EE569 Digital Image Processing

Homework#2

Wenbo Ye

wenboye@usc.edu

Issued date: 02/15/2020 due date:02/16/2020

---

1. Problem 1: Edge Detaction
   a) Sobel Edge Detector
      i. Abstract and Motivation

         Mainly for edge detection. Used to get the first degree of a digital image.

         The operator contains two sets of 3x3 matrices, respectively horizontal and vertical, which can be convoluted with the image plane to obtain the horizontal and vertical brightness difference approximations.
      ii. Approach and Procedures

         The operator contains two sets of 3x3 matrices, respectively horizontal and vertical, which can be convoluted with the image plane to obtain the horizontal and vertical brightness difference approximations. If A represents the original image, Gx and Gy represent the image gray value detected by horizontal and vertical edges respectively, the formula is as follows:

$$
\begin{aligned}
Gx &= (-1)*f(x-1, y-1) + 0*f(x,y-1) + 1*f(x+1,y-1) \\
&\quad +(-2)*f(x-1,y) + 0*f(x,y)+2*f(x+1,y) \\
&\quad +(-1)*f(x-1,y+1) + 0*f(x,y+1) + 1*f(x+1,y+1) \\
&= [f(x+1,y-1)+2*f(x+1,y)+f(x+1,y+1)]-[f(x-1,y-1)+2*f(x-1,y)+f(x-1,y+1)]
\end{aligned}
$$

$$
\begin{aligned}
Gy &=1* f(x-1, y-1) + 2*f(x,y-1)+ 1*f(x+1,y-1) \\
&\quad +0*f(x-1,y)\ 0*f(x,y) + 0*f(x+1,y) \\
&\quad +(-1)*f(x-1,y+1) + (-2)*f(x,y+1) + (-1)*f(x+1, y+1) \\
&= [f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)]-[f(x-1, y+1) + 2*f(x,y+1)+f(x+1,y+1)]
\end{aligned}
$$

         Then, we can get the magnitude by calculating the magnitude (the arithmetic square root of the sum), normalize the magnitude and add threshold, we get the final result of image.
      iii. Experimental Results

Dog sobelX.raw

Dog sobelY.raw

Dog sobel_res.raw

Gallery sobelX.raw

Gallery sobelY.raw

Gallery sobel_res.raw

Figur1. result of sobel edge detection

iv. Discussion

1. I have normalized the Gx and Gy in function “sobelGx.m” and “sobelGy.m”, the results are shown as the matrixes as “sobelX” and “sobelY”in workspace.
2. I have normalized the magnitude in function “mag.m”, the result is shown as the matrixes as “magn”in workspace.
3. I got the threshold is 0.3 in dog.raw, 0.25 in gallery.raw.

Sobel operator detects the edge according to the grayscale weighted difference between the upper and lower, the left and right adjacent points of the pixel and reaches the extreme value at the edge. It can smooth the noise, provide accurate edge direction information, and the edge positioning precision is not high enough. When the accuracy is not very high, it is a common edge detection method.

b) Canny Edge Detector

i. Abstract and Motivation

Canny edge detection is a technique to extract useful structural information from different visual objects and greatly reduce the amount of data to be processed. It has been widely used in various computer vision systems. Canny found that the requirements for edge detection were similar in different visual systems, so a widely applied edge detection technology could be realized. General criteria for edge detection include:

1) detect the edge with a low error rate, which means to capture as many edges in the image as accurately as possible.

2) the detected edge shall be precisely located at the center of the real edge.

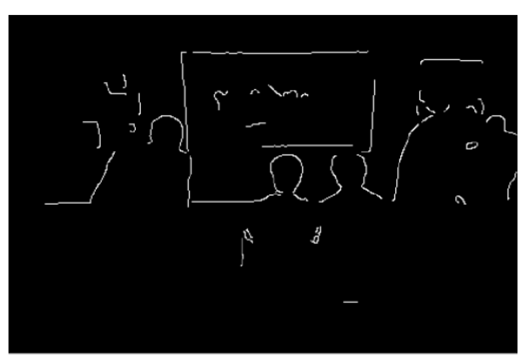
3) a given edge in an image should only be marked once, and where possible, the noise of the image should not produce false edges.

ii. Approach and Procedures

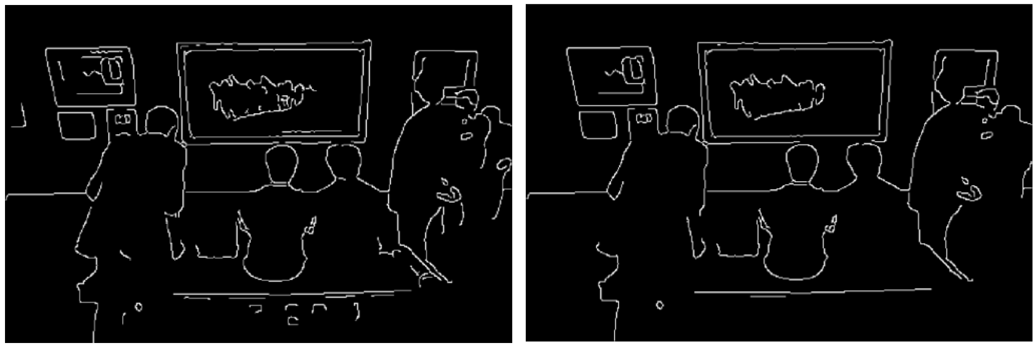
There is totally 4 steps.

a) The image is filtered by gaussian
b) Calculate the magnitude and direction of the gradient
c) The gradient - amplitude image is not greatly suppressed
d) Double threshold processing and connectivity analysis

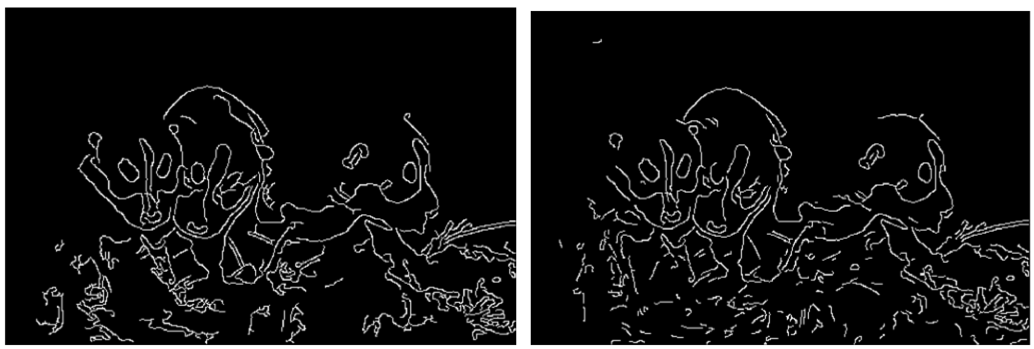
iii. Experimental Results



Gallery_canny[0.1, 0.2] Gallery_canny[0.4, 0.5]



Gallery_canny[0.2, 0.3] Gallery_canny[0.2, 0.4]



Dog_canny[0.1, 0.4] Dog_canny[0.2, 0.3]

Dog_canny[0.3, 0.4] Dog_canny[0.2, 0.4]

Figure2. result of Canny edge detection

iv. Discussion

a) Non-maximum suppression is a kind of edge thinning technique, and its function lies in "thin" edge. After the gradient calculation, the edge extracted based on the gradient value is still fuzzy. Non-maximum suppression can help suppress all gradient values other than the local maximum to 0. The algorithm for non-maximum suppression of every pixel in the gradient image is as follows: 1) compare the gradient intensity of the current pixel with the two pixels along the positive and negative gradient direction. 2) if the gradient intensity of the current pixel is the largest compared with the other two pixels, the pixel point shall be reserved as the edge point; otherwise, the pixel point shall be suppressed.

b) According to Canny's paper, the ideal radio of upper bound and lower bound is 2:1 or 3:1. In MATLAB, the function will default the lower bound is 0.4 * upper bound if only input the upper bound parameter.

c) According to the results, as for gallery picture, upper bound of 0.4 will produce the ideal edge of the picture, if upper bound is higher than 0.4, the edge cannot be consistent. Lower bound can be expected as 0.2, lower than this value will produce more tricky detail which is unnecessary.
The same as Dog picture, ideal threshold is [0.2, 0.4].

c) Structured Edge

i. Abstract and Motivation

This method is used to more precisely detect the edge of the image, which base on a data driven technology - machine learning and decision trees. The Random Forest classifier and BSDS dataset can apply to the method. First we should get the parameter of the image and also get the ground truth of parameters. Then we can start to train the dataset and improve the structured edge classifier base on decision tree and random forest. Finally the model of random forest is completed and normalize the image. If we want to thin the edge, we can apply the non-suppression maximum and also we can apply the threshold in the image. The flow chart can be depicted in next session.

ii. Approach and Procedures

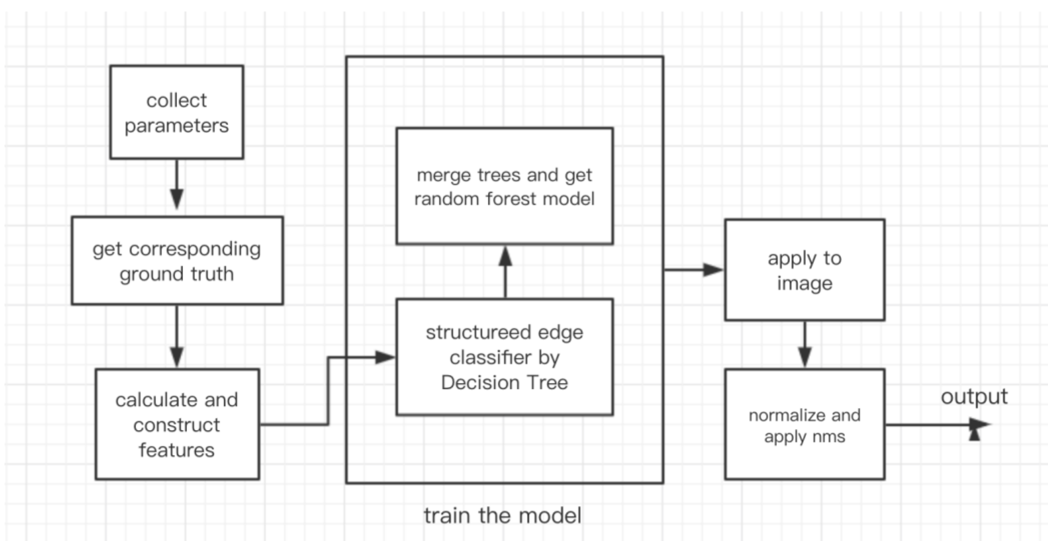
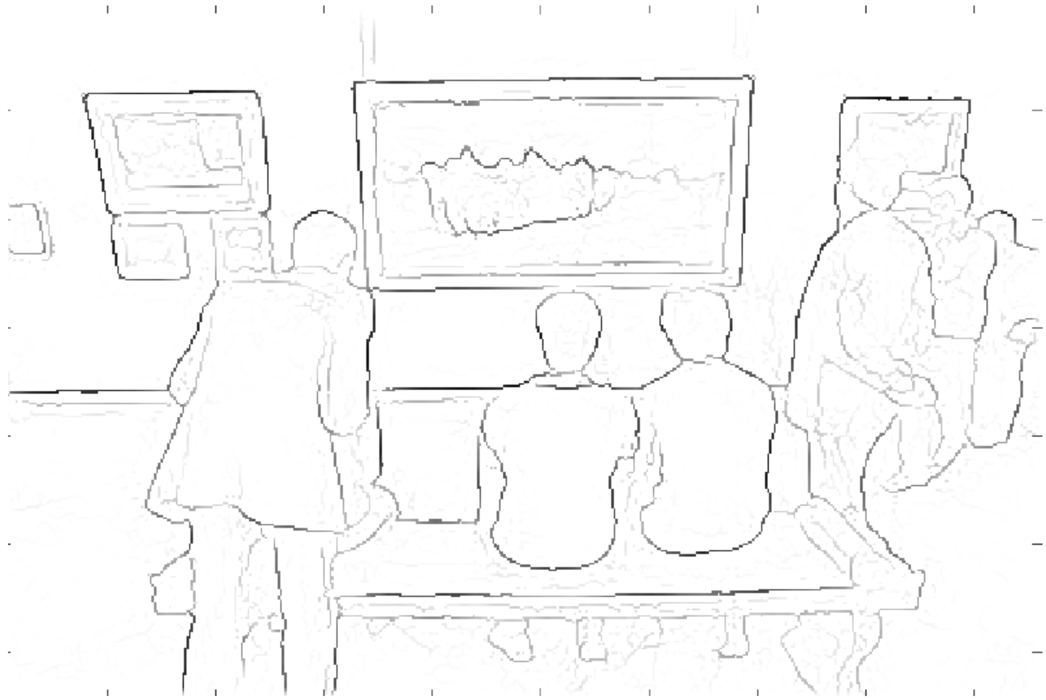


Figure3. flow charts

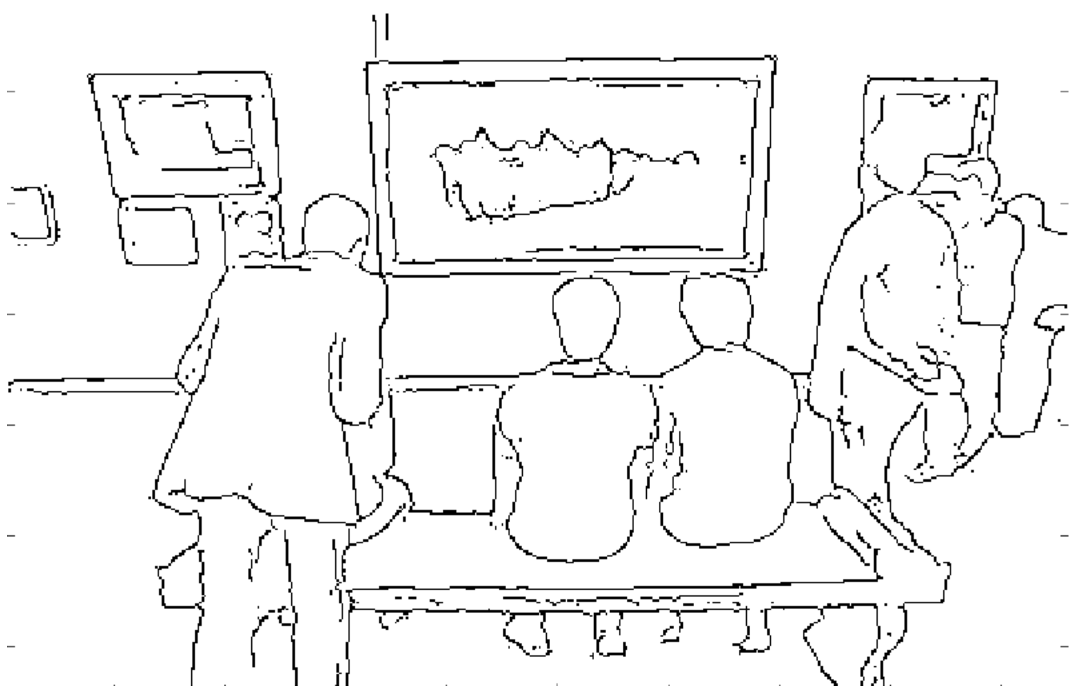
iii. Experimental Results

Dog_se.raw

Dog_se_NMS.raw (ts = 0.9)



Gallery_se.raw



Gallery_se_NMS.raw (ts = 0.9)

Figure4. Structured Edge

iv. Discussion

1） flow charts and explanation is on the former sessions.

2） In machine learning, a random forest is a classifier containing multiple decision trees, and the category of its output is determined by the mode of the category output by the individual tree. Decision trees work iteratively by breaking data into different subsets: for regression trees, they are a good choice to minimize MSE (mean square error) or MAE (mean absolute error) in all subsets. For the classification tree, the decomposition data is selected to minimize entropy or gini impurities in the resulting subset. The resulting classifier divides the feature space into different subsets. The process of determining characteristic contributions can of course be extended to random forests by averaging the contributions of all the trees in the forest. Random forest algorithm is a kind of parallel integrated learning method. Compared with decision tree, random forest introduces the selection of random attributes into the training of decision tree. Randomness is mainly reflected in the following two aspects: a) bootstrap sample method was adopted for the original data set. M sample data sets were sampled randomly for m times to obtain the sample set, and N times of such operation resulted in N sample sets containing m training samples. In this way, N base learners could be trained and combined to obtain the final learner. b) randomness is introduced into the selection of features. K features are randomly selected from K, and the optimal one is selected from K at each split.

3) The parameter apply in this session is following:

To do binary edge map, the threshold I used is 0.9, which p = 0.1.

```
model.opts.multiscale=0;        % for top accuracy set multiscale=1
model.opts.sharpen=2;           % for top speed set sharpen=0
model.opts.nTreesEval=4;        % for top speed set nTreesEval=1
model.opts.nThreads=4;          % max number threads for evaluation
model.opts.nms=1;               % set to true to enable nms
```

The results comparing to the canny is that structured edge can be completer and more consistent, there are also many details with undertint. The main edges are of border style. After binary edge selection, map can be more better than canny while canny is hard to preserve both consistent edge and details in picture.

d) Performance Evaluation

(1)

1) Dog_sobel

Ground Truth	Mean Recall	Mean Precision	Mean F-measure	Best
G1	0.1422	0.3177	0.1963	0.3810
G2	0.1644	0.2270	0.1906	
G3	0.1667	0.3076	0.2162	
G4	0.1291	0.3133	0.1830	
G5	0.2860	0.3377	0.3100	

2) Gallery_sobel

Ground Truth	Mean Recall	Mean Precision	Mean F-measure	Best
G1	0.2485	0.6020	0.4165	0.7191
G2	0.2522	0.2930	0.4449	
G3	0.2711	0.4513	0.4045	
G4	0.2021	0.5915	0.4327	
G5	0.5958	0.7614	0.4312	

3) Dog_Canny(0.2 – 0.4)

Ground Truth	Mean Recall	Mean Precision	Mean F-measure	Best
G1	0.6058	0.2496	0.3535	0.6443
G2	0.3192	0.2733	0.2945	
G3	0.4709	0.2823	0.3530	
G4	0.5891	0.2035	0.3025	
G5	0.7345	0.5736	0.6442	

4) Gallery_Canny(0.2 – 0.4)

Ground Truth	Mean Recall	Mean Precision	Mean F-measure	Best
G1	0.5741	0.6237	0.6058	0.6762
G2	0.6268	0.6077	0.3192	
G3	0.5651	0.5833	0.4709	
G4	0.6876	0.6585	0.5891	
G5	0.5750	0.8200	0.7345	

5) Dog_SE

Ground Truth	Mean Recall	Mean Precision	Mean F-measure	Best
G1	0.7010	0.3166	0.4363	0.7199
G2	0.7327	0.1938	0.3035	
G3	0.7092	0.2371	0.3554	
G4	0.6600	0.3283	0.4385	
G5	0.8647	0.2714	0.4141	

6) Gallery_SE

Ground Truth	Mean Recall	Mean Precision	Mean F-measure	Best

G1	0.8074	0.4429	0.5721	0.8500
G2	0.7823	0.4638	0.5824	
G3	0.7404	0.4385	0.5705	
G4	0.7674	0.4656	0.5705	
G5	0.9104	0.4132	0.5684	

Sobel:

Pros: Sobel operator is relatively easy to implement in space, which can not only produce better edge detection, but also reduce the effect of noise on the image by local average algorithm. The advantages of Sobel edge detection are simple calculation and fast speed. It is an effective measure when the precision is not high.

Cons: Because the processing of gradient and amplitude is relatively simple, Sobel is weak in processing the boundary of complex texture. The application of threshold is not fine enough, resulting in noise and false detection.

Canny:

Pros: Two different thresholds are used to detect the strong edge and the weak edge respectively, and the weak edge is included in the output image when the weak edge is connected with the strong edge. The resulting image outline is more as expected, and many invalid weak edges are discarded.

Cons: Due to the double edge selection, weaker edge can make the loss of more edge we have wanted. Canny also needs people to adjusts its thresholds by craft, which is hard to get a most satisfied result.

Structured Edge:

Pros: We can easily get the result that Structured Edge can most recover the edge we desire and is better than sobel and canny. The images it gets have better robustness and a greater improvement of noise. Edge can be clearer and hardly lose the details.

Cons: Need extra operation of NMS. What's more, the method requires more time to consume and need more manually adjusted parameters.

(2) Gallery can easier to get higher F-measure. As for Dogs.raw we can see there is much grass beside the dogs which will produce a disturbed edge. Gallery.raw can be more regular and the edge is more clear so Gallery is much better than the Dogs.

(3) We cannot get the get a high F measure if precision is significantly higher than recall, or vice versa. F, Recall, Precision has a relationship as follows:

$$\frac{2}{F} = \frac{1}{R} + \frac{1}{P}$$

If we set the P is larger than R, then the right side of equation will become larger due to the smaller of R, vice versa. Thus, the F will become smaller.

Supposed that c = P + R, then F = 2 * (c - R) * R / c.

Then we can ask the derivative of the right side dR/dc. Finally we get that R = c / 2 then F is at the peak. As the result, R = P, F get its maximum value.

2. Problem 2: Digital Half_toning
   a) Dithering
      i. Abstract and Motivation

         The gray value of each pixel is used to generate the black and white pixel points for semi-toning, so that the gray value originally occupying 8 bits is converted to 0 and 1, thus reducing the image storage. This makes sense for the graphics industry.
      ii. Approach and Procedures
         a) fixed threshold

            In this method, we compare each grayscale of the pixel in picture to the fixed threshold, in this program, we set it up as 128.
         b) random threshold

            In this method, we compare each grayscale of the pixel in picture to a random threshold, which is generated by the random number generator in MATLAB.
         c) dithering matrix

            First, we should consider about the input number N, which can decide the output of the picture. If N = 2, we just get the original matrix I. Otherwise, matrix can be calculated by the following formulation by log2(N) – 1 times.

            $$I = [(4*I+1)\ (4*I+2);\ (4*I+3)\ 4*I];$$

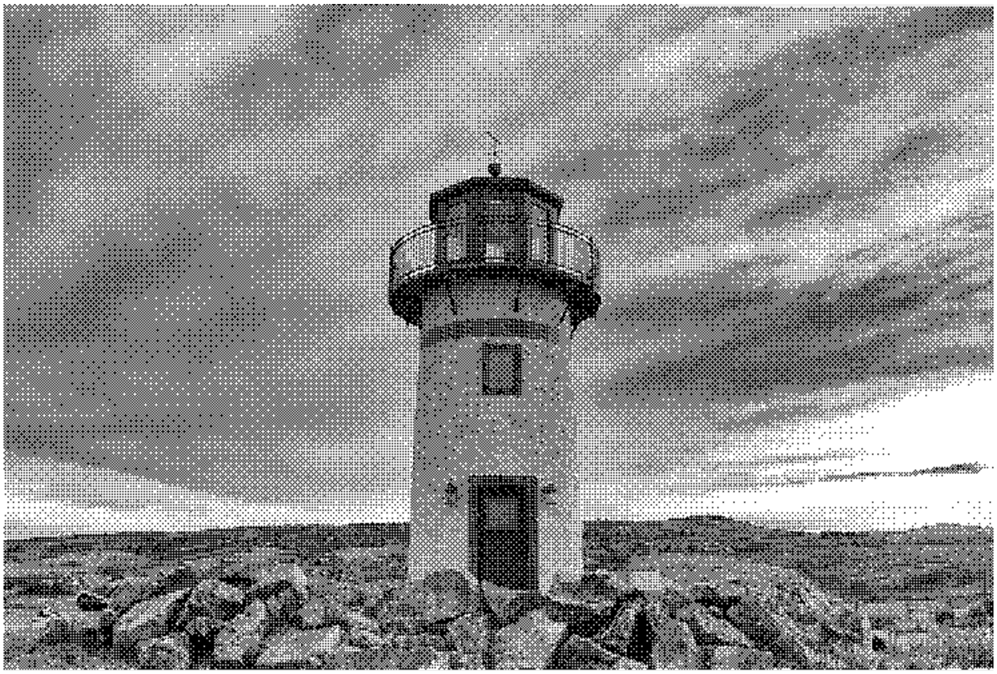
            Then we normalized the matrix I and get matrix T.

            we compare each grayscale of the pixel in picture to mod results of matrix T, and get the output pictures.
      iii. Experimental Results

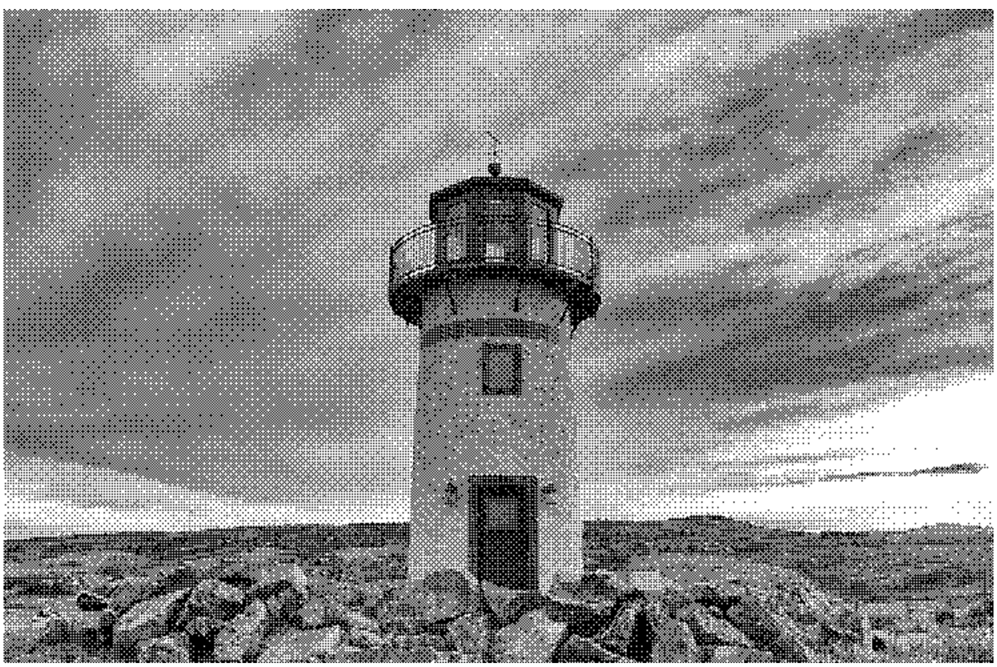
Fixed thresholding

Random thresholding

Dithering matirx I2



Dithering matirx I8



Dithering matirx I32

Figure4. dithering

iv. Discussion

Results are displayed.

b) Error Diffusion

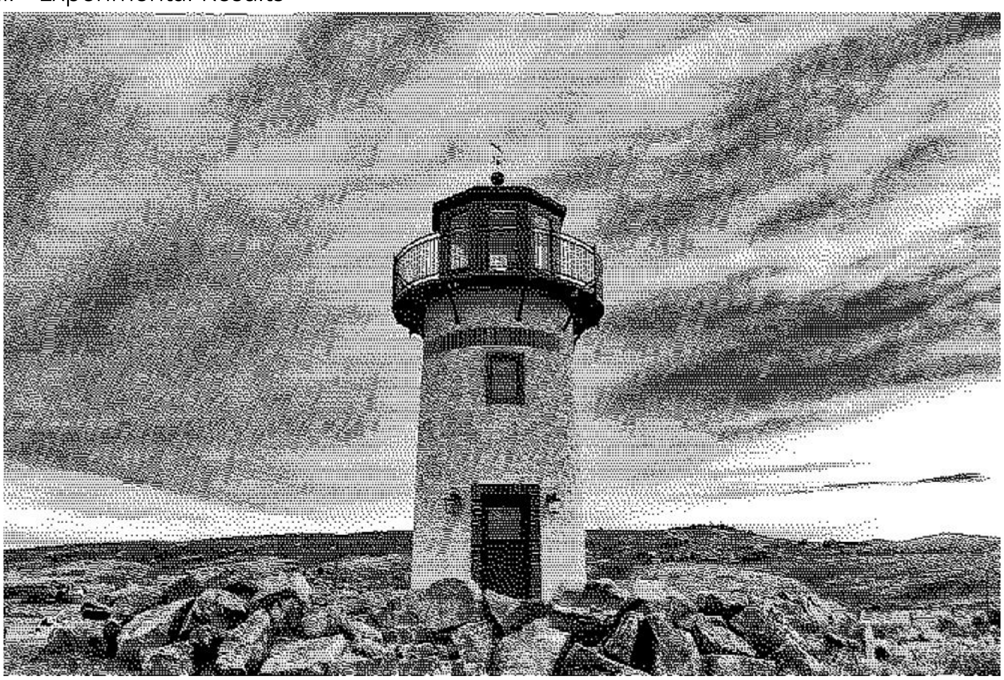
i. Abstract and Motivation

The differential diffusion algorithm transfers halftone mesh from "point processing" to "neighborhood processing". The introduction of the error diffusion algorithm has brought a revolutionary technological change to halftone mesh, which is also a milestone in halftone technology and promotes the rapid development of halftone technology. After the error diffusion processing, the half-tone image pixel distribution is different and irregular, the color is rich, and the visual effect is better. Until now, it has been widely used as one of the easy to implement and visually appealing halftone techniques.

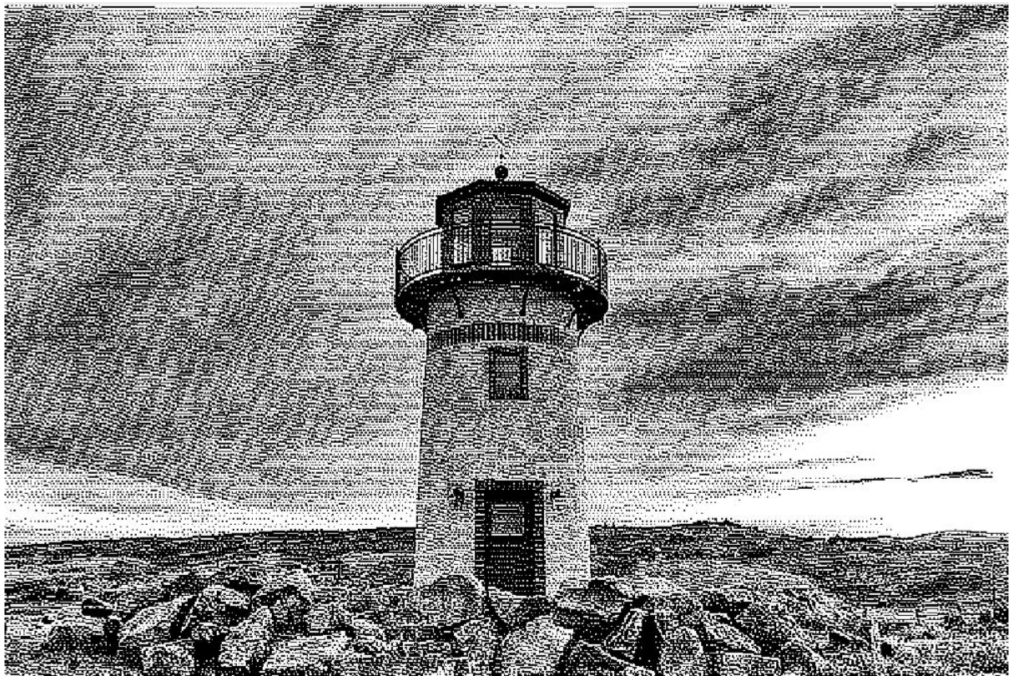
ii. Approach and Procedures

Given the threshold value t, let the original image be x(I, j) and the output be y(I, j). Perform two steps on the whole image in order from left to right and from top to bottom point by point: 1) Threshold y(I, j); 2) The quantization error is diffused to a nearby unprocessed point. Quantization error refers to the difference between the input and output. Take the Floyd-Steinberg's error diffusion as example, quantization error diffusion refers to changing the values of adjacent pixels x(I, j+1), x(I +I, j-1), x(I +1, j), and x(I +I, j+1) in space, transferring the quantization error of the current pixel to the adjacent pixel in a ratio of 7:3:5:1 and superimposed. (same to the other two error diffusion sample)

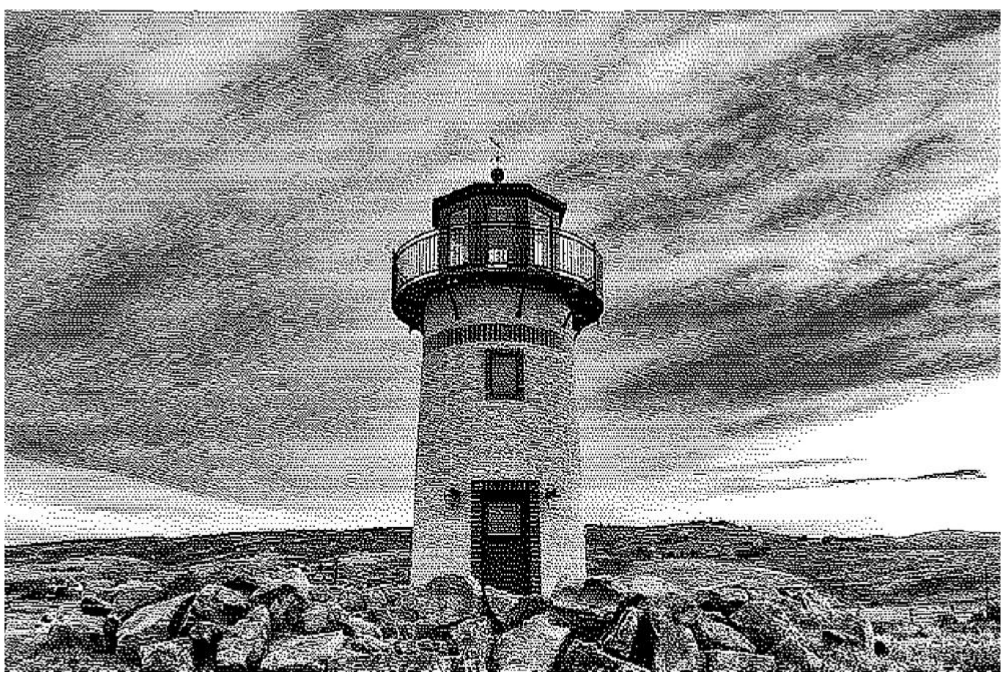
iii. Experimental Results



errorDiffusionFloyd



errorDiffusionJJN



errorDiffusionStucki

Figure5. result of error diffusion

iv. Discussion

In the results, we can see Floyd-Steinberg's error diffusion is obviously worse than the other two. However, we can hardly see some big difference between JJN and Stucki, which only shows some trivial difference on brightness.

Error diffusion is much better than the dithering because dithering only consider about the

properties of the single pixel but error diffusion cares about its neighborhood.
We can use the Hilbert curve to get a better result.

c) Color Halftoning with Error Diffusion

i. Abstract and Motivation

Same to the former section.

RGB channel's total bits can be transfer from 256^3 bits into 2^3 = 8bits, which is a large step of information storage.

ii. Approach and Procedures

a) Separable Error Diffusion

First, we should separate the image into 3 different matrixes and transfer RGB to CMY

Second, we should apply error diffusion to every matrix.

Third, we should merge rgb matrixes and transfer cmy to rgb.

b) MBVQ-based Error diffusion

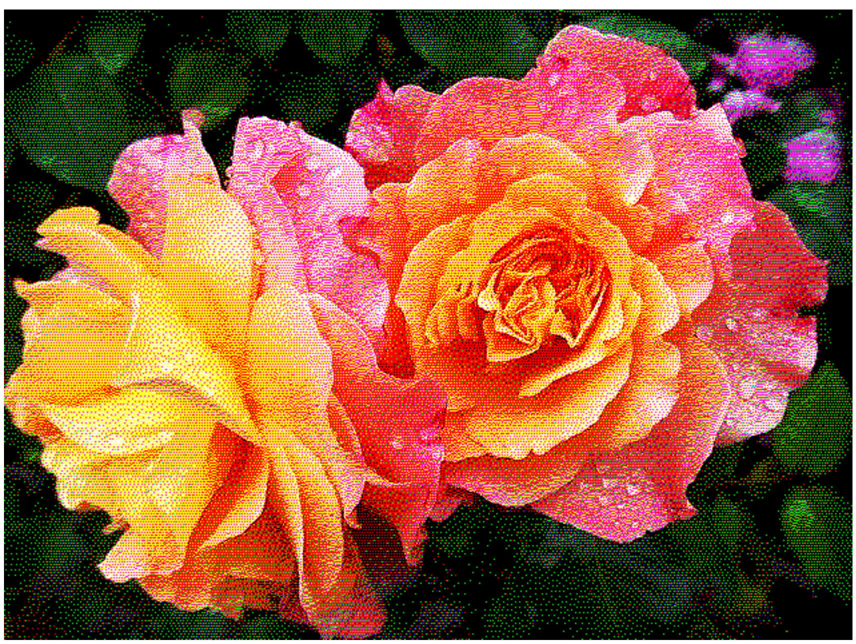
First, select the MBVQ mode of the each pixel in rgb channels.

Second, find the nearest vertex.

Third, calculate the current error of the rgb channels.

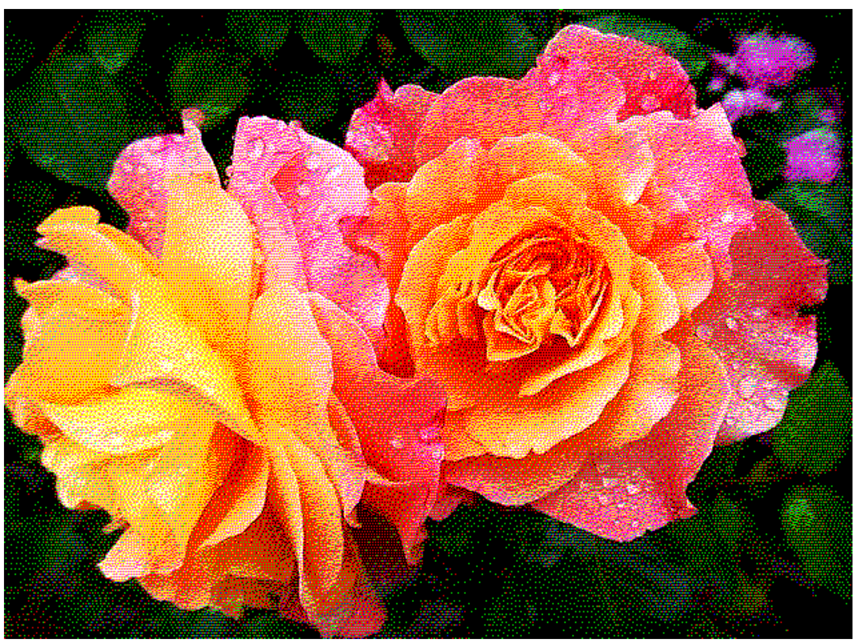
Forth, apply the error diffusion with selected matrix to calculate the error matrix.

iii. Experimental Results



rose_error_diffusion

rose_cmy_error_diffusion



rose_MBVQ

Figure6. result of Color Halftoning with error diffusion

iv. Discussion

1) The shortcoming of the separable error diffusion method will directly discard the dark pixels, so the picture looks darker.
2) MBVQ first select the mode of the pixel to decrease the range of 8 colors to 4 color vertices, which apply the color quadrants. By this way, the brightness variation can be minimum. Also, by calculate the error diffusion, the rgb channel in every pixels can be

induced into the closed vertex, which accurate the precision and decrease the halftoning noise. The dot pattern can be shown more regular.

3) We can see, in the result, MBVQ rose reflects more details of leaves in background which is relatively darker than the rose. Also, the rose can be milder and more closed to original picture than separable one.