ECE 63700 Laboratory:

Image Halftoning

Shang Hsuan (Sean) Lee

April 10, 2024

1 Introduction

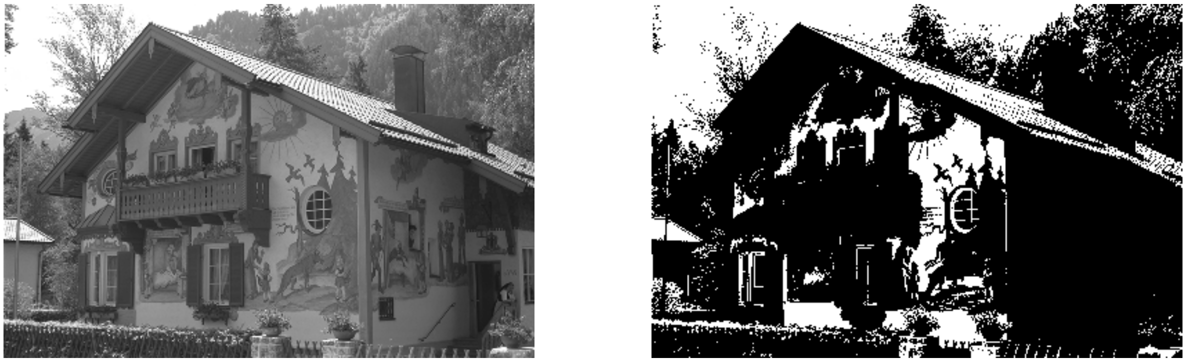
No reports needed.

2 Image Fidelity Metrics

No reports needed.

3 Thresholding and Random Noise Binarization

3.1 Original Image vs Binary Image



(a) house.tif

(b) Thresholded house.tif with T = 127

Figure 1: Original Image vs Binary Image

3.2 Computed RMSE and Fidelity Values

$$RMSE = 87.3933$$
$$fidelity = 77.3371$$

3.3 Code Listing for Fidelity Function

```
def rmse(f, b):
    f = f.astype(float)
    b = b.astype(float)

    rows, cols = np.shape(f)
    sum = 0
    for i in range(rows):
        for j in range(cols):
            sum += (f[i,j] - b[i,j])**2

    return np.sqrt((1/(rows*cols)) * sum)


def fidelity(f, b):
    gamma = 2.2
    f_l = 255 * ((f/255) ** gamma)
    b_l = 255 * ((b/255) ** gamma)

    # define h
    var = 2
    size = 7
    h = np.zeros((size, size))
    for i in range(size):
        for j in range(size):
            h[i,j] = np.exp(-((i-3)**2 + (j-3)**2) / (2*var))

    C = 1/np.sum(h)
    h = C*h

    # pass f and b through h
    filtered_f = convolve2d(f_l, h, mode='same', boundary='fill', fillvalue=0)
    filtered_b = convolve2d(b_l, h, mode='same', boundary='fill', fillvalue=0)

    filtered_f = 255 * (filtered_f/255) ** (1/3)
    filtered_b = 255 * (filtered_b/255) ** (1/3)

    fid = rmse(filtered_f, filtered_b)

    return fid
```

4 Ordered Dithering

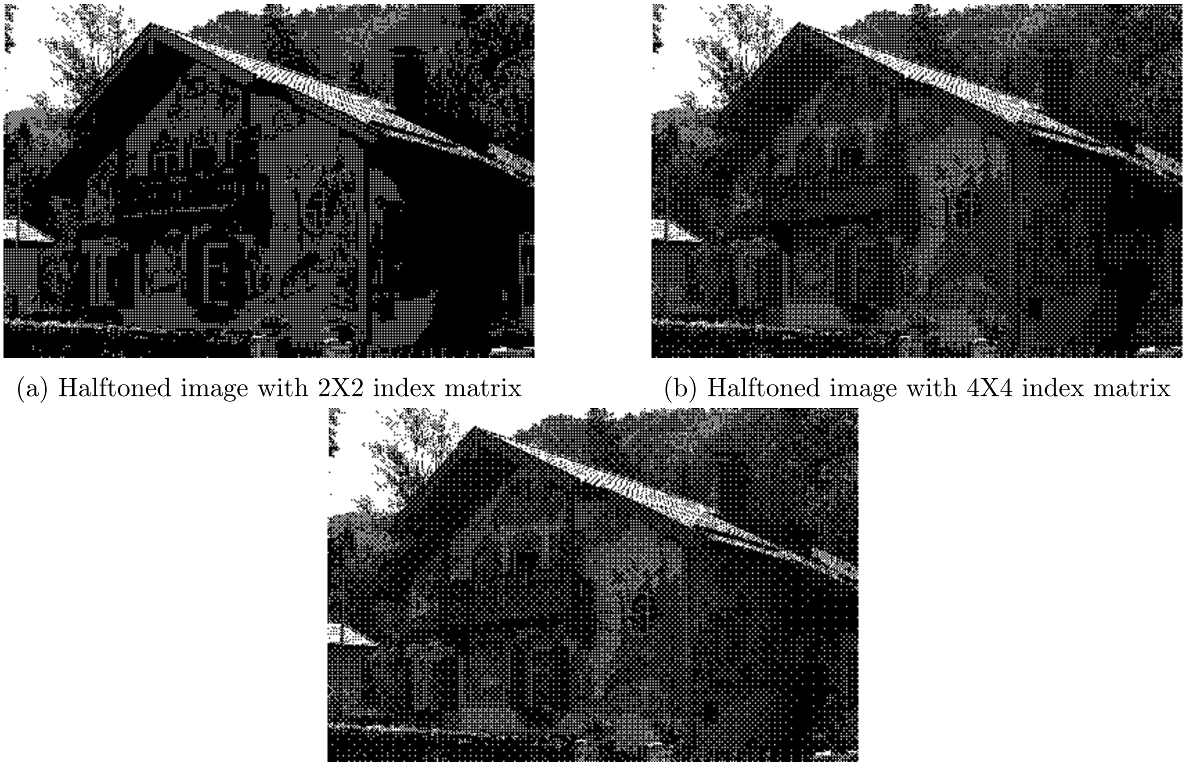
4.1 Bayer index matrices

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

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

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

4.2 Halftoned Images



(a) Halftoned image with 2X2 index matrix

(b) Halftoned image with 4X4 index matrix

(c) Halftoned image with 8X8 index matrix

Figure 2: Halftoned Images

4.3 RMSE and Fidelity for the Halftoned Images

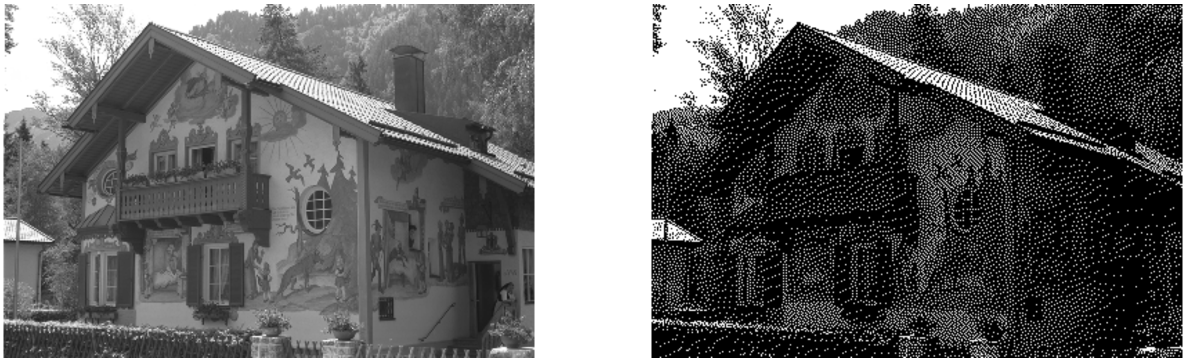
Dimension of Bayer index matrix	RMSE	Fidelity
2x2	97.6690	50.0569
4x4	101.0069	16.5583
8x8	100.9145	14.6918

5 Error Diffusion

5.1 Code Listing

```
def errorDiffusion(img_l):
    T = 127
    out = np.zeros(img_l.shape)
    img_l = np.pad(img_l, ((1, 1), (1, 1)))

    for i in range(1, img_l.shape[0]-1):
        for j in range(1, img_l.shape[1]-1):
            if img_l[i, j] > T:
                out[i-1, j-1] = 255
            else:
                out[i-1, j-1] = 0

            error = img_l[i, j] - out[i-1, j-1]

            img_l[i, j+1] += error * 7/16
            img_l[i+1, j-1] += error * 3/16
            img_l[i+1, j] += error * 5/16
            img_l[i+1, j+1] += error * 1/16

    return out
```

5.2 Error Diffusion Result



(a) house.tif

(b) Error Diffused house.tif

Figure 3: Error Diffusion Result

5.3 RMSE and Fidelity of the Error Diffusion Result

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

5.4 RMSE and Fidelity Comparison

Method	RMSE	Fidelity
Simple Thresholding	87.3933	77.3371
2x2 Dithering	97.6690	50.0569
4x4 Dithering	101.0069	16.5583
8x8 Dithering	100.9145	14.6918
Error Diffusion	98.8471	13.4273

5.5 Observations

Upon examining the comparison table, it's evident that the RMSE values show no significant differences across the various methods. However, there is a substantial variance in fidelity, with the simple thresholding method having the highest fidelity and the more complex method, such as error diffusion has the lowest score. When correlating the resulting images with their fidelity values, it can be inferred that a lower fidelity may correspond to better preservation of the original image's details.