UE 2-3 Image Processing

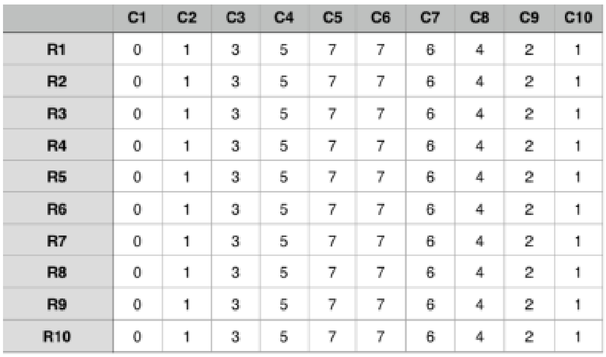
Labs 1 - Image Transformations

Submitted by **Meldrick REIMMER and Selma BOUDISSA**

Contents

1 Problem 1

Goal: calculate the size of the image below and give the number of bit used for that image.



	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
R1	0	1	3	5	7	7	6	4	2	1
R2	0	1	3	5	7	7	6	4	2	1
R3	0	1	3	5	7	7	6	4	2	1
R4	0	1	3	5	7	7	6	4	2	1
R5	0	1	3	5	7	7	6	4	2	1
R6	0	1	3	5	7	7	6	4	2	1
R7	0	1	3	5	7	7	6	4	2	1
R8	0	1	3	5	7	7	6	4	2	1
R9	0	1	3	5	7	7	6	4	2	1
R10	0	1	3	5	7	7	6	4	2	1

Figure 1: Original image

Here after the code that we create to solve the problem 1.

```
%% Problem1
% Goal: create a function that allow you to generate matrix
% file: myfunction

MATRIX =(myfunction([0 1 3 5 7 7 6 4 2 1],10));
d = size(MATRIX);% size [10,10]

imwrite(MATRIX,'problem1.jpg');
m=imread('problem1.jpg');
imshow(m);
```

prob1.m

We also create a function called "myfunction" to calculate the size of the image.

```
function [ A ] = myfunction( r, len )
A=zeros(len, len);
for i =1:len % from 1 to 3 do
  for j = 1:len %from 1 to 4 do
    A(j, i)= r(i);
  end
end
end
```

myfunction.m

The image is a matrix of 10 x 10, containing 10 rows and 10 columns. So the size of the image is 100 pixels. The number of bits used for that image is 4 bits.

2 Problem 2

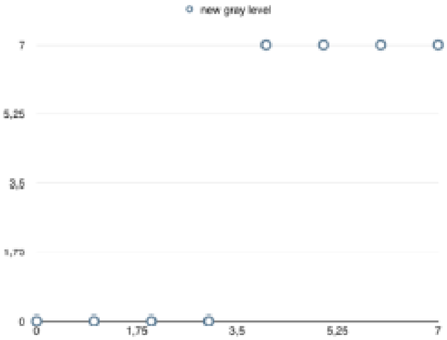


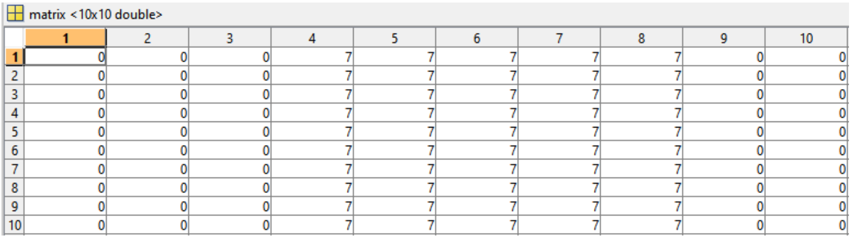
Figure 2: Input Image

This transformation is about the usage of a threshold operation. If the pixel value is greater than 4, then set it at 7, otherwise set it at 0.

```
%% Problem 2
img =(myfunction([0 1 3 5 7 7 6 4 2 1],10));
img2 = myfunction2(img,10,10);
imwrite(img2,'abc.jpg');
a = imread('abc.jpg');
imshow(a);
```

prob2.m

In the figure 3 we can see the result that we obtain with that gray level transformation it correspond to a binary image were each pixel value is 0 or 7.


matrix <10x10 double>

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	7	7	7	7	7	0	0
2	0	0	0	7	7	7	7	7	0	0
3	0	0	0	7	7	7	7	7	0	0
4	0	0	0	7	7	7	7	7	0	0
5	0	0	0	7	7	7	7	7	0	0
6	0	0	0	7	7	7	7	7	0	0
7	0	0	0	7	7	7	7	7	0	0
8	0	0	0	7	7	7	7	7	0	0
9	0	0	0	7	7	7	7	7	0	0
10	0	0	0	7	7	7	7	7	0	0

Figure 3: Result of the gray level transformation

3 Problem 3

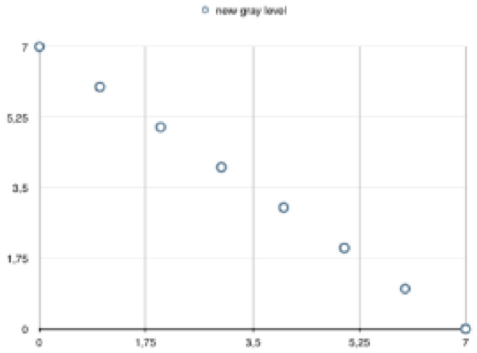


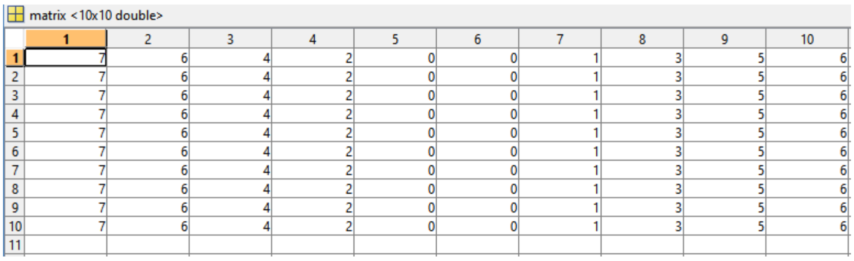
Figure 4: Input Image

The code below gives the transformation, changing each pixels value.

```
%% Problem 3
matrix = LoadMatrix(); % loadMatrix is a function to load the image called problem1
matrix = arrayfun(@(x) 7-x,matrix);
```

prob3.m

Here after in the figure 5 the result we obtain with a grey level transformation of the previous image.



	1	2	3	4	5	6	7	8	9	10
1	7	6	4	2	0	0	1	3	5	6
2	7	6	4	2	0	0	1	3	5	6
3	7	6	4	2	0	0	1	3	5	6
4	7	6	4	2	0	0	1	3	5	6
5	7	6	4	2	0	0	1	3	5	6
6	7	6	4	2	0	0	1	3	5	6
7	7	6	4	2	0	0	1	3	5	6
8	7	6	4	2	0	0	1	3	5	6
9	7	6	4	2	0	0	1	3	5	6
10	7	6	4	2	0	0	1	3	5	6
11										

Figure 5: Result of the gray level transformation

4 APPENDIX

4.1 Code for problem 1

```
%% LAB 1 - Image transformations
% UE2-3 Image Processing
% Group: Meldrick Reimmer and Selma Boudissa

%% basic command
clc % clear command window
close all % close all the figures
clear all % clear all the variables


%% Problem1
% Goal: create a function that allow you to generate matrix
% file: myfunction

MATRIX =(myfunction([0 1 3 5 7 7 6 4 2 1],10));
d = size(MATRIX);% size [10,10]

imwrite(MATRIX,’problem1.jpg’);
m=imread(’problem1.jpg’);
imshow(m);
```

problem1.m

4.2 Code for problem 2

```
%% LAB 1 - Image transformations
% UE2-3 Image Processing
% Group: Meldrick Reimmer and Selma Boudissa

%% basic command
clc % clear command window
close all % close all the figures
clear all % clear all the variables

%% Problem 2
img =(myfunction([0 1 3 5 7 7 6 4 2 1],10));
img2 = myfunction2(img,10,10);
imwrite(img2,’abc.jpg’);
a = imread(’abc.jpg’);
imshow(a);
```

problem2.m

4.3 Code for problem 3

```matlab
%% LAB 1 - Image transformations
% UE2-3 Image Processing
% Group: Meldrick Reimmer and Selma Boudissa

%% basic command
clc % clear command window
close all % close all the figures
clear all % clear all the variables

%% Problem 3
matrix = LoadMatrix(); % loadMatrix is a function to load the image called problem1
matrix = arrayfun(@(x) 7-x,matrix);
```

problem3.m