ECE 637 Lab 8 - Image Halftoning

Yifan Fei

Electrical Engineering
Purdue University

Instructor: Prof. Charles A. Bouman

Spring 2018

Section 3: Thresholding and Random Noise Binarization

1: the original image and the result of thresholding

(a) the original image (b) the result of thresholding with T = 127

Figure 1: the original image and the result of thresholding

2: the computed RMSE and fidelity values

$$RMSE = 87.3933$$

$$fidelity = 77.3371$$

3:code for your fidelity function

Function of fidelity:

```
function fid = fidelity(f,b)
        f = double(f);
        b = double(b);

        f_l = 255 * (f / 255).^2.2;
        sigma = 2;
        [i, j] = meshgrid(-3:1:3, -3:1:3);
        h = exp(-(i.^2 + j.^2)/(2*sigma));
        h = h / sum(h(:));

        f_lh = conv2(f_l, h, 'same'); % the same size as f_l
        b_lh = conv2(b, h, 'same'); % the same size as b
        f_lt = 255 * (f_lh / 255).^(1/3);
        b_lt = 255 * (b_lh / 255).^(1/3);

        [M, N] = size(f_lt);
        fid = sqrt((1 / (N * M)) * sum(sum((f_lt - b_lt).^2)));

end
```

Section 3 main:

```
clear all;
```

```
clc;

img = imread('house.tif');
[M, N] = size(img);

graymap = [0:255;0:255;0:255]'/255;
colormap(graymap);

img_thres = zeros(size(img));
for i = 1:M
    for j = 1:N
        if img(i,j) > 127
            img_thres(i,j) = 255;
        end
    end
end

image(img_thres);
truesize;
imwrite(img_thres, 'house_thresholding.tif');

img_d = double(img);
img_thres_d = double(img_thres);

RMSE = sqrt((1 / (N * M)) * sum(sum((img_d - img_thres_d).^2)));

fid = fidelity(img, img_thres);
```

Section4: Ordered Dithering

1:The three Bayer index matrices

$$I_2 = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

$$I_4 = \begin{bmatrix} 5 & 9 & 6 & 10 \\ 13 & 1 & 14 & 2 \\ 7 & 11 & 4 & 8 \\ 15 & 3 & 12 & 0 \end{bmatrix}$$

$$I_8 = \begin{bmatrix} 21 & 37 & 25 & 41 & 22 & 38 & 26 & 42 \\ 53 & 5 & 57 & 9 & 54 & 6 & 58 & 10 \\ 29 & 45 & 17 & 33 & 30 & 46 & 18 & 34 \\ 61 & 13 & 49 & 1 & 62 & 14 & 50 & 2 \\ 23 & 39 & 27 & 43 & 20 & 36 & 24 & 40 \\ 55 & 7 & 59 & 11 & 52 & 4 & 56 & 8 \\ 31 & 47 & 19 & 35 & 28 & 44 & 16 & 32 \\ 63 & 15 & 51 & 3 & 60 & 12 & 48 & 0 \end{bmatrix}$$

2:The three halftoned images

Figure 2: Halftoned image With 2 Bayer index matrix

Figure 3: Halftoned image With 4 Bayer index matrix

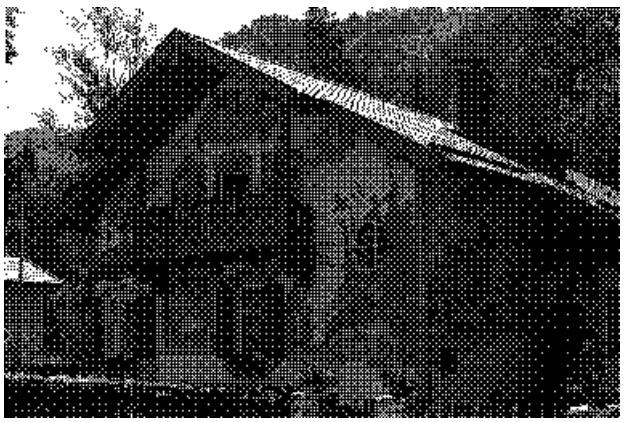

Figure 4: Halftoned image With 8 Bayer index matrix

3: The RMSE and fidelity values

Table 1: **The RMSE and fidelity values**

	RMSE	**fidelity**
I2 Dithering	97.6690	50.0569
I4 Dithering	101.0069	16.5583
I8 Dithering	100.9145	14.6918

4: Code

```
clear all;
clc;

I2 = [1 2;3 0];
I4 = [4*I2 + 1, 4*I2 + 2; 4*I2 + 3, 4*I2];
I8 = [4*I4 + 1, 4*I4 + 2; 4*I4 + 3, 4*I4];

img = imread('house.tif');
f_l = 255 * (double(img) / 255).^(2.2);
[M, N] = size(img);

N2 = 2; N4 = 4; N8 = 8;
T2 = 255 * ((I2 + 0.5) / (N2^2));
T4 = 255 * ((I4 + 0.5) / (N4^2));
T8 = 255 * ((I8 + 0.5) / (N8^2));

f_l_2 = zeros(M,N); f_l_4 = zeros(M,N); f_l_8 = zeros(M,N);

for p = 1:M
        for q = 1:N
                i = mod(p-1,N2) + 1;
                j = mod(q-1,N2) + 1;
                if f_l(p,q) > T2(i,j)
                        f_l_2(p,q) = 255;
                else
                        f_l_2(p,q) = 0;
                end
        end
end

for p = 1:M
        for q = 1:N
                i = mod(p-1,N4) + 1;
                j = mod(q-1,N4) + 1;
                if f_l(p,q) > T4(i,j)
                        f_l_4(p,q) = 255;
                else
                        f_l_4(p,q) = 0;
                end
        end
end
```

```
for p = 1:M
        for q = 1:N
                i = mod(p-1,N8) + 1;
                j = mod(q-1,N8) + 1;
                if f_1(p,q) > T8(i,j)
                        f_1_8(p,q) = 255;
                else
                        f_1_8(p,q) = 0;
                end
        end
end

imwrite(f_1_2, 'house_N2.tif');
imwrite(f_1_4, 'house_N4.tif');
imwrite(f_1_8, 'house_N8.tif');

img_d = double(img);
RMSE_2 = sqrt((1 / (N * M)) * sum(sum((img_d - f_1_2).^2)));
RMSE_4 = sqrt((1 / (N * M)) * sum(sum((img_d - f_1_4).^2)));
RMSE_8 = sqrt((1 / (N * M)) * sum(sum((img_d - f_1_8).^2)));

fid_2 = fidelity(img_d, f_1_2);
fid_4 = fidelity(img_d, f_1_4);
fid_8 = fidelity(img_d, f_1_8);
```

Section5: Error Diffusion

1:error diffusion Matlab code

```
function [f_diff] = error_Diffusion(f)
        f_1 = 255 * (double(f) / 255).^2.2;
        f_diff = padarray(f_1, [1 1]); % zero pad

    [m,n] = size(f_diff)
        for i = 2:m - 1
                for j = 2:n - 1
                        f_orig = f_diff(i,j);
                        f_diff(i,j) = (f_diff(i,j) > 127) * 255;
                        error = f_orig - f_diff(i,j);
                        f_diff(i,j+1) = f_diff(i,j+1) + error*(7/16);
                        f_diff(i+1,j+1) = f_diff(i+1,j+1) + error*(1/16);
                        f_diff(i+1,j) = f_diff(i+1,j) + error*(5/16);
                        f_diff(i+1,j-1) = f_diff(i+1,j-1) + error*(3/16);
                end
        end

        f_diff = f_diff(2:end-1, 2:end-1);

end
```

Section 5 code:

```
clear all;
```

```
clc;

img = imread('house.tif');
img_diff = error_Diffusion(img);
[M, N] = size(img);

graymap = [0:255;0:255;0:255]'/255;
colormap(graymap);
image(img_diff);
truesize;
imwrite(img_diff, 'house_errdiff.tif');

RMSE = sqrt((1 / (N * M)) * sum(sum((double(img) - img_diff).^2)));
fid = fidelity(double(img), img_diff);
```

2:error diffusion result

(a) the original image (b) error diffusion result

Figure 5: the original image and the error diffusion result

3:RMSE and fidelity of the error diffusion result

$$RMSE = 98.8471$$
$$fidelity = 13.4273$$

4: the RMSE and fidelity for different methods

Table 2: **The RMSE and fidelity values**

	RMSE	**fidelity**
thresholding	87.3933	77.3371
I2 Dithering	97.6690	50.0569
I4 Dithering	101.0069	16.5583
I8 Dithering	100.9145	14.6918
error diffusion	98.8471	13.4273

By comparison, we can find that the RMSE values are close to each other in these methods, but the fidelity values vary much.

Though the observed visual quality of the threshold is not good, it has the highest fidelity value. The observed visual quality of dithering from 2 by 2 Bayer Index Matrix to 8 by 8 Bayer Index Matrix is getting worse, while the fidelity values decrease. The observed visual quality of error diffusion result is not that bad, but it has the lowest fidelity value. In conclusion, the theory value does not reflect visual quality to some degree.