**ENGN 2605 Image Understanding**

**Lab 03**

**Instructor: Prof. Benjamin Kimia**

**Name: Kuan-Min Lee**

**Student ID: 140481859**

**Part I: Edge Detection**

**Problem 1. Types of Edges**

**Reflectance Edges:**

Reflectance edges happen in the scenario where the surface stays the same but due to the difference of lightness reflected by different angles, there happens an edge formation. The following regions can be illustrated as the examples of that:

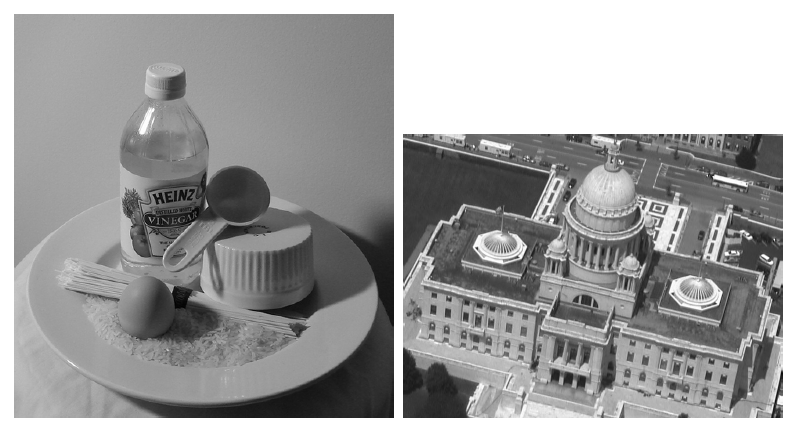


Figure 1: Illustration of Reflectance Edges

**Texture Edge:**

Texture edges happen in the scenario where the surface stays the same but varies with different textures. An edge will be formed in the interface of that:

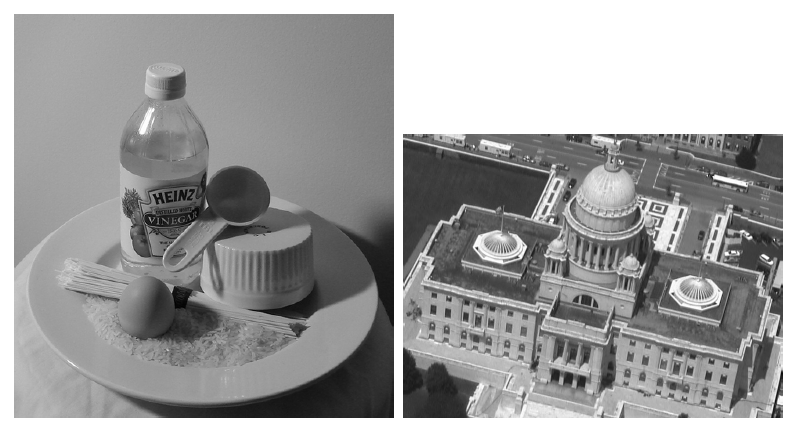


Figure 2: Illustration of Texture Edges

**Highlight Edges:**

Highlight Edges usually happen in a relative smooth surface where at certain angle, the light gets concentrated in a certain region. The following shows the region where highlight edges happen:

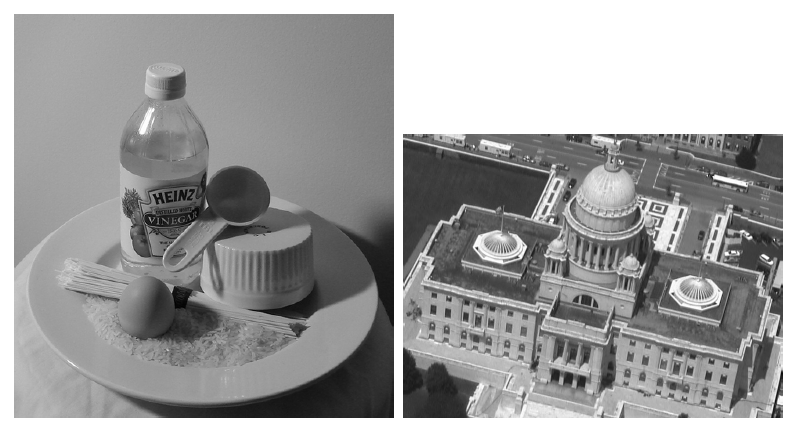


Figure 3: Illustration of Highlight Edges

**Shadow Edges:**

Usually happens in the back of an object, a shadow edges happen where the shadow happens. The following is the illustration of this type of edges:

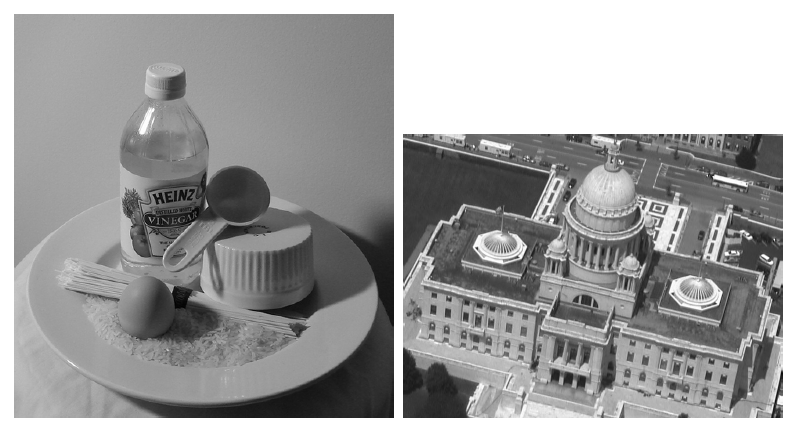


Figure 4: Illustration of Shadow Edges

**Problem 2 Intensity Based Edge Detection**

For this portion of the experiment, an intensity-based edge detector was implemented. It follows a very similar work flow as the widely known Canny Edge Detector. The input image is first input into a gaussian filter to smooth out the noise. Then, the gradient of each pixel is calculated and the variety in x and y direction is used to identify the direction of the gradient. In the very last step, non-max suppression is conducted is keep only the direction with the maximum magnitude of gradient.

For the setting of the function, the threshold is set as 3.5, and the following figures are the results:

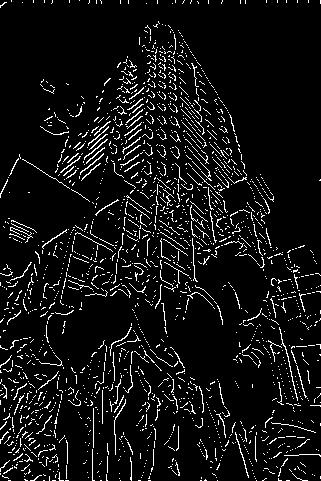


Figure 5: Comparison of Original Image and Edge Detection Image for figure 1



Figure 6: Comparison of Original Image and Edge Detection Image for figure 2



Figure 7: Comparison of Original Image and Edge Detection Image for figure 3



Figure 8: Comparison of Original Image and Edge Detection Image for figure 4

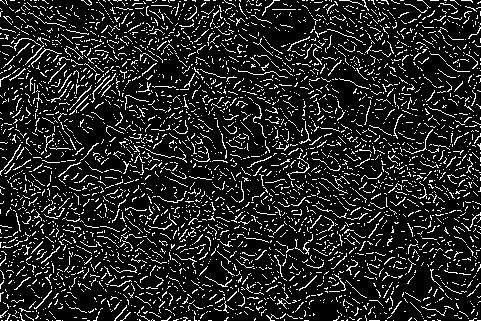


Figure 9: Comparison of Original Image and Edge Detection Image for figure 5



Figure 10: Comparison of Original Image and Edge Detection Image for figure 6

**Problem 3 Canny Edge Detector**

For this part of the experiment, the built-in MATLAB Canny Function is executed and tested with the edge detection algorithm built in the previous section. To get the best result, the threshold and sigma for canny is set to be 0.15 and 1 respectively. The following are the results:

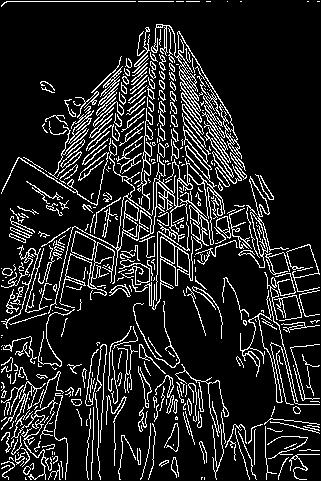
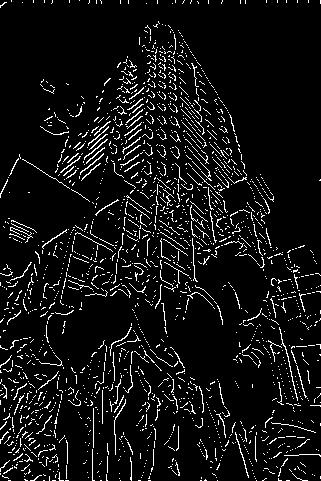


Figure 11: Comparison of Original Image and Edge Detection Images for figure 1

(Left: Origin, Middle: Intensity-Based Edge Detector, Right: Canny)



Figure 12: Comparison of Original Image and Edge Detection Images for figure 2

(Left: Origin, Middle: Intensity-Based Edge Detector, Right: Canny)

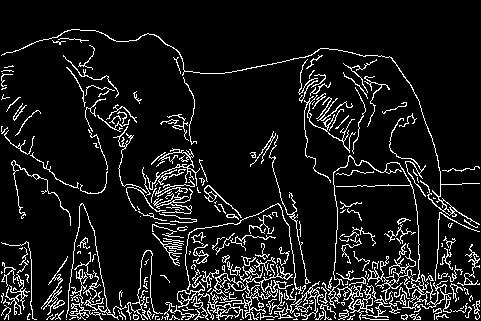


Figure 13: Comparison of Original Image and Edge Detection Images for figure 3

(Left: Origin, Middle: Intensity-Based Edge Detector, Right: Canny)



Figure 14: Comparison of Original Image and Edge Detection Images for figure 4

(Left: Origin, Middle: Intensity-Based Edge Detector, Right: Canny)

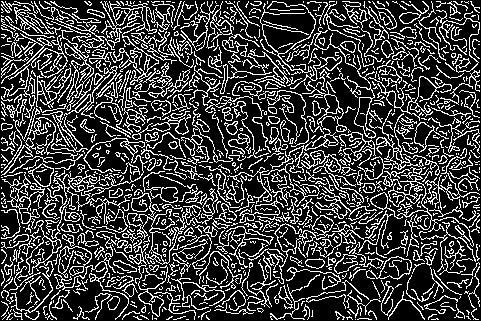
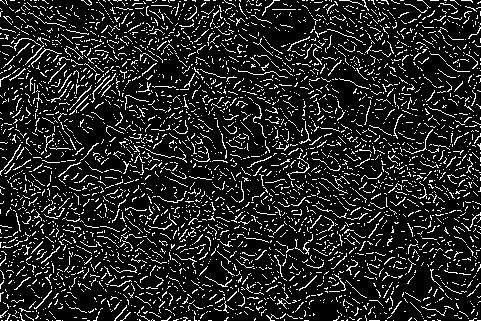


Figure 15: Comparison of Original Image and Edge Detection Images for figure 5

(Left: Origin, Middle: Intensity-Based Edge Detector, Right: Canny)

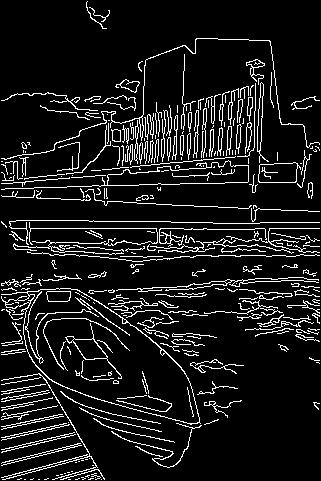


Figure 16: Comparison of Original Image and Edge Detection Images for figure 6

(Left: Origin, Middle: Intensity-Based Edge Detector, Right: Canny)

From the above figures, it’s pretty obvious to say that Canny has the strength of recovering the detail part of the image, such as the windows in figure 1, strips on T-shirt in figure 2, and strip on zebras in figure 3. But also due to this, Canny seems to be more sensitive to noises. This can be seen in the grass area of figure 4 and water portion in figure 6.

For the most detected edge types, it seems that occlusion edges get detected that most. From figure 1, the edge between the sky and building is detected by both algorithms. From figure 2, the edge on the right neck is also detected by both of them. From figure 3, the edges between the elephants and the background sky are the most obvious ones. For figure 4, the edges between the third zebra on the right is also detected by both algorithms. And for figure 6, it’s the portion between the boat and the river.

The most special case here is figure 5, whose edges are mostly texture or reflectance edges, which contains lots of details inside the image.

**Problem 4 Histogram Based Edge Detection**

This portion of the experiment is about implementing an algorithm that group the image into several circular regions, divide the region into upper and lower half and generate histograms for both regions and compare the difference between the two histograms.

For the best outcomes, the best outcome is by the setting of radius =3 and num\_bins = 16:

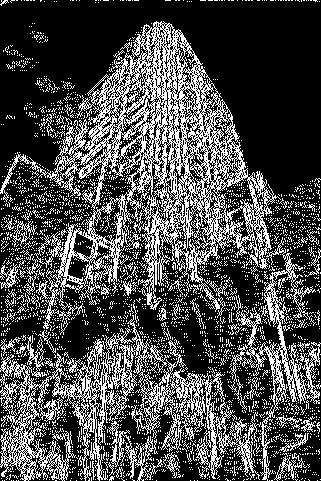


Figure 17: Comparison of Original Image and Edge Detection Images for figure 1

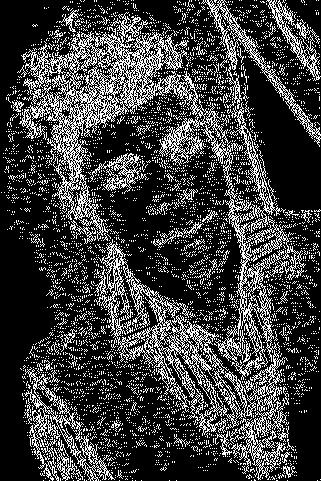


Figure 18: Comparison of Original Image and Edge Detection Images for figure 2

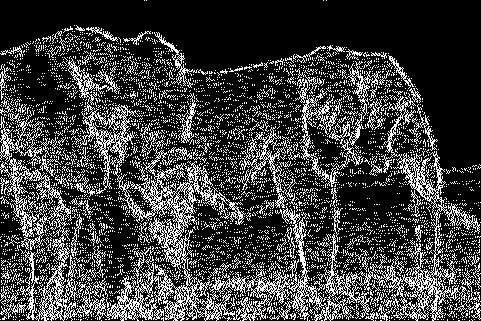


Figure 19: Comparison of Original Image and Edge Detection Images for figure 3

A group of zebras in a field

Description automatically generated with medium confidenceA picture containing plant

Description automatically generated

Figure 20: Comparison of Original Image and Edge Detection Images for figure 4

A picture containing reptile, rock, lizard, outdoor

Description automatically generatedA picture containing nature, outdoor

Description automatically generated

Figure 21: Comparison of Original Image and Edge Detection Images for figure 5

A boat on the water

Description automatically generated with low confidenceA picture containing outdoor

Description automatically generated

Figure 22: Comparison of Original Image and Edge Detection Images for figure 6

**Problem 5 Texture Based Edges**

**Problem 6 Edge Linking**

This part of the experiment follows similar concept from problem 2. Except for the part which we generated two different edge maps for two different thresholds and use them as a criterion to generate the final outcomes. For this section, a high threshold of 8.5 and low threshold of 7.0 are set. The following are the results of the final outcomes:



Figure : Comparison of Original Image and Edge Linking Result for Figure 1



Figure : Comparison of Original Image and Edge Linking Result for Figure 2

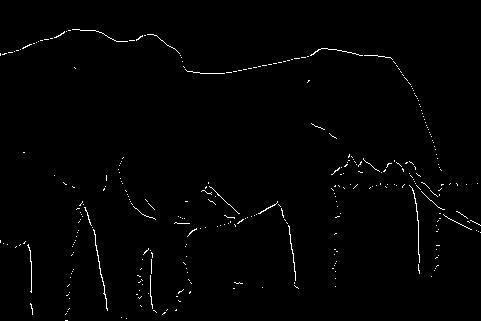


Figure : Comparison of Original Image and Edge Linking Result for Figure 3



Figure : Comparison of Original Image and Edge Linking Result for Figure 4

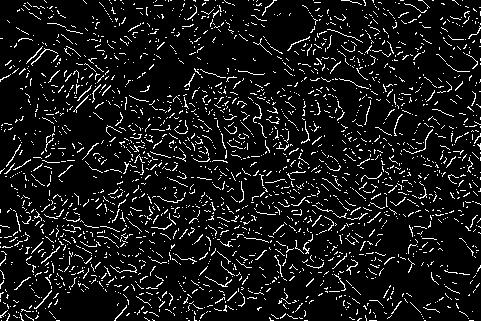


Figure : Comparison of Original Image and Edge Linking Result for Figure 5



Figure : Comparison of Original Image and Edge Linking Result for Figure 6