Video Processing

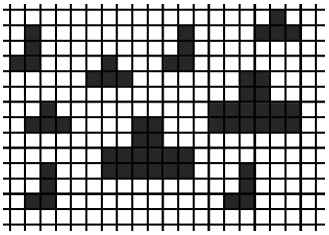
Date: 09/02/2016

Ex.1.[5 Pt] The following values represent the intensities of an image portion. For the central part (inside the thicker border) evaluate the Local Binary Pattern.

207	161	244	244	107
230	160	246	123	233
240	71	40	204	202
232	139	247	36	244

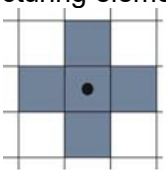
Es.2. [6 pt] Define the Huffman code to encode the following symbols A,B,C,D,E,F that have the following occurrences in a file:
A:34, B:12, C:27, D:3, E:20, F:30

Es.3. [11 pt] Describe a procedure that, using **only** morphological operators, is able to count objects in the scene and to cluster objects of the same size: the shape of each object can be assumed as already known.

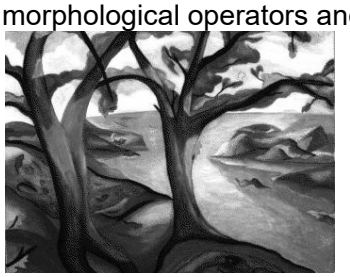


Es.4. [11 pt] Given a grayscale image in the file *trees_gray.bmp* with 256 levels of gray, and a binary image in the file *trees_bw.bmp*, implement in MATLAB the following operations:

a) Read, load and visualize the images;
b) Knowing that the binary image was obtained from the grayscale image using a certain black and white threshold, analyze the image histogram(s) to find the threshold value (hint: search for the gray value that partitions the histogram of the grayscale image according to the number of black/white pixels in the binary image);
c) Add salt & pepper noise to the grayscale image (use 0.1 as the noise density) and convert it to black and white image using the threshold from point b);
d) Given the following structuring element



remove the effects of the noise using the morphological operators. Which operators should you use and why? Apply the suitable morphological operators and show the results.



trees_gray.bmp

trees_bw.bmp

Matlab
List of functions

```
figure
im2double
im2bw
rgb2gray
imdilate
imerode
imopen
imclose
imread
imresize
imrotate
imnoise
imshow
getimage
size
zeros
find
imhist
strel
```

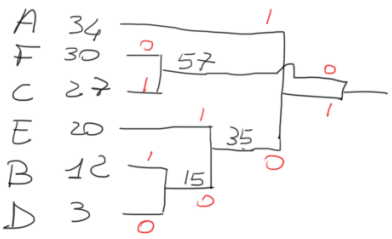
Solutions

Ex.1

Starting from the top left corner the values will be:

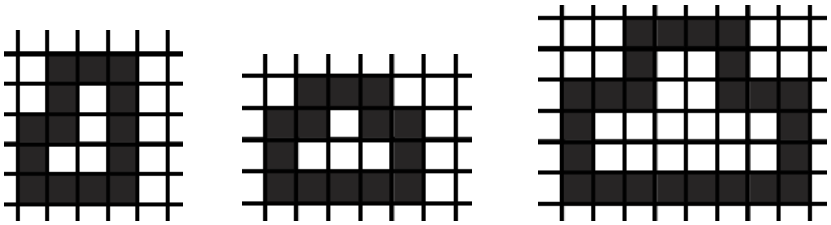
12	255	34
16	8	85

Ex.2



The coding becomes:
A: 11
F: 00
C: 01
E: 101
B: 1001
D: 1000

Ex.3 A possible solution is based on the **hit-or-miss** operator. Since we know the shape of each object we have, for each of them, to define the local background:



and then we have to apply the hit or miss operator: it will be the intersection between the erosion of the original image with a specific shape and its complementary with the local background

$$A \circledast B = (A \ominus X) \cap [A^c \ominus (W - X)]$$

counting the resulting points applying the local background for each shape will give the desired result for each them.

Ex.4

```
close all
clear all

% a)
Im_gray = imread('trees_gray.bmp');
figure, imshow(Im_gray);

Im_bw = imread('trees_bw.bmp');
figure, imshow(Im_bw);

% b)
blackPixels = sum(Im_bw(:) == 0);
% Alternatively:
% counts_bw = imhist(Im_bw);
% blackPixels = counts_bw(1);

counts = imhist(Im_gray);
accumulator = 0;
threshold = 0;
while(accumulator < blackPixels)
    threshold = threshold + 1;
    accumulator = accumulator + counts(threshold);
end
threshold = threshold/256;

% c)
Im_noise = imnoise(Im_gray,'salt & pepper', 0.1);
Im_noise_bw = im2bw(Im_noise,threshold);
figure, imshow(Im_noise_bw);

% d)
SE = strel('arbitrary',[0,1,0;1,1,1;0,1,0]);
% Opening should eliminate small details ("salt")
% Closing should eliminate small holes ("pepper")
Im_open = imopen(Im_noise_bw,SE);
figure, imshow(Im_open);
Im_open_close = imclose(Im_open,SE);
figure, imshow(Im_open_close);
```