Name: Kuimu Ren
USC ID Number:1473482531
USC Email:kuimuren@usc.edu
Submission Date: 02/18/2024

Problem1

Abstract and Motivation:

Edge detection is a fundamental technique in image processing used to identify the boundaries of objects within images. The primary purpose of edge detection is to locate significant changes in intensity or color within an image, which often correspond to the boundaries of objects or changes in texture.

(a) Sobel Edge Detector

Approach and Procedures:

The Sobel edge detector is a popular method for detecting edges in digital images. It works by calculating the gradient magnitude of an image, which highlights areas of rapid intensity change.

First, convert the RGB image to grayscale.

$$\mathrm{Y} = 0.2989 * \mathrm{R} + 0.5870 * \mathrm{G} + 0.1140 * \mathrm{B}$$

Second, apply Sobel kernels to the grayscale image to calculate the gradient in both the x and y directions.

$$sobel\ X = \{\{-1, 0, 1\}, \{-2, 0, 2\}, \{-1, 0, 1\}\}$$
$$sobel\ Y = \{\{1, 2, 1\}, \{0, 0, 0\}, \{-1, -2, -1\}\}$$

Third, compute the gradient magnitude at each pixel using the results from the Sobel X and Sobel Y convolutions.

$$Gradient\ Magnitude\ =\ sqrt((Sobel\_X)\text{^}2\ +\ (Sobel\_Y)\text{^}2)$$

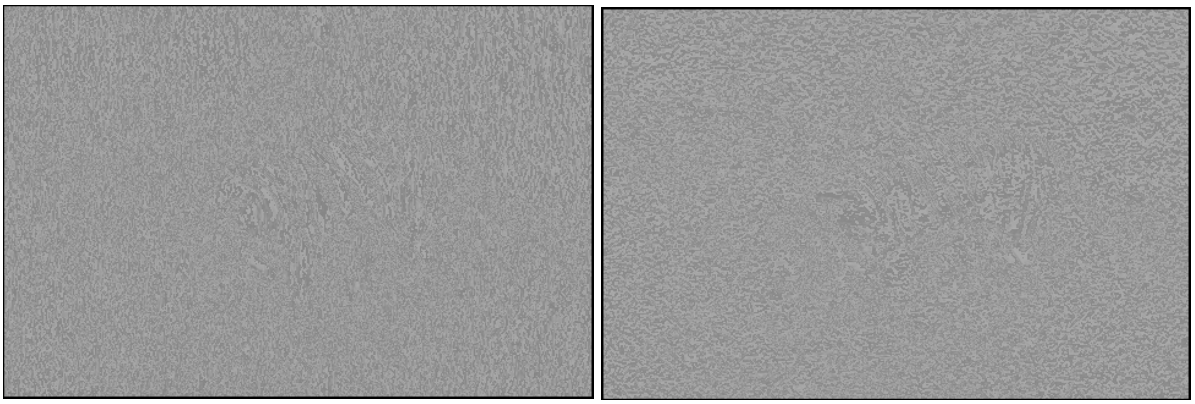
Fourth, normalize the gradient magnitude values to ensure they fall within an appropriate range, typically mapping the magnitude values to the range [0, 255].

$$normalized\ gradient = \frac{currentGradient - minGradient}{maxGradient - minGradient}$$

Last, Apply a threshold to the normalized gradient magnitude image to identify significant edges and output the image.

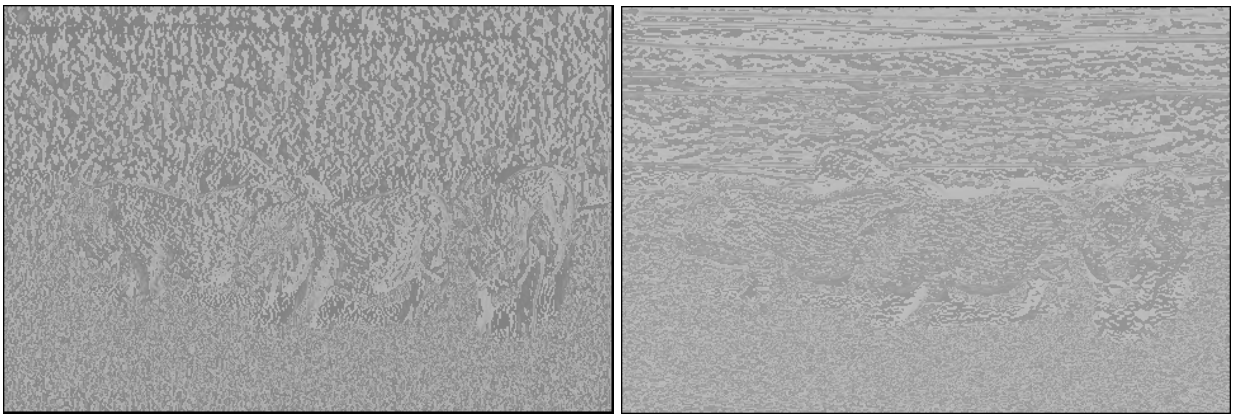
Experimental Results:

Shown below are the results of Problem 1(a):



(a) (b)

Figure1: (a)x-gradient of Tiger image and (b)y-gradient of Tiger image



(a) (b)

Figure2: (a)x-gradient of Pig image and (b)y-gradient of Pig image

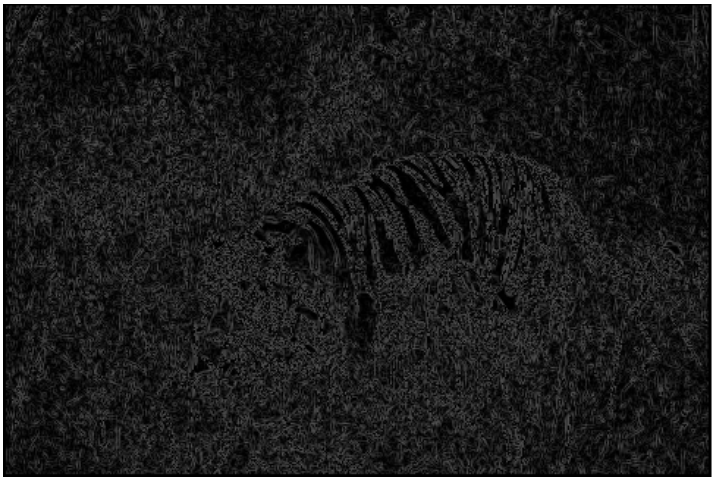


Figure3: normalized gradient magnitude map of Tiger image

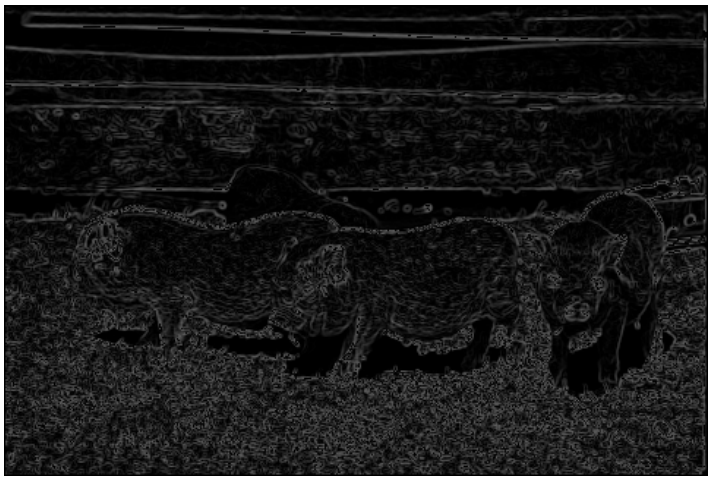


Figure4: normalized gradient magnitude map of Big image

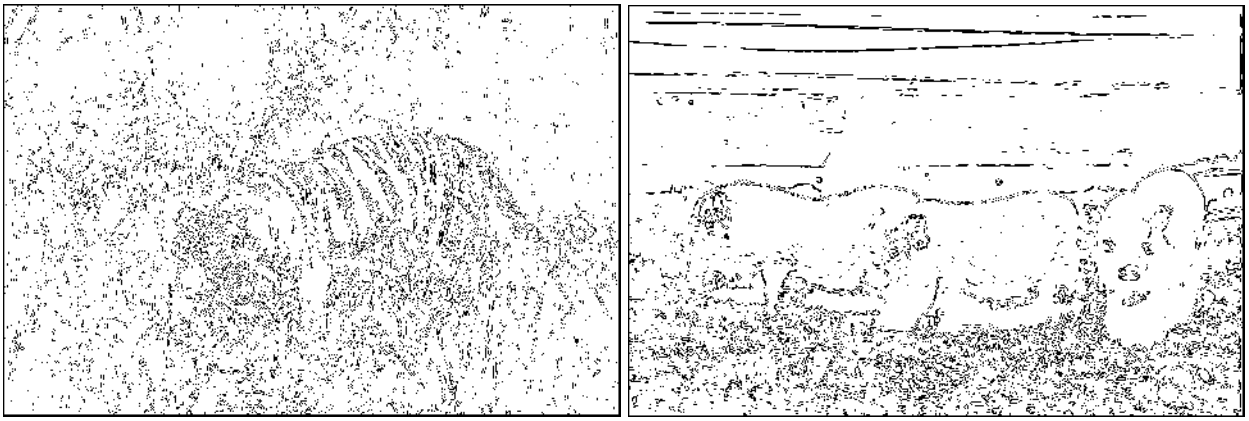


Figure5: edge map of Tiger image and Big image with threshold(80%)

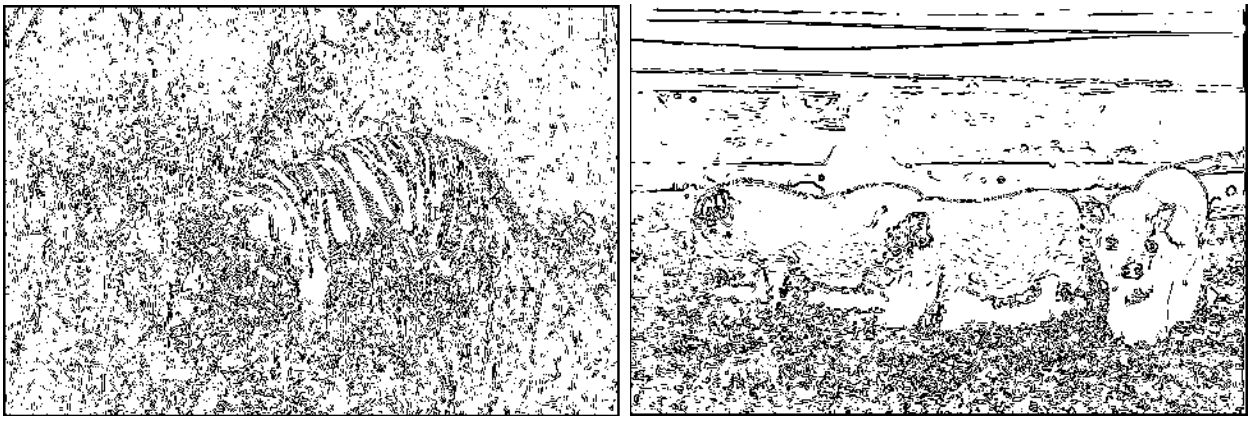


Figure6: edge map of Tiger image and Big image with threshold(85%)

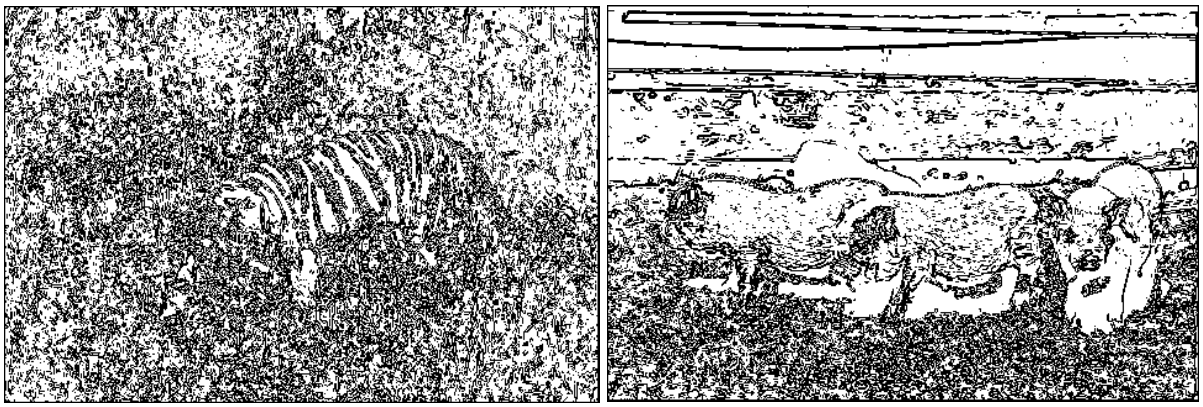


Figure7: edge map of Tiger image and Big image with threshold(90%)

Homework Answer and Discussion:

(1) Figure1(a) is the result of tiger image which normalize the x-gradient values to 0-255
Figure1(b) is the result of tiger image which normalize the y-gradient values to 0-255
Figure2(a) is the result of pig image which normalize the x-gradient values to 0-255
Figure2(b) is the result of pig image which normalize the y-gradient values to 0-255

(2) Figure3 is the result of normalized gradient magnitude map of tiger image
Figure4 is the result of normalized gradient magnitude map of pig image

(3) From Figure5,6,7 we can find that the image with threshold(90%) has too many cluttered lines, and that image with threshold(80%)has relatively little edge information. So I think tune the threshold(85%) can obtain an edge map with the best visual performance.

(b) Canny Edge Detector

Approach and Procedures:

First, the input image is smoothed using a Gaussian filter to reduce noise.

Second, on the smoothed image, gradients of the image intensity are calculated using operators like the Sobel operator. This helps in finding the magnitude and direction of gradients in the image.

Third, Non-maximum suppression is applied to the gradient image to refine the edges.

Fourth, Pixels in the gradient image are categorized into strong edges, weak edges, and non-edges based on two thresholds (high and low).

Last, Edge tracking and hysteresis are used to further refine the edges. This step involves linking strong edge pixels with neighboring weak edge pixels to form complete edges. Typically, weak edge pixels are traced in the pixel neighborhood, and based on their connectivity with strong edge pixels, it is decided whether they should be retained as edges.

Experimental Results:

Shown below are the results of Problem 1(b):

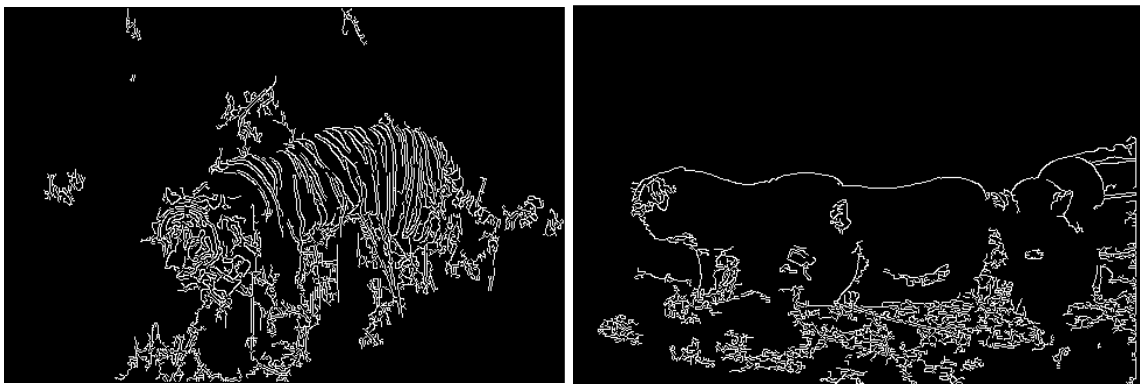


Figure8: Tiger image and Pig image by canny edge detector

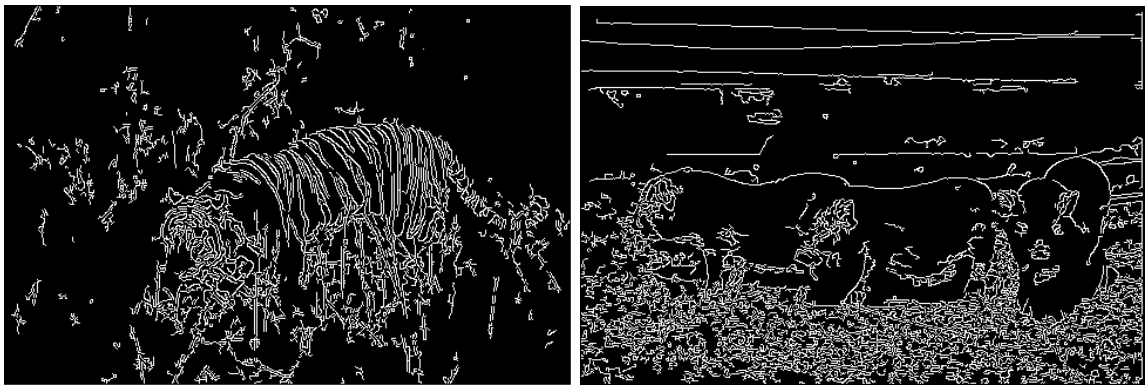


Figure9: Tiger image and Pig image by canny edge detector

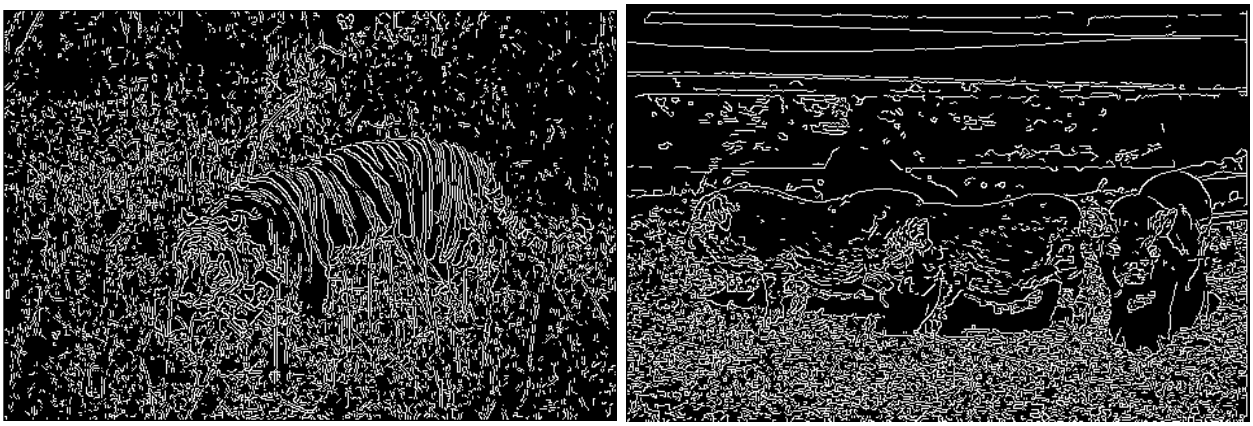


Figure10: Tiger image and Pig image by canny edge detector

Homework Answer and Discussion:

(1) Using non-maximum suppression's purpose is to refine detected edges, making them clearer and more precise.
Specifically, for each detected edge pixel, the non-maximum suppression algorithm examines the gradient direction in its neighborhood. if a pixel has a higher intensity value than its neighbors along the gradient direction, it remains unchanged; otherwise, it's considered non-maximum and gets suppressed.

The aim of non-maximum suppression is to enhance the detected edges by emphasizing only the strongest responses and thinning out the edges, thus producing a more accurate representation of the edges in the image.

(2) The high threshold is used to identify strong edge pixels in the gradient magnitude image. Any edge pixel with a gradient magnitude higher than the high threshold is considered a strong edge pixel and is retained in the final edge map.
The low threshold is employed to identify weak edge pixels. These are edge pixels with a gradient magnitude lower than the high threshold but higher than the low threshold. These pixels are considered edge pixels.
Then, after edge pixels above the high threshold are definitely retained as strong edges. Edge pixels below the low threshold are discarded. However, those falling between the low and high thresholds are subject to further scrutiny. We need to examine weak edge pixels and connecting them to strong edge pixels to form continuous edges. Only weak edge pixels connected to strong edge pixels are retained in the final edge map.

(3) In figure8, tiger image's high threshold(using gradient) is 100, and low threshold is 600
In figure8, pig image's high threshold(using gradient) is 100, and low threshold is 500
In figure9, tiger image's high threshold(using gradient) is 150, and low threshold is 400
In figure9, pig image's high threshold(using gradient) is 150, and low threshold is 300
In figure10, tiger image's high threshold(using gradient) is 150, and low threshold is 200
In figure10, pig image's high threshold(using gradient) is 100, and low threshold is 150
From the result, we can find that if the low threshold is set too high, many weak edges may be discarded, resulting in fewer edges being detected and there are a lot of intermittent lines. If the low threshold is set too low, too many weak edges may be retained, possibly including noise and there are many continuous lines. If the high threshold is set too high, many useful edges may be discarded because their gradient magnitudes are not sufficient to meet the high threshold requirement. This can lead to a lack of important edges in the edge detection results. If the high threshold is set too low, this can result in the edge detection results containing many useless edges.

(c) Structured Edge
Approach and Procedures:

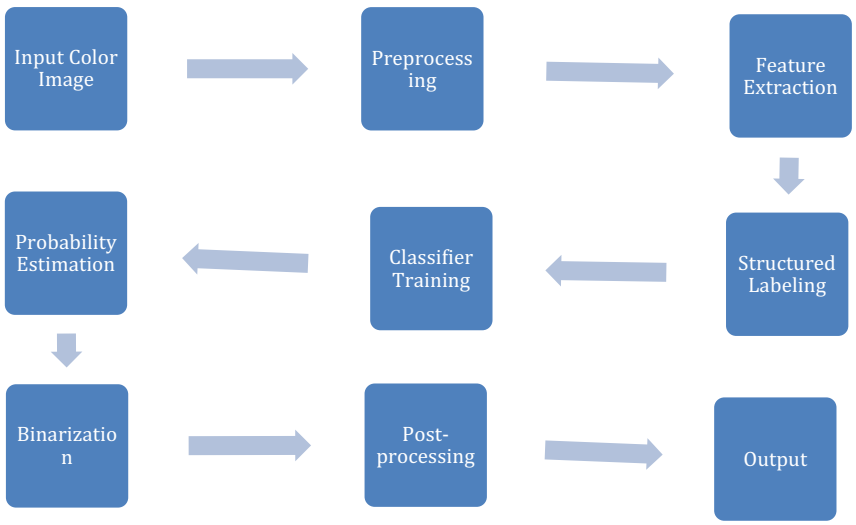


Figure11: flow chart of SE detection algorithm

Experimental Results:

Shown below are the results of Problem 1(b):

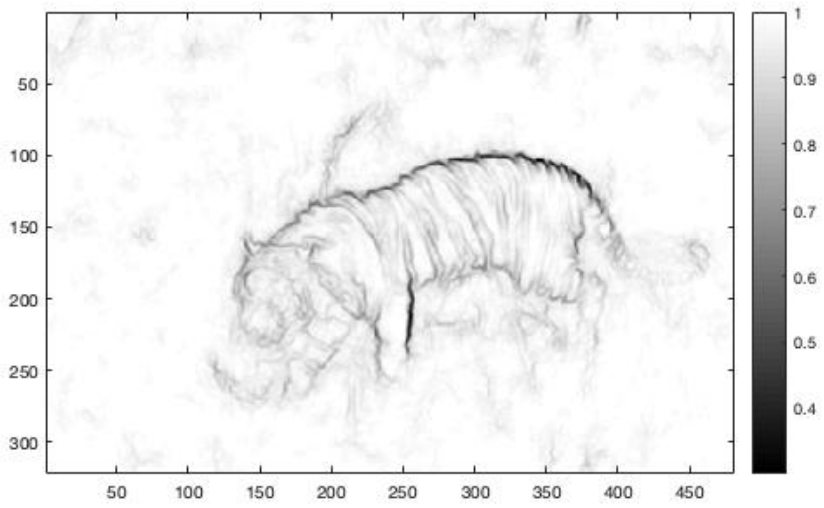


Figure12: Probability edge map of tiger image

Figure13: Binary edge map (with p>0.92) of tiger image

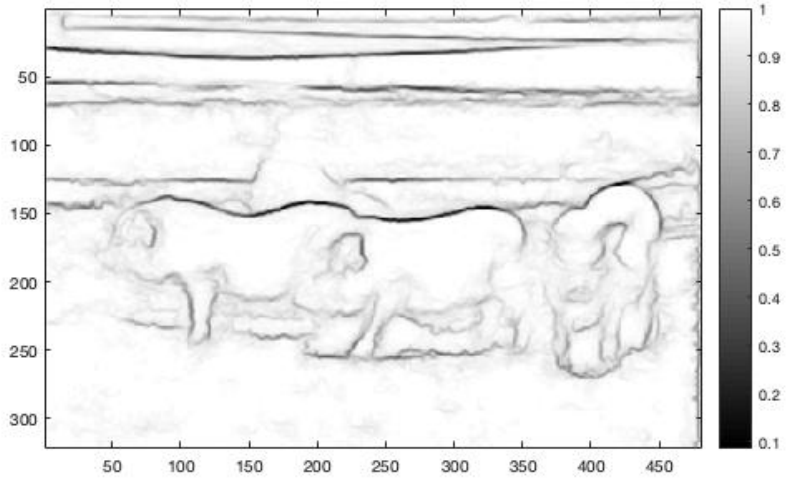


Figure14: Probability edge map of pig image

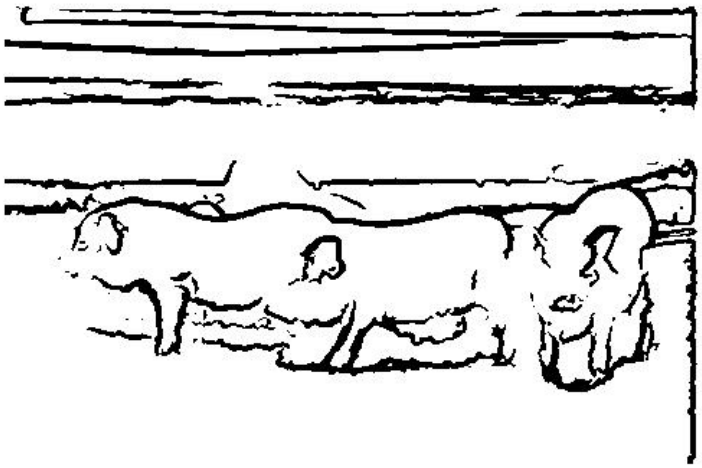


Figure15: Binary edge map (with p>0.88) of pig image

Homework Answer and Discussion:

(1) Figure11 shows the flow chart of SE detection algorithm.

1.Input Color Image: Take a color image as input.

2. Preprocessing: Optionally, perform any necessary preprocessing steps such as resizing or color space conversion.

3. Feature Extraction: Extract features from the input image, including gradient magnitude and orientation features, as well as self-similarity features.

4. Structured Labeling: Perform structured labeling to group similar pixels into coherent regions based on extracted features.

5. Classifier Training: Train a classifier using labeled patches from the input image and corresponding ground truth labels.
6. Probability Estimation: Estimate the probability of edge presence at each pixel location using the trained classifier.
7. Binarization: Binarize the probability map using a threshold to obtain a binary edge map.
8. Post-processing (optional): Optionally, apply post-processing techniques such as non-maximum suppression or morphological operations to refine the binary edge map.
9. Output: Output the final binary edge map highlighting the detected edges in the input color image.

(2) 1. Decision Tree Construction Process：
A decision tree is a non-parametric supervised learning method used for classification or regression based on feature values.
First, we need to choose the best feature as the splitting node.
Second, based on the selected feature, we need to divide the dataset into subsets.
Third, repeat the above process for each subset recursively until stopping criteria are met.
Last, when stopping criteria are met, label the current node as a leaf node and assign it a specific class for classification problems.

2. Random Forest (RF) Classifier Principle:
Random Forest is an ensemble learning method consisting of multiple decision trees.
First, randomly select a subset of samples from the training set to train each decision tree.
Second, for each node of each decision tree, randomly select a subset of features from the feature set to use for splitting.
Last, during classification or prediction, each decision tree independently classifies or predicts samples. The final classification result or predicted value is determined by aggregating the votes or averaging the predictions of all decision trees.

3. Influence of decision tree depth and number on prediction results:
The depth of decision trees affects the model's complexity and generalization ability. Deeper trees can better fit the training data but are more prone to overfitting. Shallower trees may underfit and fail to capture complex relationships in the data.
The number of decision trees in a Random Forest impacts the model's stability and accuracy. Increasing the number of trees typically improves performance and reduces the impact of randomness. However, having too many trees may increase computational cost and could lead to overfitting.

(3) Figure12 shows Probability edge map of tiger image.
Figure13 shows Binary edge map (with $p>0.92$) of tiger image
Justify selection: Because the grass contour is relatively small, we can choose a high threshold to highlight the tiger contour.
Figure14 shows Probability edge map of pig image.
Figure15 shows Binary edge map (with $p>0.88$) of pig image.

Justify selection: Although there are not many non-target contour interference in the figure, there are two pigs overlapping and the contour segmentation is not obvious, so we cannot use a high threshold.

Compare and comment on the Canny and SE detectors' visual results:
The Canny edge detector is well-suited for scenarios where precise edge localization is critical and noise levels are relatively low.
The SE detector excels in complex scenes with cluttered backgrounds or varying lighting conditions, where it can capture edges more effectively due to its structured approach.
While the Canny detector may produce sharper edges, the SE detector often provides more context-rich edge maps that better represent the underlying structure of the scene.

(d) Performance Evaluation

Experimental Results:

Shown below are the results of Problem 1(d):

```
Precision and Recall for Each Ground Truth:
-----------------------------------------
 GT | Precision | Recall
-----------------------------------------
  1 |   0.3620   |  0.6370
  2 |   0.4083   |  0.6776
  3 |   0.4437   |  0.6559
  4 |   0.9526   |  0.3571
  5 |   0.4510   |  0.5747
-----------------------------------------
Mean Precision: 0.5235
Mean Recall:    0.5805
F Measure:      0.5505
```

Figure16: Performance of tiger image(Structured Edge)

```
Precision and Recall for Each Ground Truth:
-----------------------------------------
 GT | Precision | Recall
-----------------------------------------
  1 |   0.3005   |  0.8004
  2 |   0.3223   |  0.7939
  3 |   0.5316   |  0.8177
  4 |   0.7031   |  0.8605
  5 |   0.5318   |  0.8379
-----------------------------------------
Mean Precision: 0.4779
Mean Recall:    0.8221
F Measure:      0.6044
```

Figure17: Performance of pig image(Structured Edge)

```
Precision and Recall for Each Ground Truth:
-----------------------------------------
 GT | Precision | Recall
-----------------------------------------
  1 |   0.1857   |  0.9597
  2 |   0.1991   |  0.9706
  3 |   0.2226   |  0.9661
  4 |   0.6608   |  0.7275
  5 |   0.2206   |  0.8255
-----------------------------------------
Mean Precision: 0.2977
Mean Recall:    0.8899
F Measure:      0.4462
```

Figure18: Performance of tiger image(Sobel Edge)

```
Precision and Recall for Each Ground Truth:
-----------------------------------------
 GT | Precision | Recall
-----------------------------------------
  1 |   0.4803   |  0.5142
  2 |   0.5300   |  0.5350
  3 |   0.5505   |  0.4950
  4 |   0.9846   |  0.2245
  5 |   0.5985   |  0.4638
-----------------------------------------
Mean Precision: 0.6288
Mean Recall:    0.4465
F Measure:      0.5222
```

Figure19: Performance of tiger image(Canny Edge)

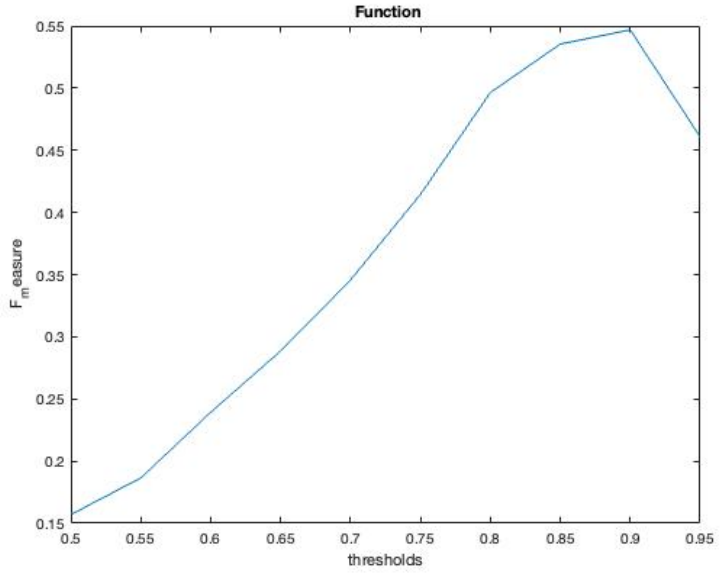


Figure20: F measure changes by threshold of tiger image

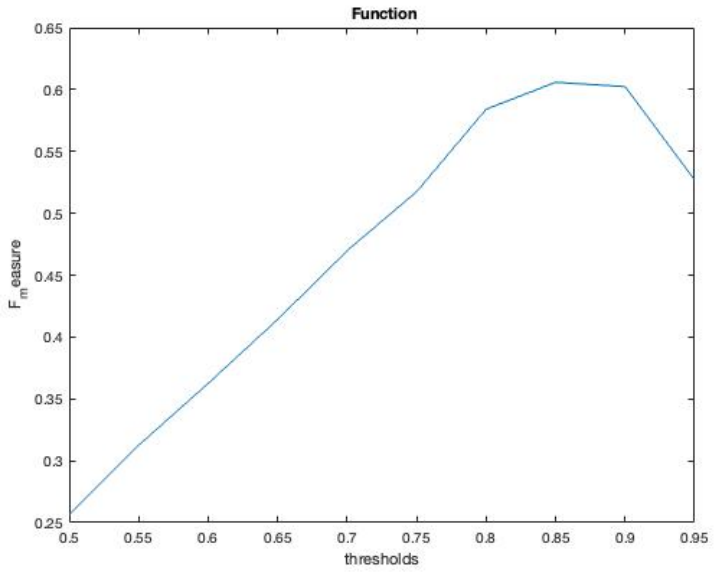


Figure21: F measure changes by threshold of pig image

Homework Answer and Discussion:

(1) Figure 16,17 use table to show the precision and recall for each ground truth of tiger and pig image by Structure Edge Method.
Figure 18,19 use table to show the precision and recall for each ground truth of tiger image by Sobel Edge Method and Canny Edge Method.
Sobel Edge Detector: A relatively high recall means that the detected edges account for a large proportion of the correct edges, but a relatively low precision means that
Canny Edge Detector: Recall and precision are neither very high nor very low, indicating that certain edges can be detected with a relatively good accuracy rate.
Structured Edge: Recall is relatively high, and precision also has a relatively good value. Although the proportion of detected edges is general, the accuracy of detecting actual edges is high.

(2) Figure 20,21 show the F measure changes by threshold on tiger and pig image by structured edge. We can find that with the increase of threshold, the result of F measure will gradually become larger, indicating that the edge detection effect of the image is better. But when the threshold increases to 90%-95, we will find that F measure will drop sharply, indicating that because the threshold is too large, as a result, the image edges can not be identified, which will affect the evaluation results.

In fact, for different edge detectors, the F measure changes by threshold is the same with Structured Edge

(3) From the above data, we can find that pig image is easier to get high F measure. Because there is not a lot of noise around the pig compared to the tiger in the grass, it is easier to detect the outline of the object in the pig image. While the tiger in the grass, part of its body is occluded by the very grass, which will lead to poor continuity of the edge, and there will be broken marks in the detection.

(4) Precision measures the proportion of true positive results among all positive predictions, while recall measures the proportion of true positive results among all actual positives. Using precision alone may not provide a comprehensive assessment of the model's performance. The F-score considers both false positives (which affect precision) and false negatives (which affect recall), making datasets balance. And using the F-score as a single metric simplifies the evaluation process by condensing precision and recall into one value. This simplification facilitates comparison between different models or algorithms.

(5) Rationale behind the F measure definition: The F-measure utilizes the mean to combine precision and recall. This averaging method ensures that both precision and recall are adequately considered without being dominated by extreme values. If precision is significantly higher than recall, or vice versa, the F-measure may decrease. This is because the F-measure is the mean of precision and recall and is influenced by the lower value.

The F-measure reaches its maximum value when precision equals recall. This occurs because, in this case, the harmonic mean of precision and recall achieves its maximum value, leading to the maximum F-measure.

Problem2

Abstract and Motivation:

Digital halftoning is a technique used in image processing to reproduce continuous-tone images using only binary (black and white) or limited-color output devices, such as printers and displays. The process involves creating patterns of dots or pixels of varying sizes and densities to simulate the appearance of shades of gray or colors.

(a) Dithering

1. Fixed thresholding

Approach and Procedures:

First, an appropriate threshold needs to be selected.

Second, for each pixel in the image, its grayscale value is compared to the chosen threshold. If the pixel's grayscale value is greater than (or equal to) the threshold, it is classified as white. If the pixel's grayscale value is less than the threshold, it is classified as black.

$$G(i,j) = \begin{cases} 0 & if\ 0 <= F(i,j) < T \\ 255 & if\ T \leq F(i,j) < 256 \end{cases}$$

Last, based on the classification results, a binary image is generated.

2. Random thresholding

Approach and Procedures:

First, an appropriate threshold needs to be selected.

Second, for each pixel, generate a random number in the range 0 ~ 255 as threshold. If the pixel's grayscale value is greater than (or equal to) the threshold, it is classified as white. If the pixel's grayscale value is less than the threshold, it is classified as black.

$$G(i,j) = \begin{cases} 0 & if\ 0 <= F(i,j) < rand(0,255) \\ 255 & if\ rand(0,255) \leq F(i,j) < 256 \end{cases}$$

Last, based on the classification results, a binary image is generated.

3. Dithering Matrix

Approach and Procedures:

First, choose an appropriate index matrix.

$$I_2(I,J) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

Second, define Bayer index matrices.

$$I_{2n}(i,j) = \begin{bmatrix} 4 * I_n(i,j) + 1 & 4 * I_n(i,j) + 2 \\ 4 * I_n(i,j) + 3 & 4 * I_n(i,j) + 0 \end{bmatrix}$$

Third, transform index matrix to threshold matrix for an input grayscale image with normalized pixel values

$$T(x,y) = \frac{I_N(x,y) + 0.5}{N^2} * 255$$

Last, based on the threshold matrix, a binary image is generated.

$$G(i,j) = \begin{cases} 0, & if\ F(i,j) \leq T(imodN, jmodN) \\ 255, & otherwise \end{cases}$$

Experimental Results:

Shown below are the results of Problem 2(a):

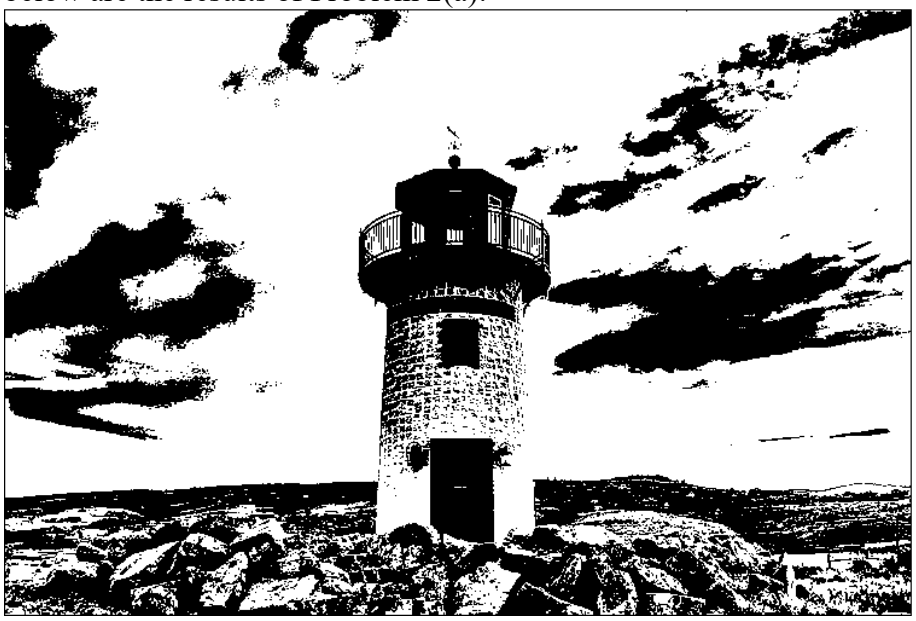


Figure1: Half-toning by fixed thresholding

Figure2: Half-toning by random thresholding

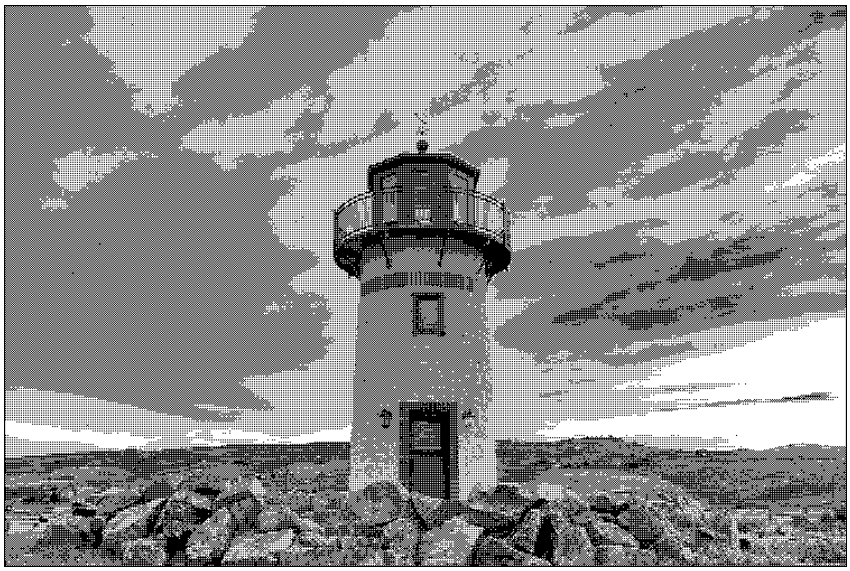


Figure3: Half-toning by Dithering matrix(I_2)

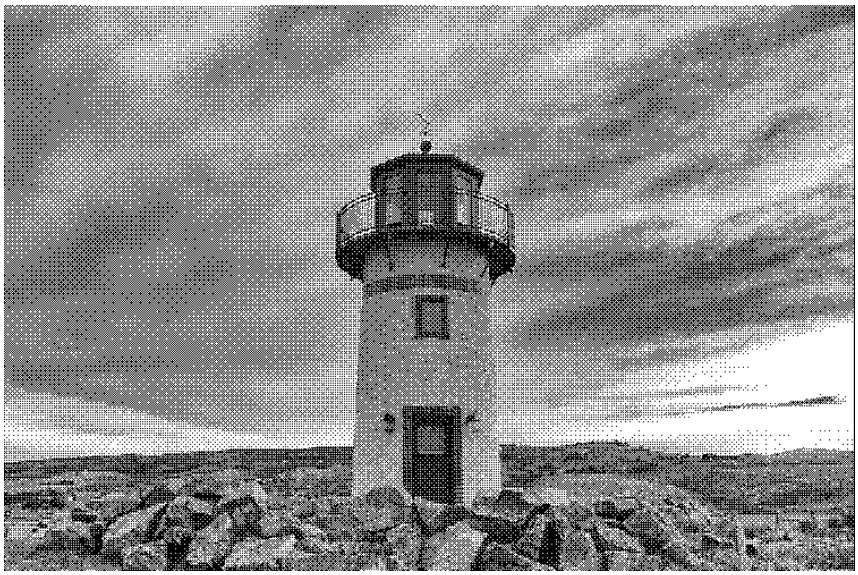


Figure4: Half-toning by Dithering matrix(I_8)

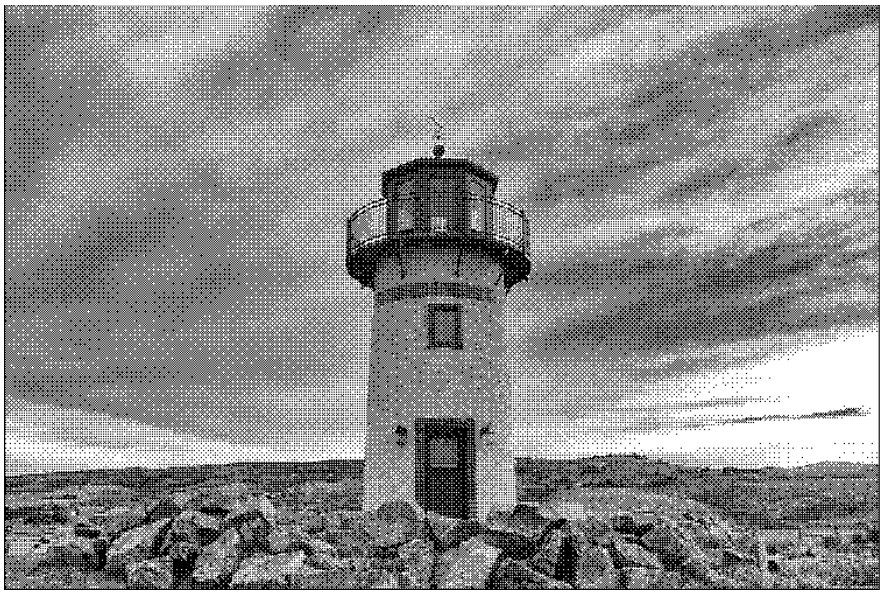


Figure5: Half-toning by Dithering matrix(I_{32})

Homework Answer and Discussion:

(1) Figure1 show the light house image using half-toning by fixed thresholding.

(2) Figure2 show the light house image using half-toning by random thresholding. And in my code, I use Mersenne Twister to create random number.

(3) Figure3 show the halftone light house image using I_2 threshold matrices.
Figure4 show the halftone light house image using I_8 threshold matrices.
Figure5 show the halftone light house image using I_{32} threshold matrices.
From figure3,4,5 we can find that as the index matrix becomes larger, finer adjustments can be allowed in the halftone process. This increased granularity can lead to a more detailed representation of grayscale or color transitions in an image. And as the size of the index matrix increases, the individual dots or patterns used for dithering become smaller and more densely packed. This can lead to a reduction in the visibility of the halftoning patterns, making them less noticeable or intrusive in the final image.

(4) Compare between 1,2,3 we can conclude that Fixed Thresholding is straightforward but sensitive to changes in lighting and contrast; Random Thresholding can reduce sensitivity but may yield unstable results; Dithering Matrix preserves image quality while displaying continuous-tone images, but it comes with higher computational complexity.

(b) Error Diffusion

Approach and Procedures:

First, the image is processed iteratively, pixel by pixel, starting from the top-left corner and moving sequentially through each row and column by serpentine scanning or others.

Second, for each pixel in the image, we choose to use fixed thresholding to binarize the original data.

Third, after quantization, the difference between the original intensity value and the quantized intensity value is calculated for the current pixel. This difference represents the quantization error associated with that pixel.

Fourth, the quantization error from the current pixel is distributed to neighboring pixels according to a predetermined error diffusion kernel or matrix. This kernel specifies the weights or coefficients to be applied to the error, and determines how much of the error is propagated to neighboring pixels. Common error diffusion kernels include the Floyd-Steinberg, Jarvis-Judice-Ninke, and Stucki kernels.

Fifth, the error matrix is updated with the distributed errors, reflecting the cumulative error contribution from previously processed pixels.

Last, Steps 2 to 5 are repeated for each pixel in the image until all pixels have been processed.

Experimental Results:

Shown below are the results of Problem 2(b):

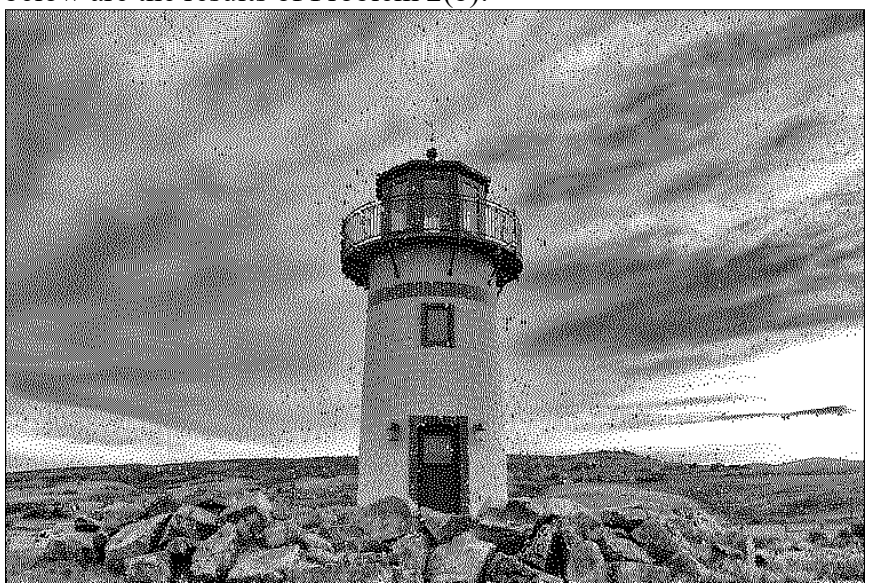


Figure6: Half-toning by Error Diffusion(Floyd-Steinberg)

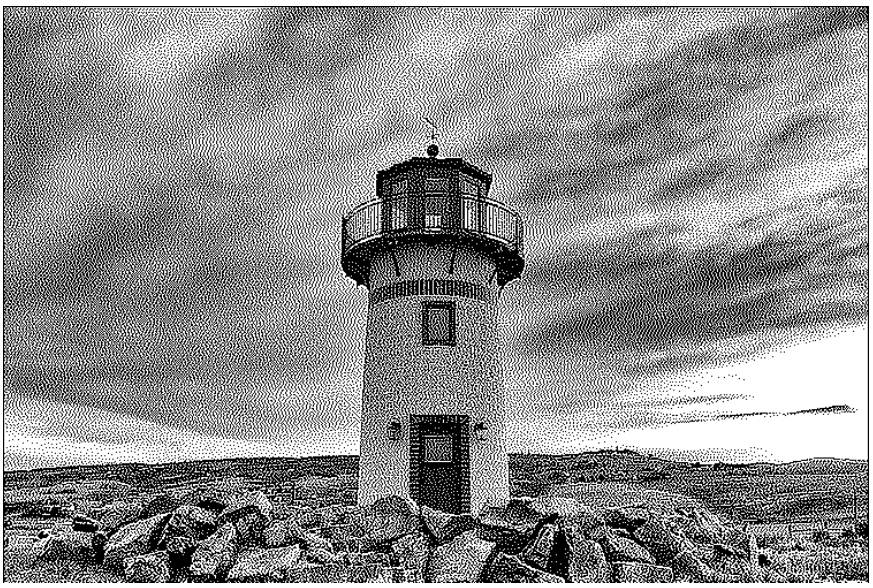


Figure7: Half-toning by Error Diffusion(Jarvis-Judice-Ninke)

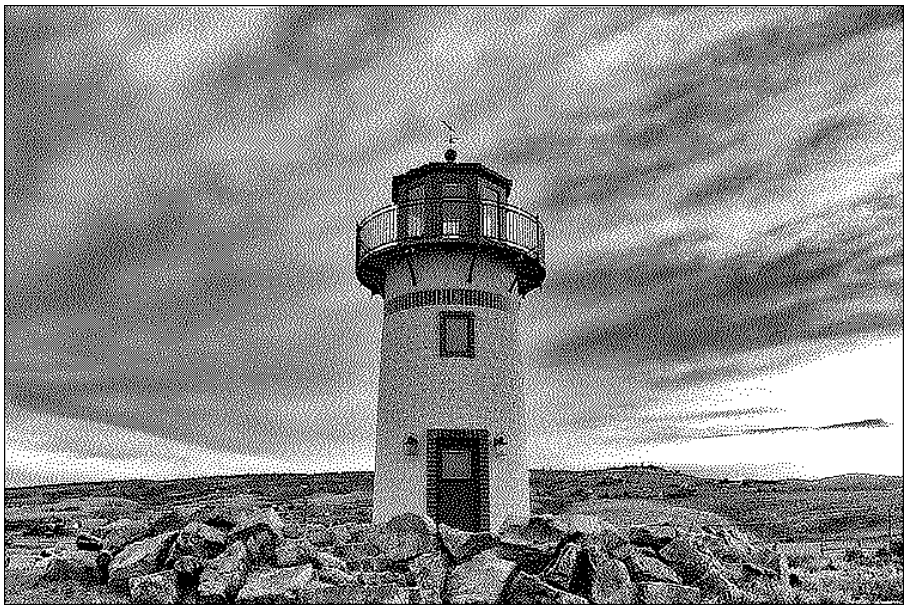


Figure8: Half-toning by Error Diffusion(Stucki)

Homework Answer and Discussion:

(1) Figure6 show the light house image using half-toning by Error Diffusion(Floyd-Steinberg).

(2) Figure7 show the light house image using half-toning by Error Diffusion(Jarvis-Judice-Ninke)

(3) Figure8 show the light house image using half-toning by Error Diffusion(Stucki)

(4) Compare raster parsing with serpentine paring.
Raster parsing scans the matrix data row by row or column by column in a fixed order, so I think the first part of each line will not suffer from a lot of error diffusion. It is simple to implement, easy to understand, but may not be suitable for certain algorithms, such as spiral scanning or processing localized regions in image processing.
Serpentine parsing alternates directions along rows or columns, scanning one row from left to right and the next row from right to left, and so forth. It requires slightly more complex implementation, but I think the pixels of each part can receive feedback from error diffusion and makes the resulting image look smoother.

(5) Compare these results with Dithering Matrix method in part (a). I prefer Error Diffusion Method. Because, this method's result Produces visually pleasing results with smooth tonal transitions. It not only can preserve fine details and texture in the image, but also can reduce the visibility of quantization artifacts such as banding or contouring.

Problem3

Abstract and Motivation:

We will introduce Separable Error Diffusion method and MBVQ-based Error diffusion method to solve the Color Half-toning with Error Diffusion problem

(a) Separable Error Diffusion

Approach and Procedures:

First, we need to convert RGB images to CMY images.

Second, we need to use Floyd-Steinberg error diffusion(problem2) to each channel of CMY images.

Last, we need to convert CMY images to RGB images and output the image.

Experimental Results:

Shown below are the results of Problem 3(a):

Figure1: Color Half-toning by Separable Error Diffusion

Homework Answer and Discussion:

Figure1shows color half-toning by Separable Error Diffusion.

The main shortcoming of this approach:

Separable error diffusion requires applying error diffusion independently to each color channel. Independent processing of each color channel may result in a loss of spatial correlation. While colors are often correlated spatially in the original color image, separable error diffusion may disrupt this correlation, resulting in visually unnatural effects.

Handling each color channel independently may lead to inconsistencies in color. In some cases, there may be issues with color bias or distortion, particularly at edges or transitions between different colors.

(b) MBVQ-based Error Diffusion

Approach and Procedures:

First, for each pixel we need to find out which MBVQ space it lies in.

Second, we need to find the point in this space that is closest to the target point and treat it as the result.

Third, we need to use Floyd-Steinberg error diffusion(problem2) to update the neighboring pixel.

Last, output the image.

Experimental Results:

Shown below are the results of Problem 3(b):

Figure2: Color Half-toning by MBVQ-based Error Diffusion

Homework Answer and Discussion:

Figure2shows color half-toning by MBVQ-based Error Diffusion.

(1) The key ideas on which the MBVQ-based Error diffusion method is established:

1. Minimal Brightness Variation Criterion (MBVC): The MBVC is a fundamental concept in color halftoning that aims to minimize the brightness variation in rendered colors.
2. Color Space Transformation: The method involves transforming the halftone pattern to preserve the average color of the input while reducing brightness variation.
3. Halftone Quadruples: Through the application of the MBVC and color space transformation, the method determines the minimal set of participating halftone colors required to render a given input color.

4. Ink Relocation Post-Process: The method incorporates an Ink Relocation post-process, which further refines the halftone output by adjusting the placement of colored dots based on the MBVC criteria.

The reason why this method can overcome the shortcoming of the separable error diffusion method:

The MBVQ-based Error Diffusion method overcomes the shortcomings of separable error diffusion by prioritizing color accuracy through the Minimal Brightness Variation Criterion, and reduce visual artifacts, and select optimal halftone colors, then utilizing an Ink Relocation post-process for refinement, and enhancing color perception based on human color characteristics.

(2) Compare figure1 and figure2.

Compare to separable error diffusion, the MBVQ-based Error Diffusion method tends to provide higher color accuracy. And we can find that the MBVQ-based Error Diffusion method aims to reduce visual artifacts and image noise, thereby enhancing overall visual quality. So the MBVQ-based Error Diffusion method typically produces clearer and more accurate halftone outputs by optimizing color selection.