

## Homework #2

Issued: 01/31/2024

Due: 02/19/2024

### Problem 1: Edge Detection

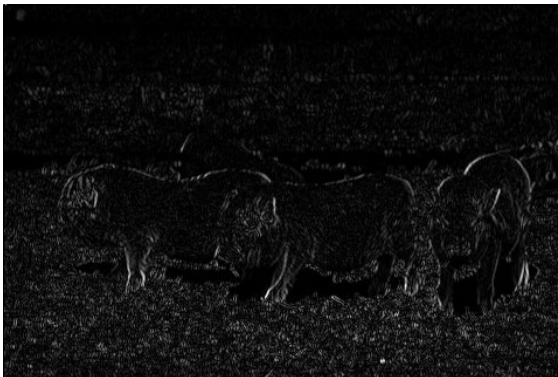
#### (a) Sobel Edge Detector



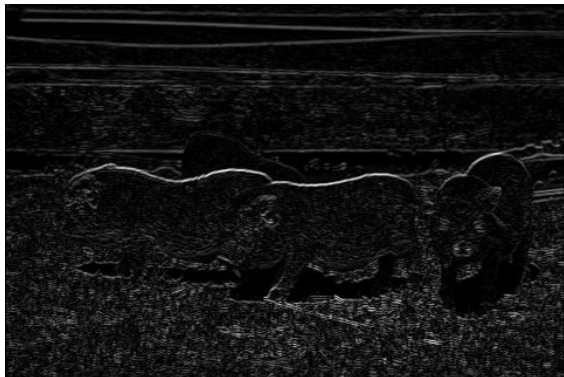
(a) Tiger X-Gradient



(b) Tiger Y-Gradient

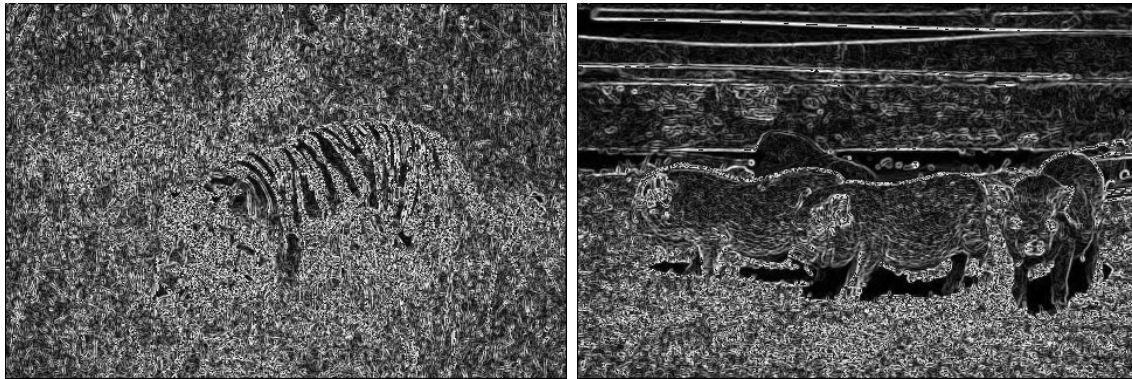


(c) Pig X-Gradient



(d) Pig Y-Gradient

Figure 1: (1) Normalized Results of X and Y gradients



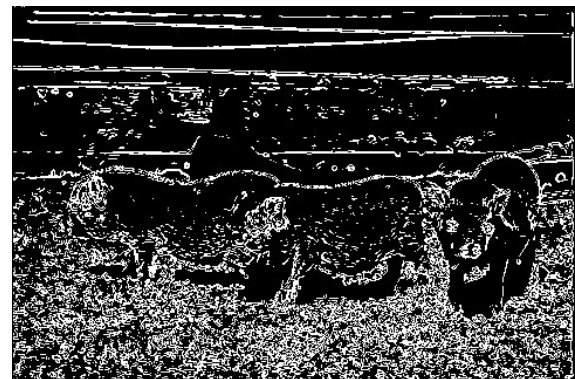
(a) Tiger Gradient Magnitude Map

(b) Pig Gradient Magnitude Map

Figure 2: (2) Normalized Gradient Magnitude Maps



(a) Tiger Edge Map with Threshold of 90%

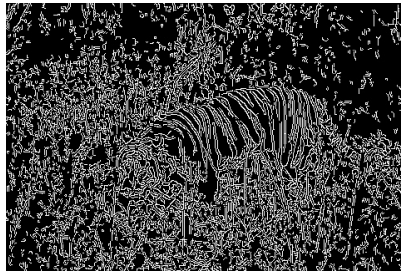


(b) Pig Edge Map with Threshold of 80%

Figure 3: (3) Edge Maps with Threshold

## (b) Canny Edge Detector

- (1) Non-maximum suppression in the Canny edge detector is an approach to find edges. First we search through all pixels in the image and find those points which are higher than a high threshold, and label them as edges. Then pixels near the determined edges are checked, those higher than a lower threshold are also labeled as edges.
- (2) High Threshold are used to determine the most significant edges. Low Threshold can determine how much the edges expand.
- (3) Show in Figure with label figure 4 and figure 5. We can see that when higher high threshold is applied, only more significant pixels with larger gradients will be classified as primary edges, which makes less false positives, but increases the chance of false negative. And when lower high threshold is applied, more points will be classified as primary edge pixels, increasing more environment point that are wrongly classified as edges. And a higher low threshold will decrease how much each edge expand. When similar high thresholds are applied, pictures with higher low threshold will have shorter edges expand from the basic edge points.



(a) Threshold 120/220

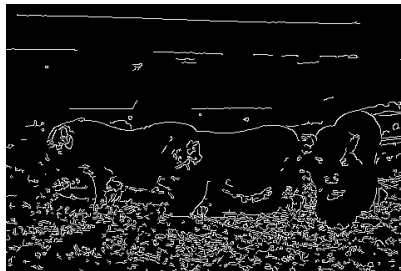


(b) Threshold 200/300



(c) Threshold 180/600

Figure 4: Tiger Canny Edge Map



(a) Threshold 150/300



(b) Threshold 200/400



(c) Threshold 180/600

Figure 5: Pig Canny Edge Map

## Problem 2: Digital Half-toning

### (a) Dithering



Figure 6: Lighthouse Image with Fixed Thresholding



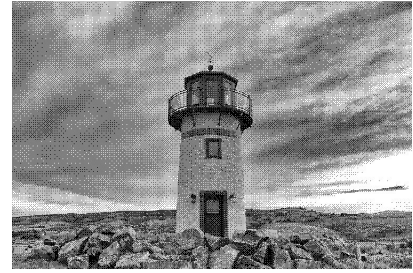
Figure 7: Lighthouse Image with Random Thresholding



(a) Processed Image with  $I_2$  Dithering Matrix



(b) Processed Image with  $I_8$  Dithering Matrix

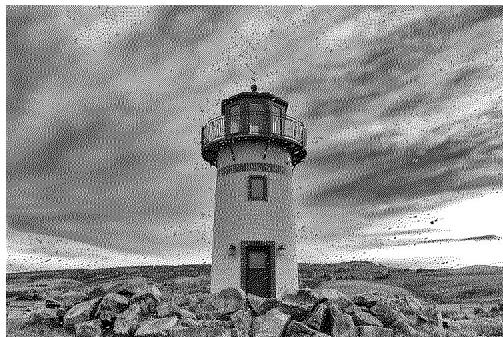


(c) Processed Image with  $I_{32}$  Dithering Matrix

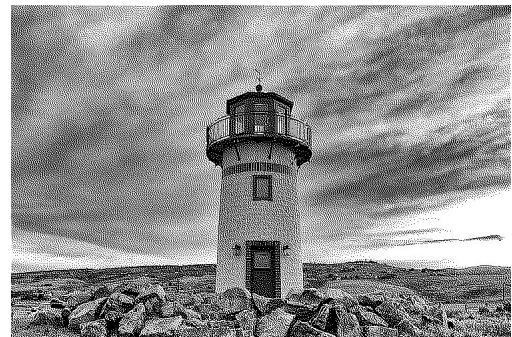
Figure 8: Pig Canny Edge Map

(3 )The  $I_2$  matrix results in coarse dithering with more pronounced patterns and potential loss of detail. Transitioning to  $I_8$  provides smoother halftoning and better detail preservation, suitable for higher resolutions. However, while  $I_{32}$  can provide better halftone quality than  $I_8$ , in the case of Lighthouse.raw, the improvement becomes less noticeable.

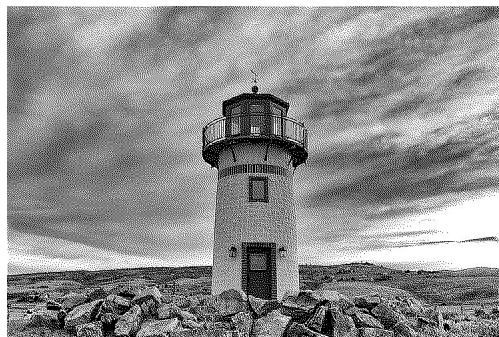
## (b) Error Diffusion



(a) Processed Image with Floyd-Steinberg's error diffusion



(b) Processed Image with Jarvis, Judice, and Ninke (JJN)



(c) Processed Image with Error diffusion proposed by Stucki

Figure 9: Error Diffusion with three different Matrix Methods



In this case, I prefer the method of stucki and JJN, which keeps most details of the original image, and with almost no noise. The method of floyd keeps also good details but brings a little random noises to the image. The fixed thresholding method lost lots of details, and the random thresholding method brings severe salt&pepper noises. And Dithering Matrix's result is not as good as error diffusion in terms of contrasts and other details.

I have an idea of pre-processing local enhancement processing, such as edge enhancement and contrast amplification, should be applied to highlight main features before employing the error diffusion technique. This approach would prevent the loss of details and minimize noise introduction. Additionally, further processing of the halftoned image post error diffusion, like applying median filtering, could remove isolated noise points. For instance, the Floyd method may generate isolated noise which can be mitigated with this technique.

(4) Raster scanning processes an image line by line from one side to the other, makes the processed image with unnatural strips in horizontal direction, which is called directional artifacts. While serpentine scanning alternates direction every line, reducing such artifacts and evenly distributing errors for a more consistent image quality.

### Problem 3: Color Half-toning with Error Diffusion

#### (a) Separable Error Diffusion



Figure 10: half-toned color Bird image with Separable Error Diffusion

The most significant issue with this image is random noises. The combination of the 8 colors cannot form good enough natural colors for human's eyes.

### (b) MBVQ-based Error diffusion

- (1) The key idea of the MBVQ is that instead of directly using combination of 8 color channels, it uses a method that could sharply decrease the existence of K/W, which are on the two ends of brightness diagram. It turns each pair of K/W with other colors into closer color pairs in the brightness diagram. For example, K-W into M-G, K-Y into R-G, K-C into B-G, K-M into B-R, vice versa for pairs containing W. This could make colors more precisely in human eyes, since human vision is more sensible to brightness than color. Less combination of color pixels with high changes in brightness could decrease the number of noises and unnatural pixels, making the image more natural and with more saturation.
- (2) I met a wired problem that I could not find where it is.



Figure 11: half-toned color Bird image with MBVQ Diffusion