EE 569 Digital Image Processing: Homework #2

EE 569: Homework #2

Issued: 01/29/2020 Due: 11:59PM, 02/16/2020

Problem 1: Edge Detection

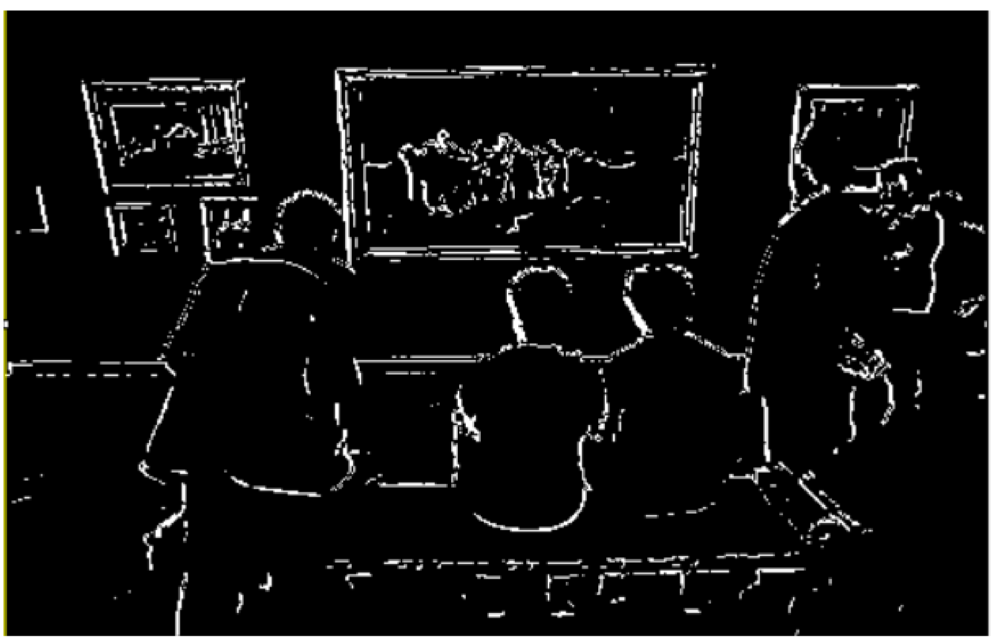
a) Sobel Edge Detector

Steps to implement sobel edge detector:

Gradient along the x direction

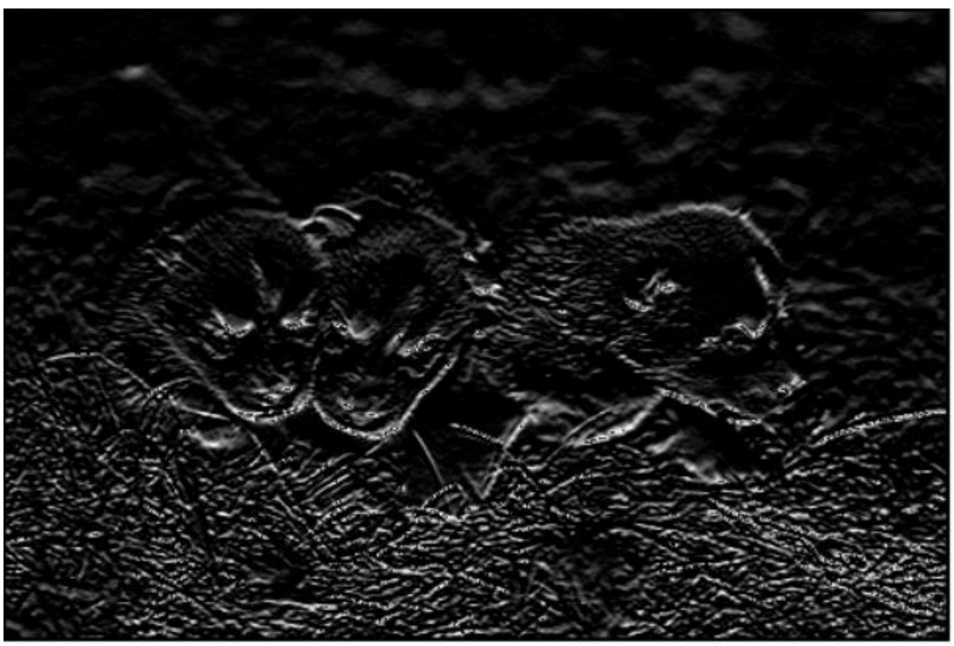
Gradient along y direction

Gradient Magnitude Image: Gallery.raw

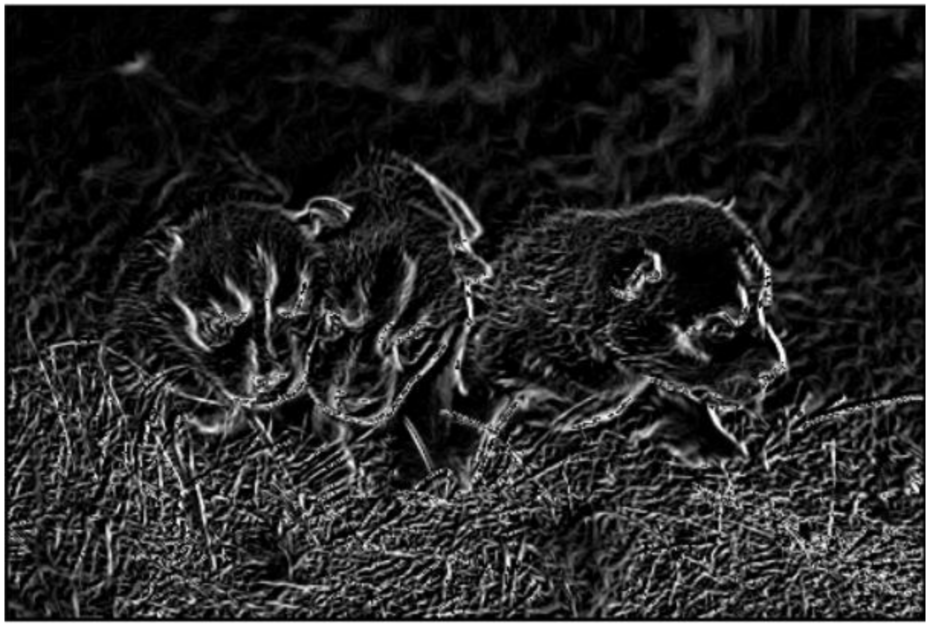


Binary Image

Gradient along the x direction



Gradient along y direction



Gradient Magnitude Image: Dogs.raw

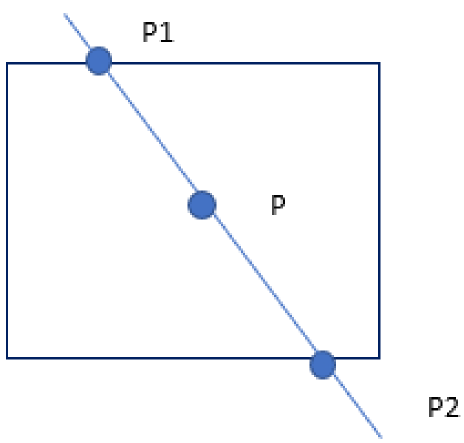
Binary Image

b) Canny Edge Detector

Canny Edge Detector consists of the four following steps:

- Apply gaussian filter for denoising
- Find the image gradient of the resulting image in the x and y direction and compute the gradient magnitude map.
- Non maximum Suppression
- Double Thresholding using Hysteresis method.

(1) Non-Maximum Suppression is the edge thinning technique. It converts the edge hand to edge line. It retains only those pixels that are an edge. The gradient image which is edge probability map is fed to the non-maximum suppression section. The edge probability map consists of thick edges. Generally, along the direction of highest change in intensity or along the direction of the gradient the center pixel has maximum gradient magnitude compared to the other pixels. Non maximum suppression suppresses all the other pixels and retains only the local maxima along the gradient direction. By doing this only thin edges are preserved and thus having only one accurate response to the edge.



In the figure, we have considered three pixel points P1, P and P2 along the gradient direction. Non maximum suppression will choose the local maxima value P over the gradient values at P1 and P2 along the gradient direction because P has a higher magnitude. Thus, suppressing the values at P1 and P2 to zero. Thus, the resulting image will only have thin edges.

(2) Double Thresholding is used in Canny Edge Detector. If the gradient is lower than the lower threshold then it is mapped to zero and considered as ‘no edge’. If the gradient is higher than upper threshold then it is mapped to 255 and it is considered as an ‘edge’. If it is between the upper and lower threshold and if it has neighborhood of strong edges, then it is considered as edge else it is considered as no edge. The lower threshold eliminates edge pixels caused by noise or change in color. By doing double thresholding the weak edges are suppressed and the strong edges are retained. Thus, the resulting image is a binary image. Generally, the ratio of upper threshold to lower threshold is 3:1 or 2:1.

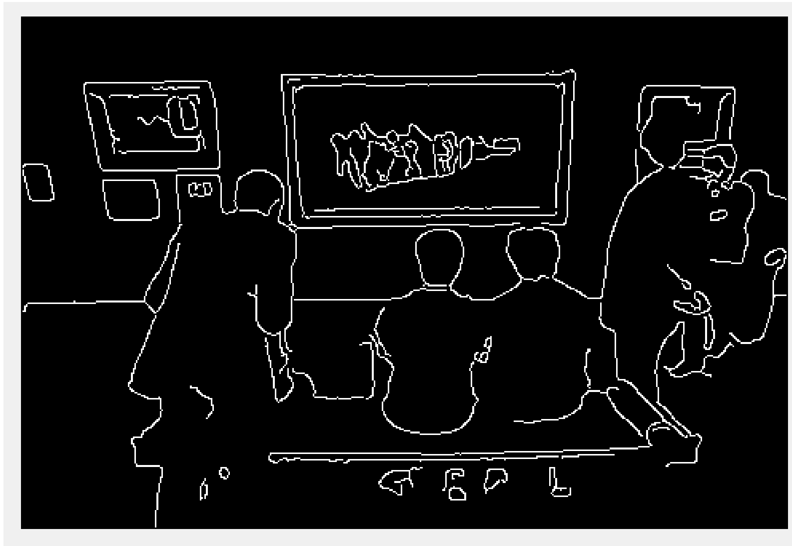
(c) **Results**

The edges are thinner compared to the sobel edge detector due to non-maximum suppression.

It can be observed that for lower and upper threshold values in 1:2 ratio the edges that are below 0.1 are eliminated and the edges above 0.2 are accepted. Whereas in 1:3 ratio edges below 0.1 are eliminated and above 0.3 are accepted. Thus, it eliminates weaker edges more. The sigma value for the gaussian filter is set to standard deviation of 1.4.

Input Image

Output Image: Lower: Upper=1:2 [0.1, 0.2]



Output Image Lower:Upper=1:3 [0.1,0.3]

It can be observed that for Dogs.raw image with lower and upper threshold values in 1:2 ratio the edges that are below 0.2 are eliminated and the edges above 0.4 are accepted. Whereas in 1:3 ratio edges below 0.15 are eliminated and above 0.45 are accepted. Thus, it eliminates weaker edges more. The sigma value for the gaussian filter is set to 1.4.

Output Image Lower:Upper=1:2 [0.2,0.4]

Output Image Lower:Upper [0.15,0.3]

Output Image Lower: Upper =1:3 ratio [0.15:0.45]

c)Structured Edge

1) Structured Edge

Procedure:

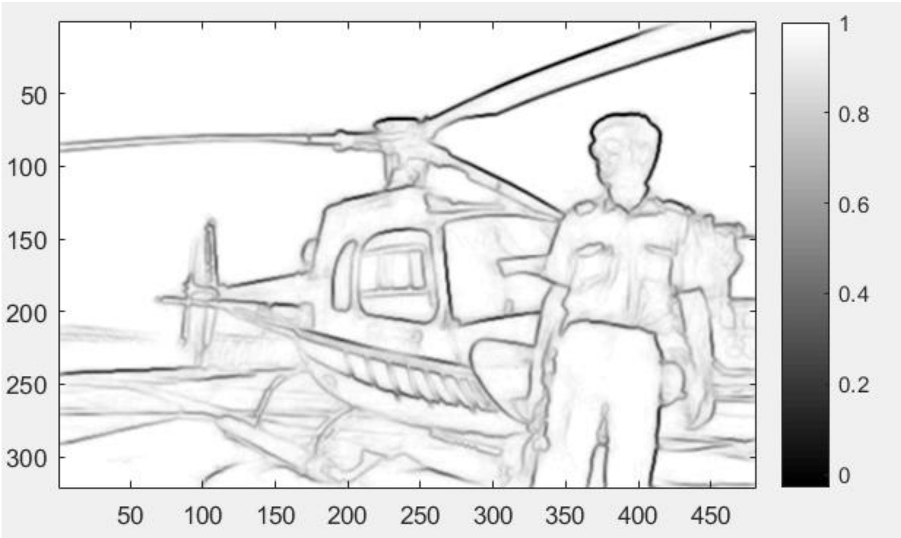
- 32x32 image patch is considered and the resulting 16x16 segmentation mask is predicted
- The tree is built during the training phase and traversed during the testing phase.

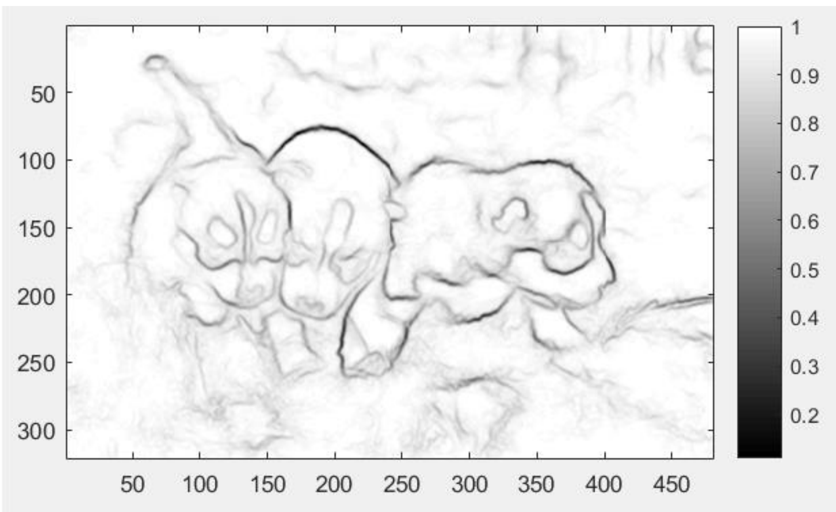
- Random forest is a combination of multiple decision trees. For each decision tree, and at each node the best splitting criteria is determined by the gini impurity or Shannon entropy.
- Based on the splitting criteria at each node the traversal would be to left branch or the right branch.
- We use an intermediate mapping to map the structured labels Y to Z. Z checks whether every pair belongs to a segment or not. And then from Z to C. C indicates the discrete labels based on the similarity of the y structured labels.
- Take the average of the response from each tree to obtain a resulting image with darker edges that correspond to the maximum votes given by ground truth image. Lighter shades of the edges correspond to those edges that are not frequent among the ground truth images.

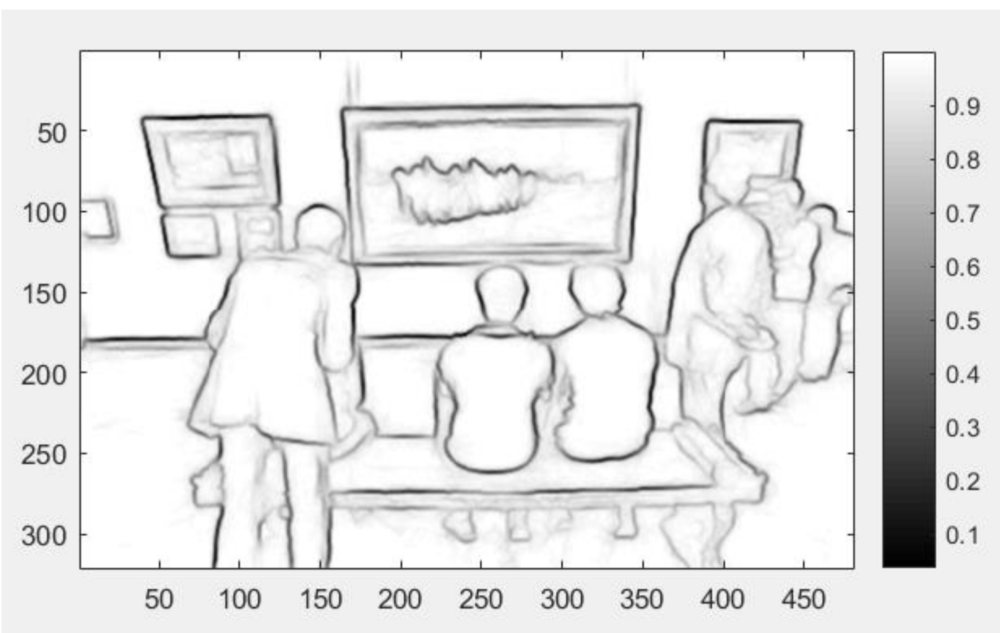
(2) Random Forest classifier

Random Forest classifier uses random forest which is a combination of multiple decision trees. During the training phase, the tree is built and during the testing phase the tree is traversed to predict the label for the input. For each decision tree, we traverse the tree and classify the input based on splitting criteria at each node of the decision tree. Based on the splitting criteria the input will be shifted towards the right or the left of node. The training is ended when the maximum depth is reached. Each decision tree is trained independently. At each node, a good split function is created based on information gain criteria. The theta for the split function is selected to maximize the information gain. The tree is traversed recursively via left node and later via right node. To estimate the information gain Shannon entropy or Gini impurity can be used. Randomness is injected at every node of the decision trees by randomly choosing the features to train each node. Randomness increases the accuracy. Each decision tree has high variance and by combining the results of multiple decision trees the output will be better. In Structured Random Forest, the input is an image patch and the output label would be a segmentation mask or binary edge map. Two main challenges in training random forest with structured edge are it is very expensive, and the information gain is not well defined. The goal is to map all the structed labels y into discrete class labels C. All the similar structured labels y is mapped to a discrete class label C. Two stage approach is applied. Firstly, mapping Y to Z(intermediate mapping) followed by Z to C mapping. In intermediate mapping, the structured labels y (16x16 segmentation mask) is used to define a mapping function that classifies if a pair of pixels belongs to a segment or not. But in this case the no of unique pixel pairs would be 32640 and it is expensive. Therefore, we sample m dimensions in Z. The mapping function is applied at each node and thus adds additional randomness. The principal component analysis further reduces the dimensionality to 5 from m=256 dimensions. The discrete labels can be two class or multi-class. The discrete labels C is obtained using k-means or top PCA dimensions.

Result


50
100
150
200
250
300
50 100 150 200 250 300 350 400 450
1
0.8
0.6
0.4
0.2
0


50
100
150
200
250
300
50 100 150 200 250 300 350 400 450
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2


50
100
150
200
250
300
50 100 150 200 250 300 350 400 450
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

It can be observed that the results of the structured edge is better than that obtained using Sobel or Canny Edge detection. Since structured edge is trained using the ground truth image done by humans its performance mimics human nature. It has ignored all the unwanted noise from the image and only the prominent edges are displayed. The lighter shades of the edges depend on the no of times those edges have occurred in the ground truth image.

d) Performance Evaluation

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

3) Precision is the ratio between true positive and the sum of (true positive and false positive). Recall is the ratio between true positive and the sum of (true positive +false negative). If we consider everything as a edge then the recall is high and the precision is low. In order to balance both the entities the F measure is calculated as a combination of recall and precision. A higher F measure indicates better performance. It is possible to get a high F measure if P>R. To have a higher F measure both the recall and precision should be high. The F measure is calculated by 2*(P.R)/(P+R). When P=R , the result of F is P. Thus the F measure will be maximum.

Problem 2: Digital Halftoning

Motivation

Digital Halftoning is the technique to create a gray scale visual effect by varying the density of the dots. It is used in technologies that can support only a binary output. For example, printers can either fire a dot or not. The density is based on dpi (dots per inch). In order, to get a darker shade, the density of the pixels in that region is increased and to get a lighter shade, the density of the pixels in that regions is sparse. This is due to the property of the eyes, the dots become invisible if they are closer to each other and the human eye just records the overall intensity. In Digital Halftoning, the output image has just two levels, either one or zero. Thus, Digital Halftoning gives the illusion of continuous gray scale output with just binary values. Most of the printers do not support different shades of gray or other colors. Thus, the output image must be formed from just two levels white (no ink) and black (ink). It improves the quality of the output image with minimal cost.

(a)Dithering

It is a technique used in digital halftoning. It creates the output image with the same no of dots as the no of pixels in the input image. By thresholding each pixel in the input image, an output image with binary values is generated.

1)*Fixed thresholding*

Procedure:

- Read the input grayscale image LightHouse.raw.

- At each pixel, if the pixel value is greater than 128, the corresponding output pixel is set to 255 else zero.
- Thus, the resulting image is binary with just two levels.

Results

We can see that there is a loss of the details of the image. It produces a poor quality of continuous output. Since it is based on a fixed threshold, the output image has a lot of monotonicity and thus reducing the continuous gray scale visual effect.

2) Random Thresholding

In order to reduce the monotonicity, random thresholding is used. Each pixel value is compared with a random value threshold and the output binary image is generated. In this case, we introduce uniform distribution noise by picking random values from the uniform distribution samples.

Procedure

- Read the input image LightHouse.raw.
- Each pixel value is quantized based on a random sample from uniform distribution (rand(i,j)). Using the random package generated uniformly distributed samples.
- If the pixel value is greater than rand(i,j) then the corresponding output pixel is 255 else the output is zero.

Result

Due to random thresholding, at some locations there is a higher threshold and at some locations there is lower threshold. Due to this adaptive thresholding, it does not produce patterned artifacts. The output is better than fixed thresholding by reducing monotonicity and gives a visual effect of continuous gray scale output. But, the details of the output image are not great due to random thresholding and the noise deteriorates the fine image details and the image is not sharp.

3) Dithering Matrix

The dithering matrix determines in which order the dots are turned on. For example, if we consider an index matrix I=[1,2; 3,0]. The lowest threshold is at 0 and highest threshold is at 3. The order in which the dots will be turned on is 0,1,2,3. Thus, completing one diagonal direction followed by the other diagonal direction. A specific pattern is followed across the image. Thus, the output image has a distinctive cross hatch pattern.

Procedure

- Read the input image LightHouse.raw
- Initialize the index matrix 2x2 [1,2;3,4]
- The Bayer's index matrix is generated using the below formula

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

- The threshold matrix is computed using index matrix using the following formula to fit a dynamic range 0-255

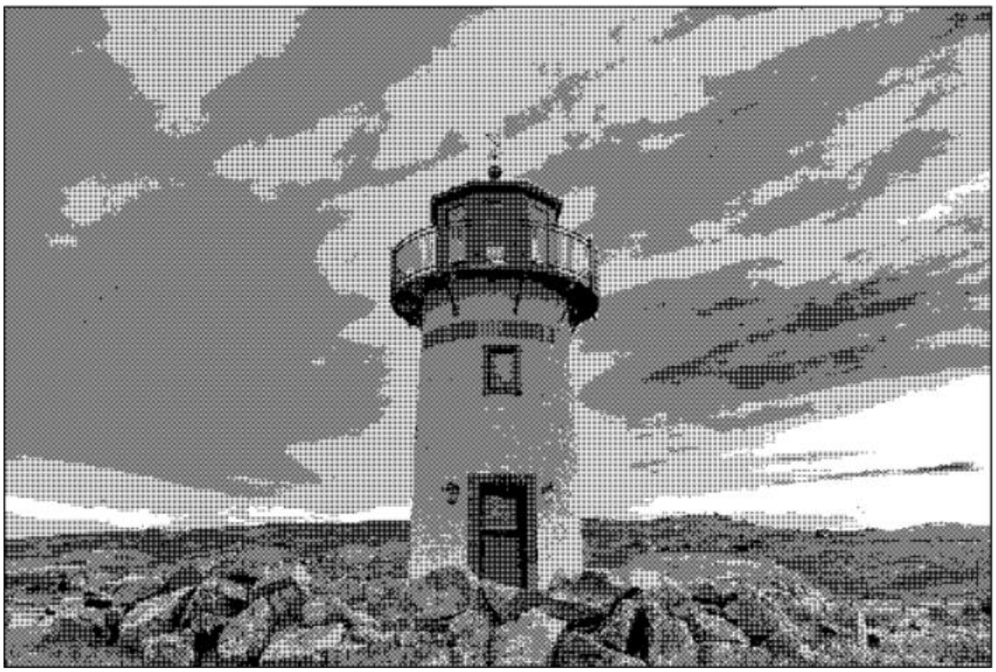
$$T(x,y) = \frac{I_N(x,y) + 0.5}{N^2} \times 255$$

- If the input image pixel value is greater than threshold, the output pixel is 255 else zero. The bayer's matrix is repeated across the image.

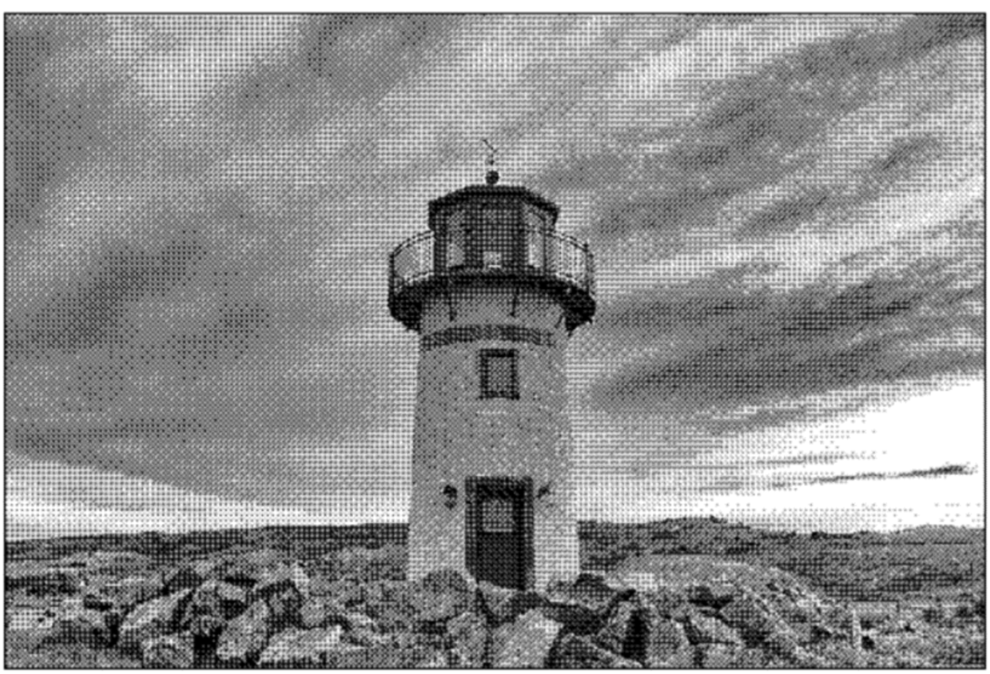
Results

We can observe that the output image generated by dithering matrix produces periodic cross hatch patterns. The output produced by index matrix I of size 2x2 has distinctive texture like visual patterns. The output image for index matrix I of size 8x8 also has sketch patterns but it is smoother. The output image for index matrix I of size 32x32 has better detail retention.

I 2X2 Index Matrix

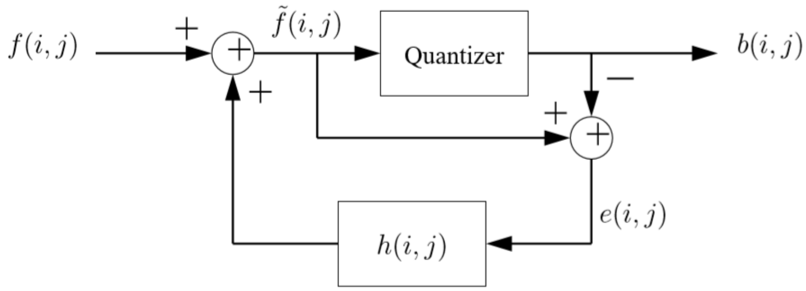


Index 8x8 Index Matrix



(b)Error Diffusion

Error Diffusion is a digital half toning technique. It is not just a point wise operation, it disperses the error generated at each pixel to its neighbours pixels. It computes the error caused ny thresholding a given pixel and propogating it to neighbouring pixels to compensate for the gain or loss in intensity. Error Diffusion has a higher resolution compared to other halftoning methods. Error Diffusion is one technique that can generate blue noise which is optimum for producing better quality images.



Procedure

The direction of diffusion of error is in the direction of scanning.

- Read the input image
- At each pixel location based on fixed thresholding, if the pixel local is greater than 128 then the b(I,j) is set to 255 else zero.
- The error e(I,j) is calculated between the current pixel value and the b(I,j).
- This error is difused to the future pixels based on the diffusion matrix.
- Go to step 2
- The output image is a binary image

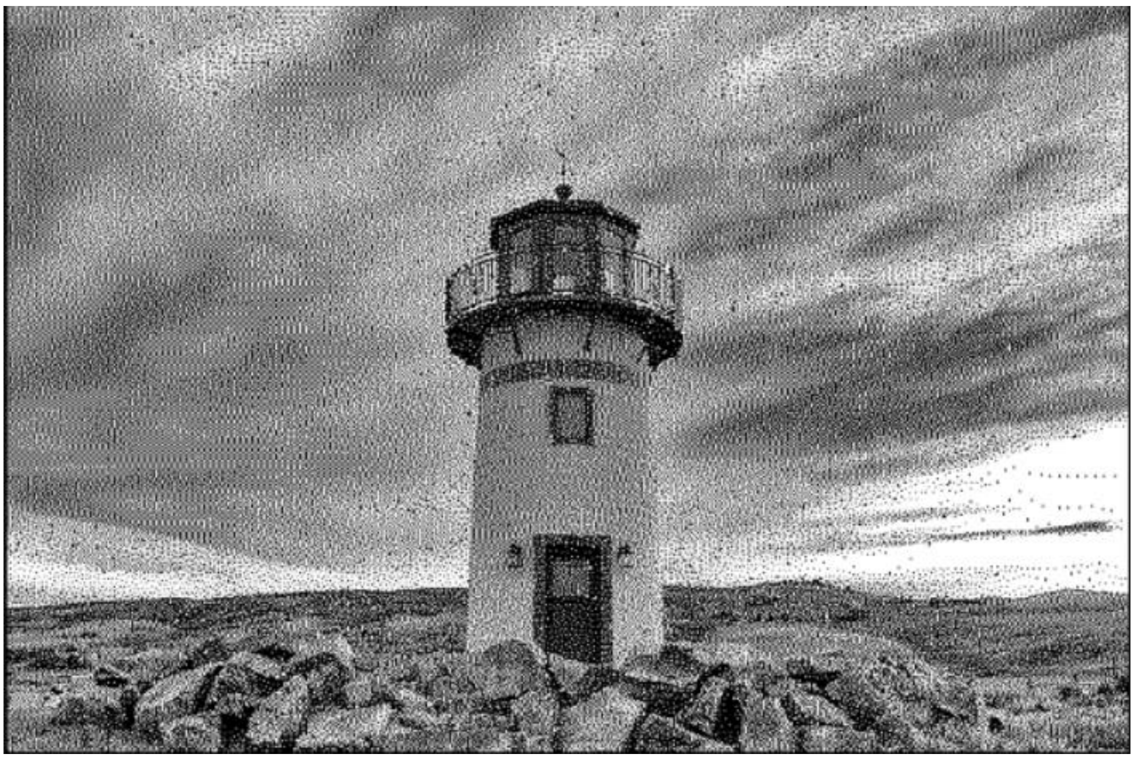
1)Flyod-Steinberg's Error diffusion

It uses the diffusion matrix of size 3x3. It diffuses the quantization error to the neighbouring pixels. The error is diffused to the future pixels. Sepentine scanning is used in which the image is scaned from left to right and then followed by right to left. Serpentine scanning is better than raster scan(scanned row by row from left to right). The reason is that the error gets accumulated at the last pixel in each row in raster scan. In sepertine scanning the neighbouring pixels are always diffused with errors from the previous pixels in the scan. The following diffusion matrix is used.

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

Result

It can be observed that Floyd-Steinberg's error diffusion is better than the dithering matrix in part a. The dithering matrix produces patterned artifacts like cross hatch patterns. But error diffusion using Floyd produces a more finer image. Image quality is good in Floyd.



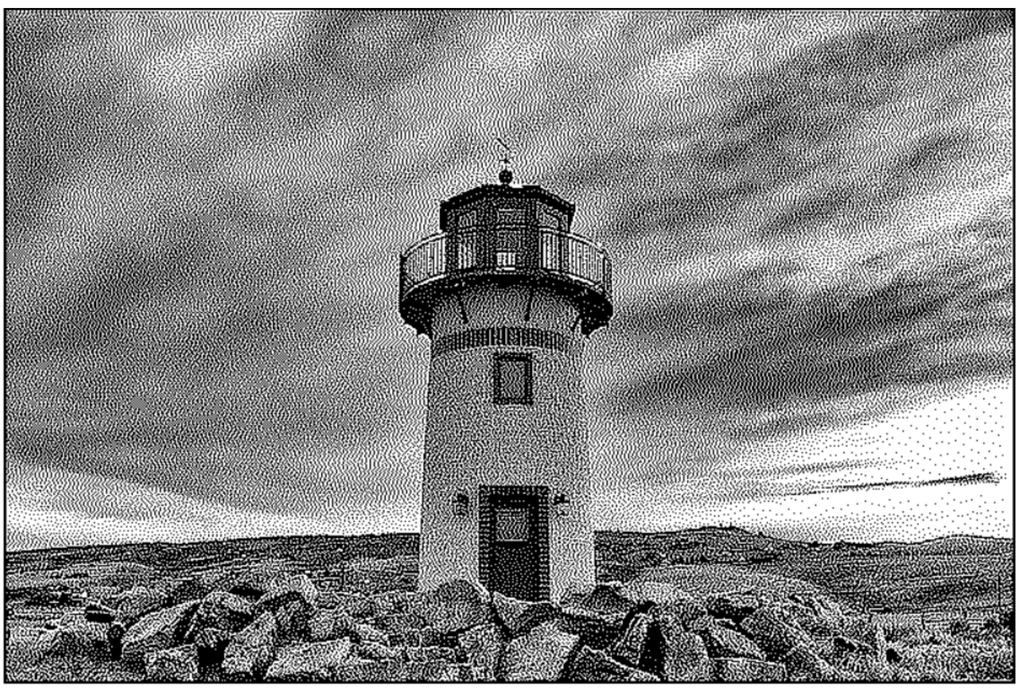
2 b) Jarvis, Judice, and Ninke (JJN) Error Diffusion

The Jarvis, Judice, and Ninke (JJN) Error Diffusion using a diffusion matrix that diffuses error to the neighbouring pixels with one additional level compared to Floyd error diffusion. The diffusion matrix used is as follows:

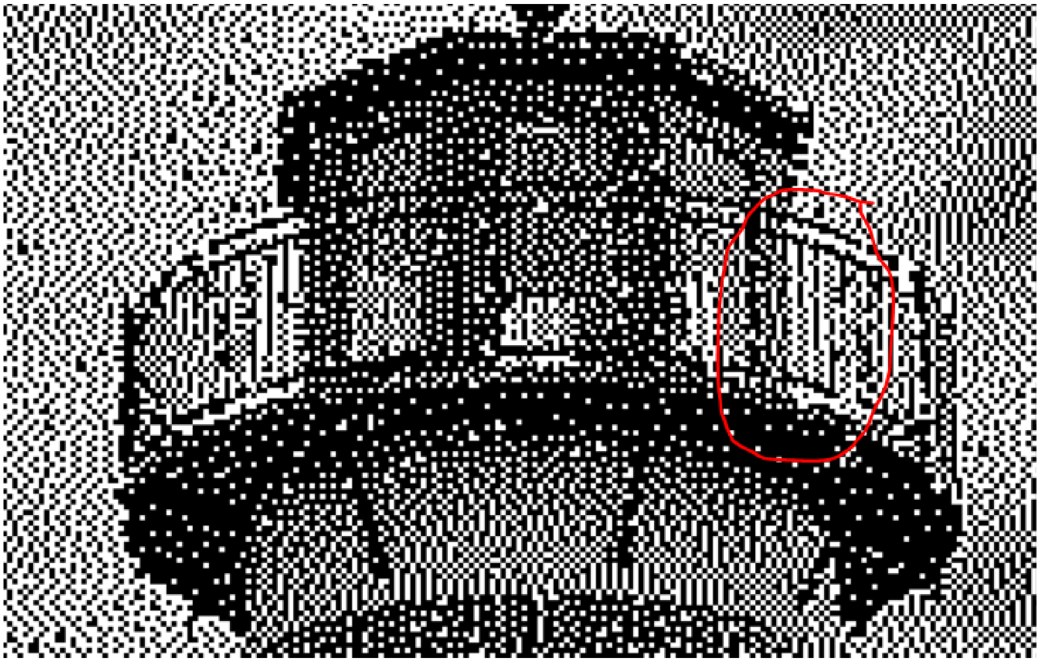
$$\frac{1}{48}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{bmatrix}$$

Results

The dithering is coarser compared to floyd, but it has fewer visual artifacts. It is comparatively slower. The dithering matrix produces patterned artifacts like cross hatch patterns. Thus JNN error diffusion is better.



It can be observed that the edges are connected or clearer in JNN error diffusion compared to Floyd error diffusion.

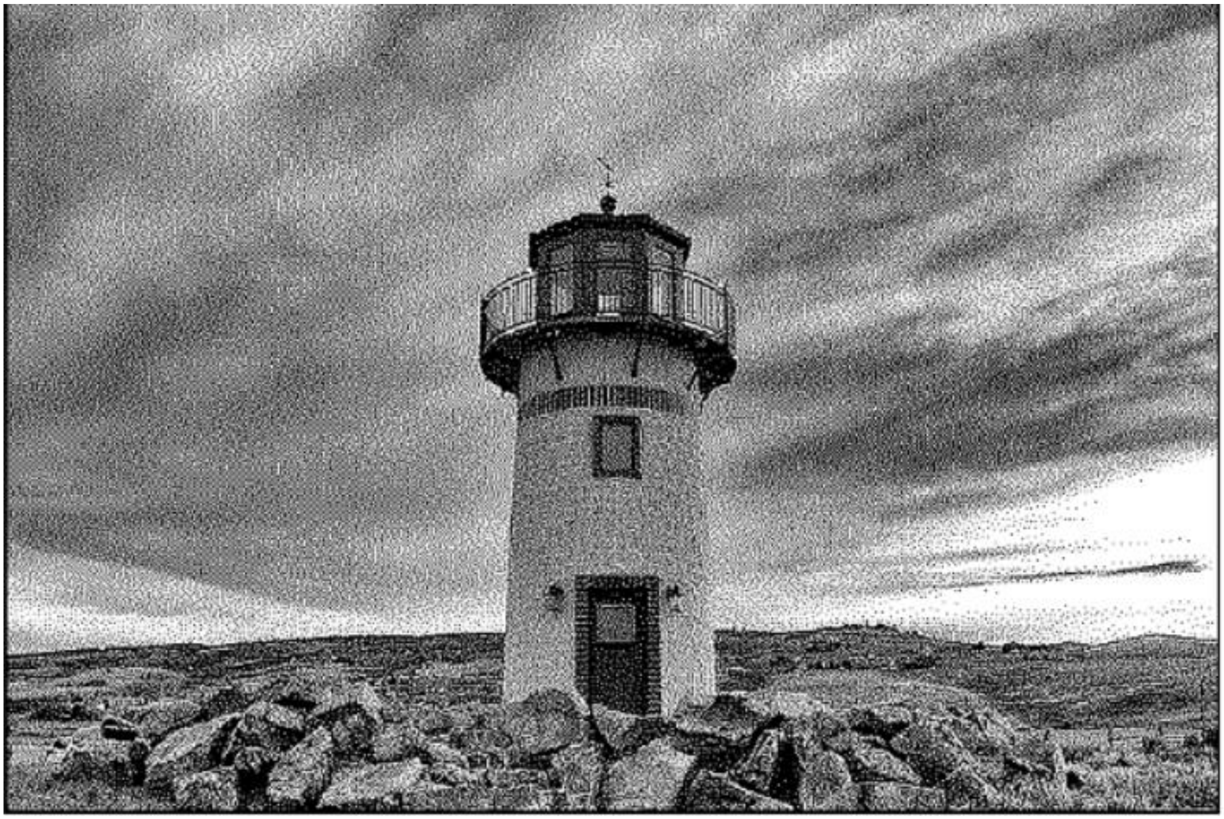


Floyd

JNN

3) Stucki Error Diffusion

It a slight modification of the JNN diffusion matrix where it differs by the coefficients of the 5x5 diffusion matrix. It is compared to be faster. The dithering matrix produces patterned artifacts like cross hatch patterns. Thus, Stucki error diffusion is better. The output image has sharper edges compared to the other alogorithms.



c) Color Halftoning with Error Diffusion

1) Separable Error Diffusion

The input image is converted from the RGB color space into CMY color space.

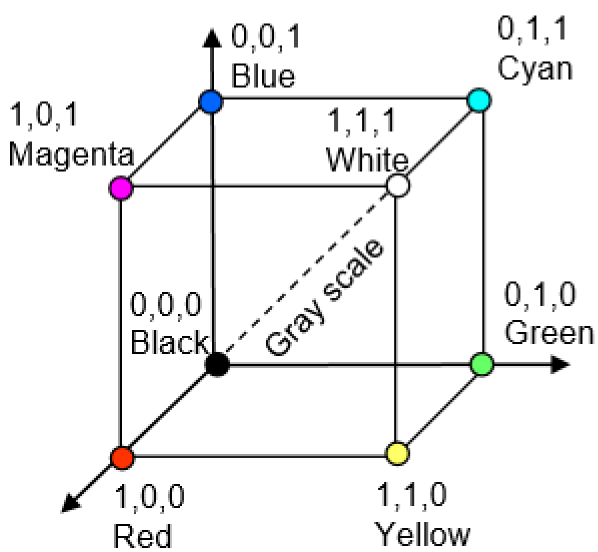
The Floyd error diffusion is applied indvidually on each C,M,Y channels

The processed image is converted back to RGB space by subtracting C,M,Y values from 255.

The noise is higher in separable diffusion compared to MBVQ based color diffusion.

2)MBVQ based color Diffusion

In Minimal Brightness Variation Quadruples(MBVQ), the colors which have the minimal brightness difference are choosen. The input image is mapped to 8 colors that are generated in color cube.



Procedure:

- Read the input color image Rose.raw.
- Based on the R,G,B values at each pixel, one of the six tetrahedral are used.

- Based on the quadrant, the nearest vertex is chosen among the four vertices of the tetrahedra.
- Floyd Steinberg's color diffusion is implemented.
- The generated output image will have eight colors.