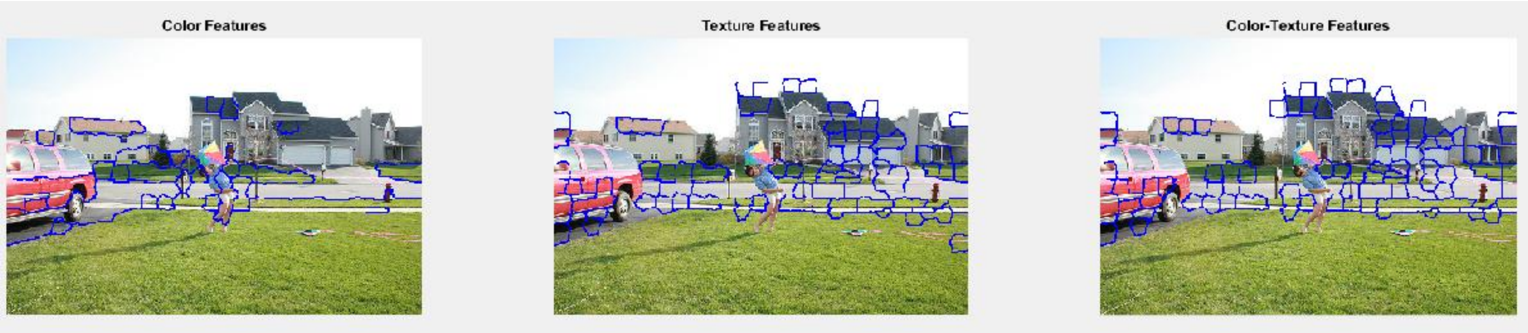

Figure 33

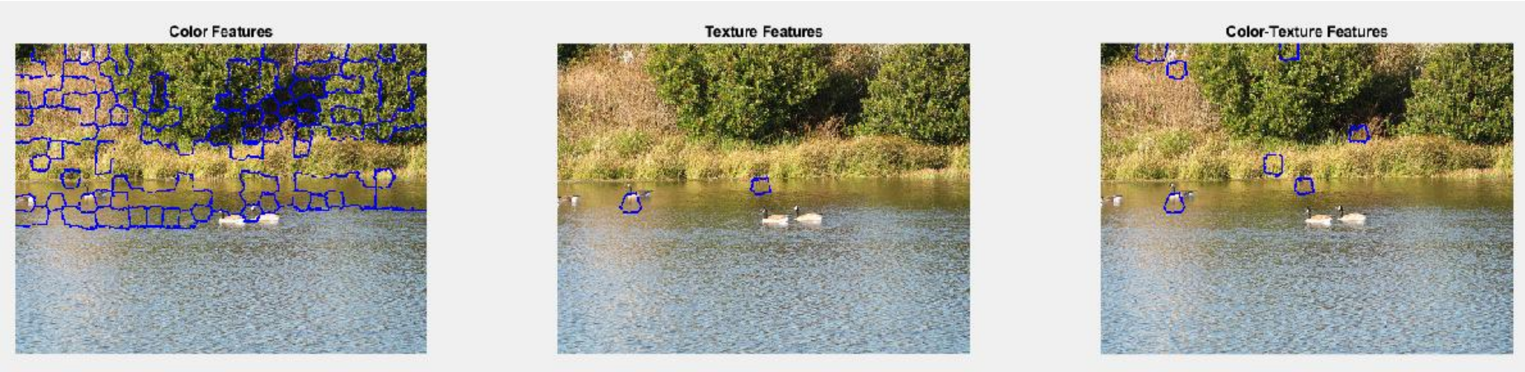

Figure 34

Figure 35

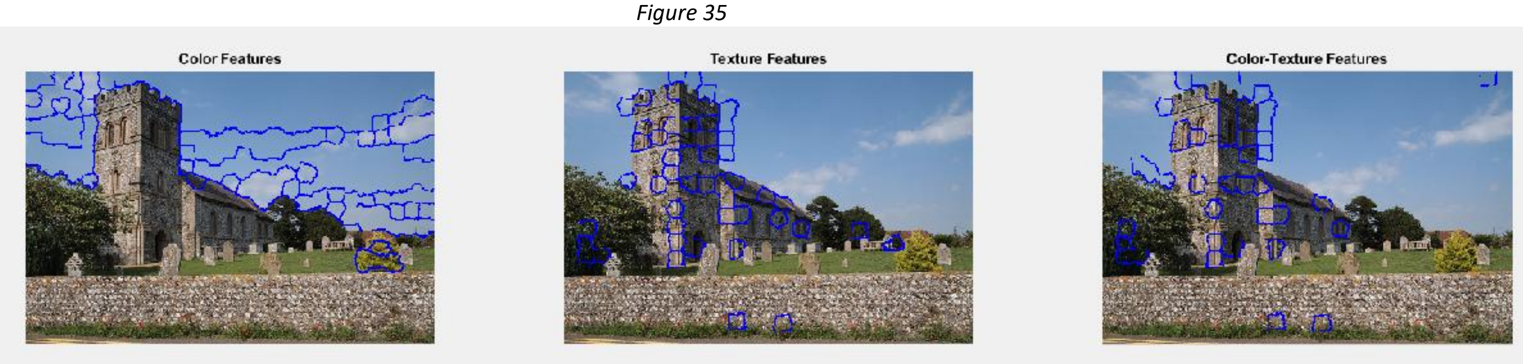

Figure 36

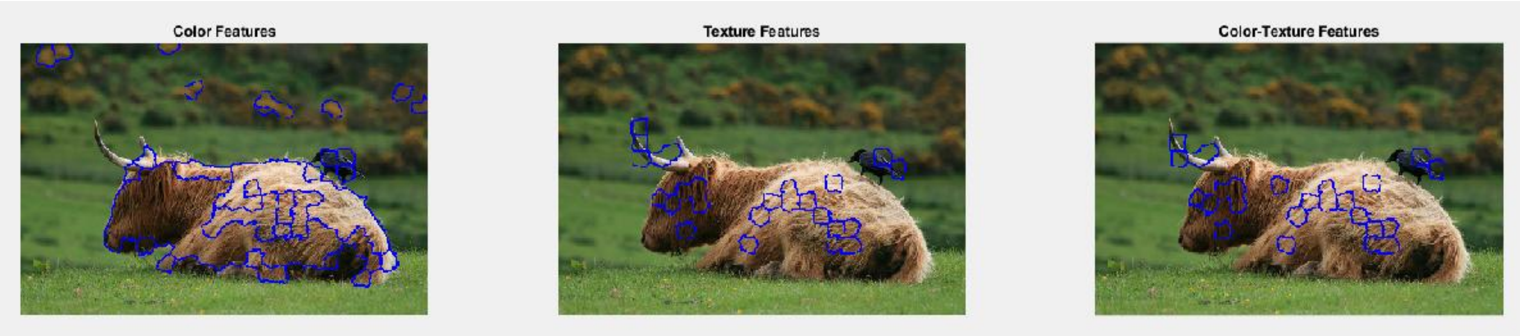

Figure 37

Figure 38

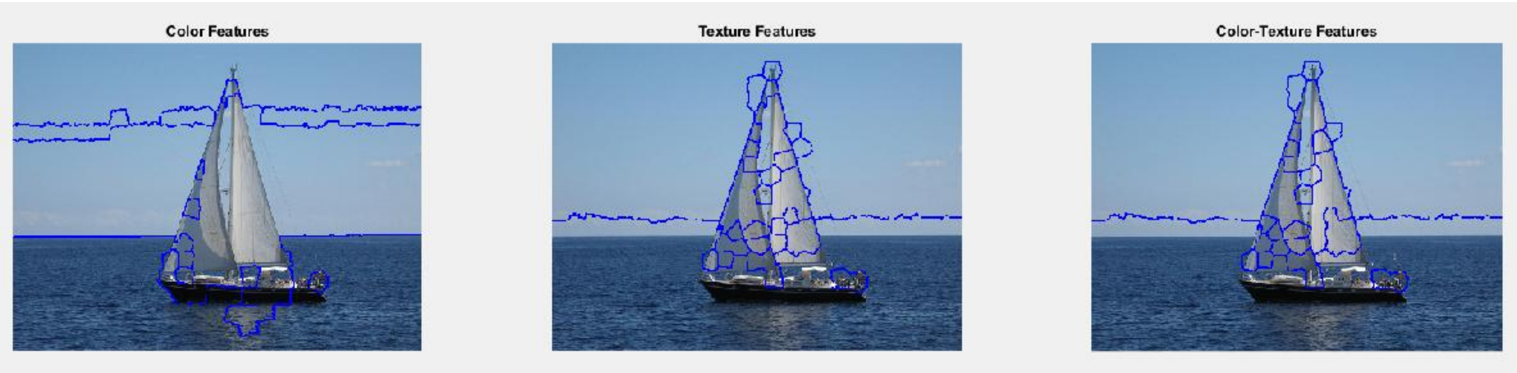

Figure 39

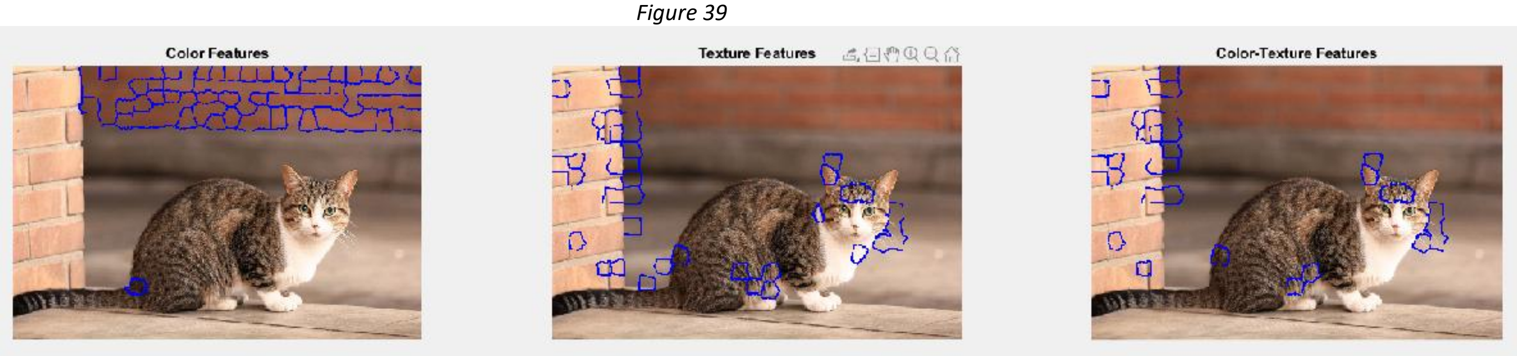

Figure 40

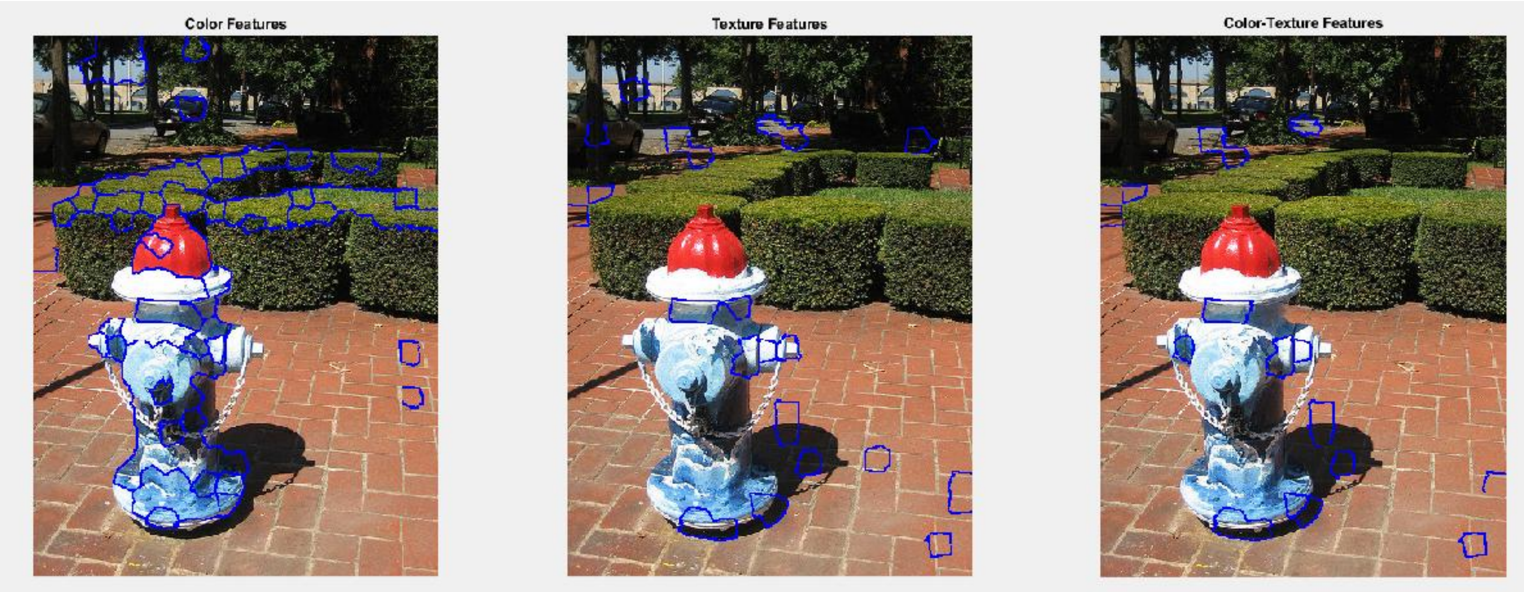

Figure 41

Figure 42

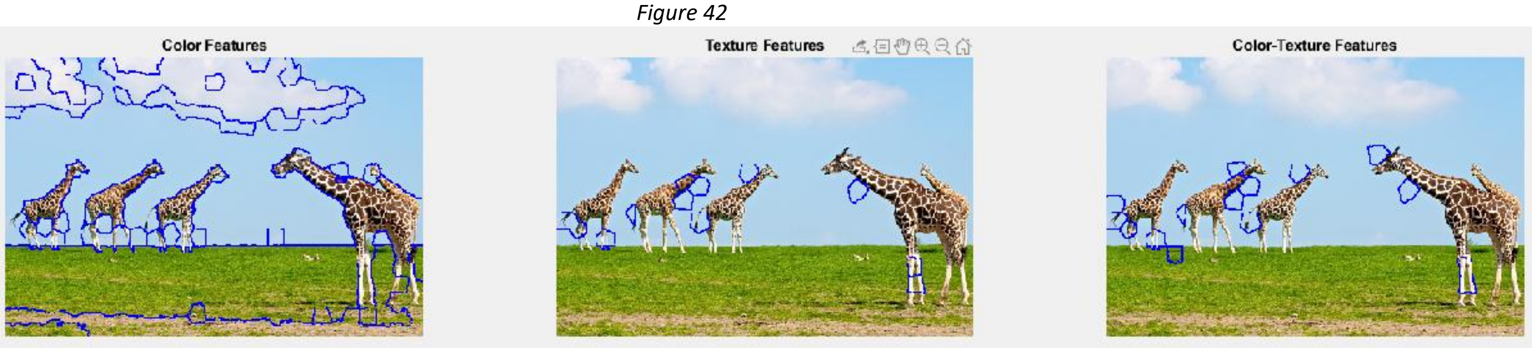

Figure 43

Figure 44

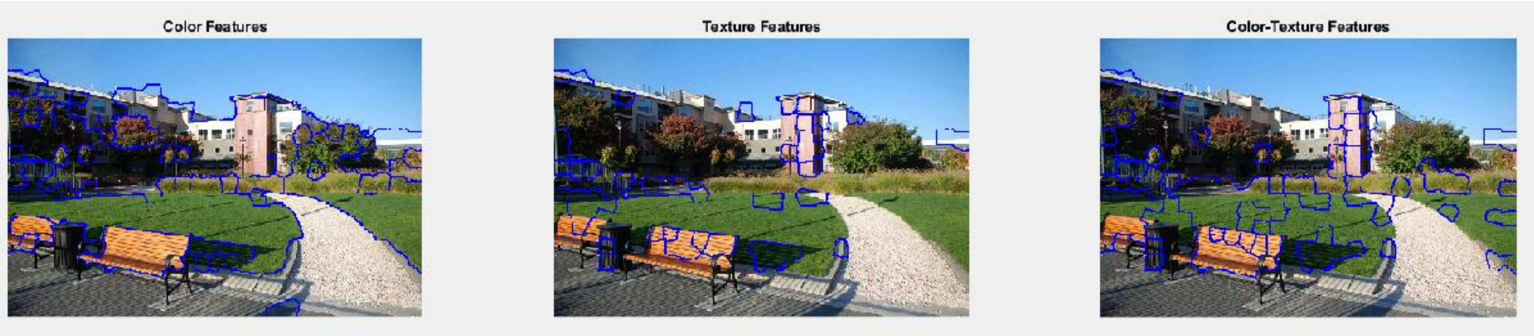

Figure 45

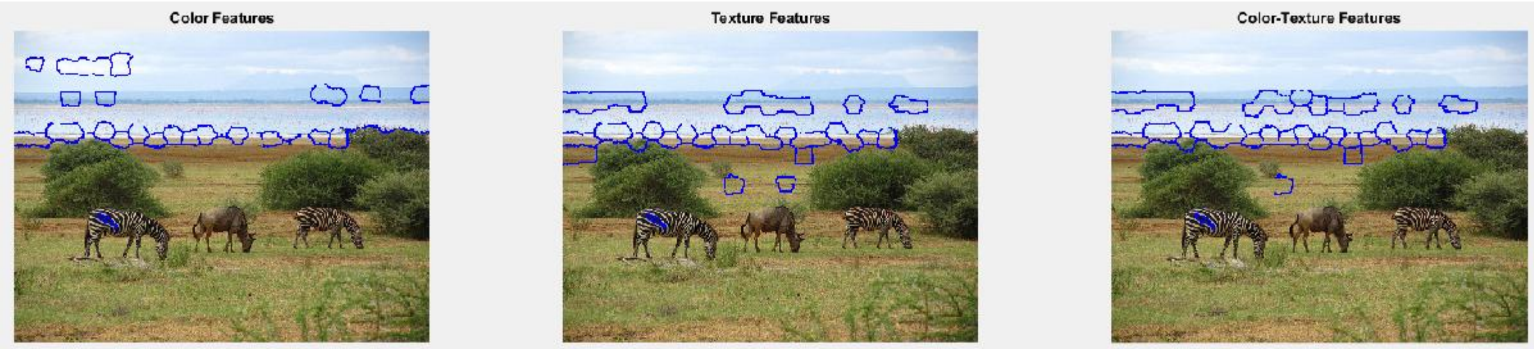

Figure 46

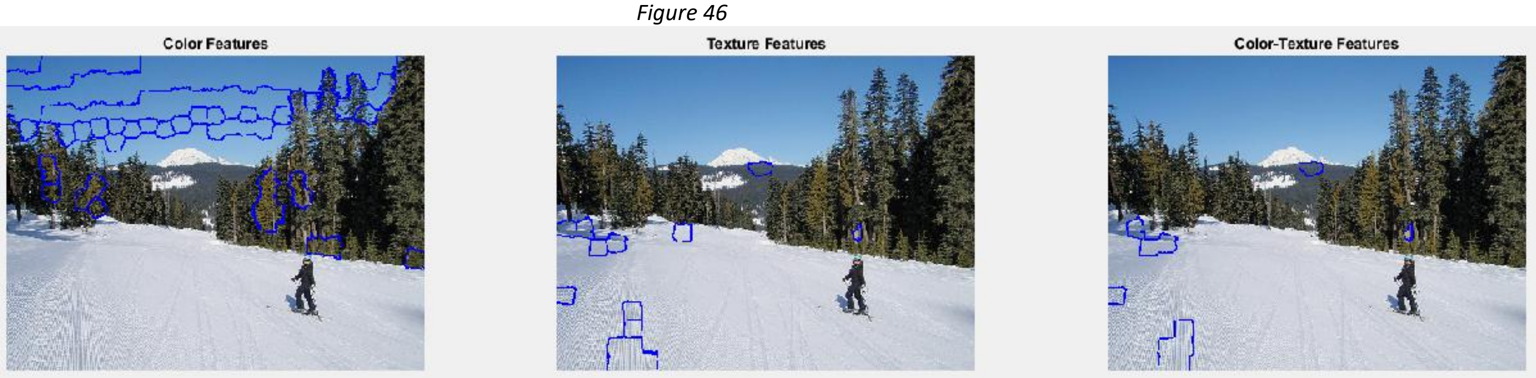

Figure 47

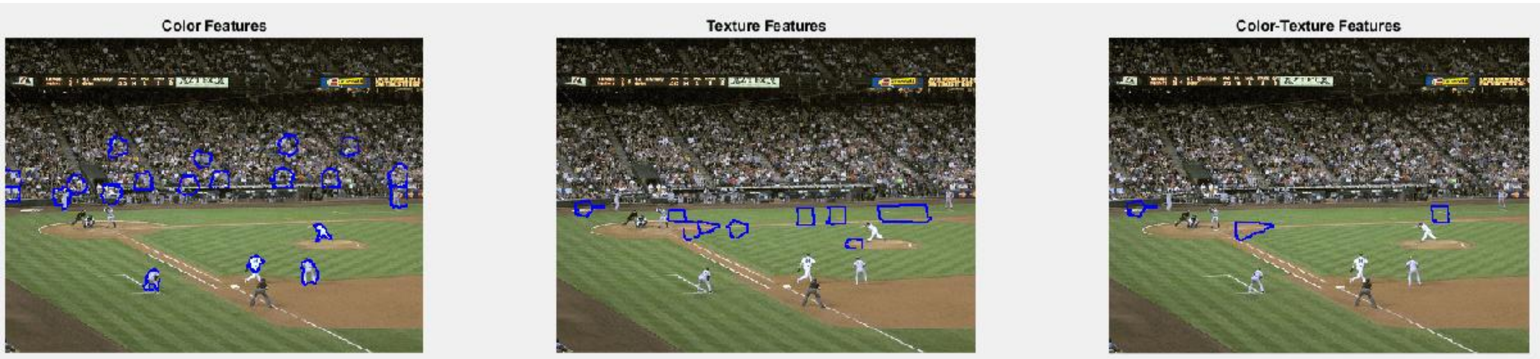

Figure 48

Figure 49

Discussion

As I mentioned while describing the functions, I could not find a good threshold for all of the images. For the images that have a clearly distinguishable colour difference, the colour filtering was working fine. In the images such as there are sky and grass below, it put a good line between them such as Figure 45. However, in the images like red bus (Figure 29), since the bus is red and there are also some different colours like windows, it could not put a good line between bus and windows, or in Figure 44, since there are lots of waves in the sea, and we are taking the mean of the colours, probably the mean changed because of the colour of the waves and colour mask was not good enough. It could have been better if I select my number of pixels higher, but then I thought I could also lose some borders if I would do that. Therefore, I did not change the number of superpixels. For Gabor filtering, it also put a good border in lines that it can detect, however, since it cannot differentiate colours, in some images that colour filtering was good, Gabor filtering was not good. I think colour filtering was overall better in my implementation. Moreover, since there are 16 variables in the Gabor mask, I could not understand if I did the implementation correct. There were sixteen variables in the Gabor filtering, so I think it lost many details while I calculated the distance. I think if I would just apply Gabor filtering in four scales and one orientation such as 0, The borders for that angle would be much better. However, since I combined all the angles, and calculated their distances together, I lost many details. For the colour-texture filter, since it is the combined version of both colour and Gabor, it was better than Gabor. However, I think since three average colour value distanced with sixteen Gabor value, colour means also lost many details and it was worse than the colour filter. You can see from the example results that there was not much difference between the texture and colour-texture mask since texture values have a higher number. According to my observation from my results, the colour mask was nearly always better than the others. This could be because of my implementation, but I could not find any better result. Also, my results generally were not that good, so there could be something wrong with my implementation.