Image and Video processing
Laboratory 2
Dithering

Prof. Touradj EBRAHIMI

Reports submission – !!! read carefully !!!

Students will have to produce and submit a report two weeks after completion of each laboratory session. An electronic version of the report is submitted via Moodle's interface for uploading assignments. There will be a hard deadline for each submission. Late submissions will have to be justified and explained to course assistant by email. A structure of the report is explained below. In addition to the report, the source code produced should also be submitted in electronic form. The report and the source code should be submitted in **ONE ZIP** file using Moodle platform.

Structure of the reports and m-files

- the name of the final ZIP file will be: `lab_`*number-of-lab_your-surname*`.zip`
- the final ZIP file will contain
  - final m-file called `run_lab_`*number-of-lab_your-surname*`.m`
  - all created m-files and functions which are necessary to obtain the results
  - all source images and data needed for successful running of the final m-file
  - an electronic version of your report in **PDF** format
- the structure of the final m-file
  - the final m-file will contain all necessary commands and functions, by running this m-file one must get all results[1]
  - submitted m-files will be commented
  - each figure will be properly titled
  - the final m-file will be divided into cells[2] according to the example bellow
- the parts of program codes should not be included in the final report
- description of the results should be a part of the final report
- don't forget to answer all the questions in your report
- check whether your final m-file can be launched without problems - only then submit your report

---

[1] of course it is upon you whether you want to include everything in the final m-file, or create different functions which you will call in the final m-file

[2] those who do not know how to use the cells in Matlab to create the program code more transparent and easier to read, they should look at `http://www.mathworks.com/demos/matlab/developing-code-rapidly-with-cells-matlab-video-tutorial.html`

An example of final m-file

```
% Lab (number of lab) - your name and date

%% Exercise (number of exercise) - "Name of exercise"

% your own program code with your coments
a = imread('picture.tif');
% all figures will be properly titled
figure('name','Name Of The Figure')
imshow(a,[]);

%% Exercise (number of exercise) - "Name of exercise"

% your own program code with your coments
[output1, output2, ..., outputN] = function_name (input1, input2, ..., inputN);
```

1 Fixed threshold method

Implement the dithering method based on fixed thresholding. All pixels with intensity less or equal to half the dynamic range are assigned to black and the remaining to white. Apply this method to `lena-y.png` and `wool.png`.

2 Random threshold method

Repeat the exercise by adding a discrete uniform noise to the image (hint: use the function `unidrnd`). Choose the amplitude of the noise so as to achieve a good compromise between the rendered gray levels and the loss of quality due to the added noise.

3 Ordered threshold method

The ordered threshold method is given by a matrix of threshold values used to pave the image plane. An example is given in the following clustered dot matrix

$$S = \begin{bmatrix} 34 & 29 & 17 & 21 & 30 & 35 \\ 28 & 14 & 9 & 16 & 20 & 31 \\ 13 & 8 & 4 & 5 & 15 & 19 \\ 12 & 3 & 0 & 1 & 10 & 18 \\ 27 & 7 & 2 & 6 & 23 & 24 \\ 33 & 26 & 11 & 22 & 25 & 32 \end{bmatrix}.$$

By convention the smallest threshold is always zero. The gray levels that a threshold matrix can represent, in a normalized scale $[0, 1]$, are $i/(N-1)$, $i = 0, \ldots, N-1$. In the previous example $N = 37$ gray levels.

The image is first quantized to N gray levels where N is the number of gray levels represented by the matrix. Then, the quantized image is compared to the thresholds of the matrix. In other terms, the dithering is obtained by the relation

$$\hat{x}(k,l) > s(k,l), \tag{1}$$

where $\hat{x}(k,l)$ is the quantized image and $s(k,l)$ is the matrix obtained by paving the image plane with the matrix of thresholds. Note that $\hat{x}(k,l)$ takes the integer values between 0 and $N-1$.

Find the adequate quantization function $Q(x)$ to convert $x(k,l)$ to $\hat{x}(k,l)$. Find an equivalent expression for $Q(x)$ that does not require operators such as `floor`, `ceil` or `round` (in other words, only using addition, substraction, multiplication or division). With this result, write a Matlab function that performs the thresholding. Set the image and the matrix of threshold values as inputs to the function.

4 Ordered matrix with centered points

$$C_6 = \begin{bmatrix} 34 & 25 & 21 & 17 & 29 & 33 \\ 30 & 13 & 9 & 5 & 12 & 24 \\ 18 & 6 & 1 & 0 & 8 & 20 \\ 22 & 10 & 2 & 3 & 4 & 16 \\ 26 & 14 & 7 & 11 & 15 & 28 \\ 35 & 31 & 19 & 23 & 27 & 32 \end{bmatrix}$$

is a matrix with a central white point over a black background (In other words, white points whose size depends on the gray level are drawn).

$$E_6 = \begin{bmatrix} 30 & 22 & 16 & 21 & 33 & 35 \\ 24 & 11 & 7 & 9 & 26 & 28 \\ 13 & 5 & 0 & 2 & 14 & 19 \\ 15 & 3 & 1 & 4 & 12 & 18 \\ 27 & 8 & 6 & 10 & 25 & 29 \\ 32 & 20 & 17 & 23 & 31 & 34 \end{bmatrix}$$

is a matrix called balanced centered point.

Apply these two matrices to both images. See the shape of the white and black points in the dithered images. What is the main difference between these two methods?

5 Diagonal ordered matrix with balanced centered points

$$O_8 = \begin{bmatrix} O_{8,1} & O_{8,2} \\ O_{8,2} & O_{8,1} \end{bmatrix},$$

where

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

and

$$O_{8,2} = \begin{bmatrix} 18 & 22 & 26 & 19 \\ 25 & 30 & 31 & 23 \\ 21 & 29 & 28 & 27 \\ 17 & 24 & 20 & 16 \end{bmatrix},$$

is a matrix called diagonal ordered matrix with balanced centered point. Note that the element paving the image plane is rhombus-shaped:

$$\begin{array}{ccccccc} & & & 18 & & & \\ & & 8 & 25 & 30 & & \\ & 3 & 4 & 21 & 29 & 28 & \\ 7 & 11 & 15 & 17 & 24 & 20 & 16 \\ 22 & 26 & 19 & 13 & 9 & 5 & 12 \\ & 31 & 23 & 6 & 1 & 0 & \\ & & 27 & 10 & 2 & & \\ & & & 14 & & & \end{array}$$

Apply O_8 to both images. Compare the results with those obtained in the previous exercise. What do you notice regarding eye sensitivity to diagonal and horizontal/vertical orientations?

6 Ordered matrix with dispersed dots

Bayer proposed a recursive algorithm to create a threshold matrix with a sparse pattern. With a threshold matrix D_n of dimension $n \times n$ one obtains the matrix $2n \times 2n$

$$D_{2n} = \begin{bmatrix} 4D_n & 4D_n + 2U_n \\ 4D_n + 3U_n & 4D_n + U_n \end{bmatrix},$$

where U_n is the $n \times n$ matrix with all the elements equal to 1. The matrices obtained with this algorithm minimize the occurrence of low frequencies, more visible to the eye. Matrices with basis

$$D_2 = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$

or

$$D_3 = \begin{bmatrix} 8 & 4 & 5 \\ 3 & 0 & 1 \\ 7 & 2 & 6 \end{bmatrix}$$

are usually used.

Use the matrix with sparse thresholds D_6 with both images.

7 Error diffusion method

The principle of this method is summarized in Figure 1. Each pixel in the image is thresholded. Then, the error introduced by the thresholding is calculated $(\mathrm{err}(i,j) = \mathtt{in}(i,j) - T(\mathtt{in}(i,j)))$. Using the *diffusion filter* given in Figure 1, the dithering error is distributed on the neighboring pixels using the indicated proportions. The pixels are processed in a line-by-line basis. Apply the error diffusion method by using

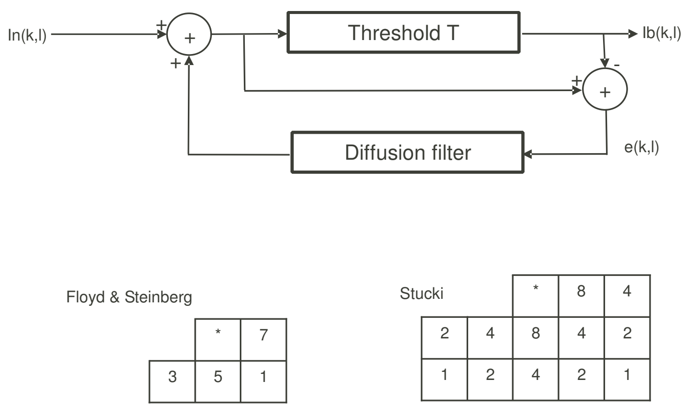

Figure 1: Block diagram of the error diffusion method.

a normalized Floyd-Steinberg filter, and then Stucki.

8 Discussion

Compare the various methods in terms of :

- Visual quality.
- Mean Square Error between the dithered image and the original.
- Complexity (number of operations).