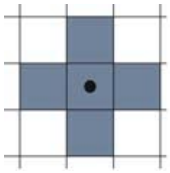
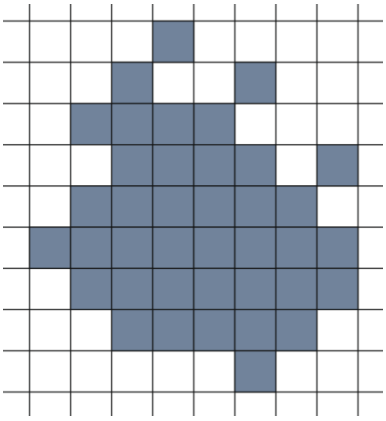
Video Signals

Date: 02/02/2017

Ex.1.[14 Pt] Given the black (value 1) and white (value 0) image on the right, extract its skeleton using the following structuring element:



In particular, for every skeletonization step k, provide the skeleton representation $S_k(A)$ and then provide the iterative steps to reconstruct the original object from the skeleton representing the results at each iteration.



Es.2. [7pt] We need to extract horizontal edges from a gray scale image when, moving from top to bottom we encounter a light to dark transition as depicted below:

Describe a possible filter to get exactly this result, the operations that have to be applied to the original image and what constraints should be applied.

Es.3. [10 pt] Given a grayscale image in the file *trees_gray.bmp* with 256 levels of gray implement in MATLAB the following operations:

a) Read, load and visualize the images;
b) Find the gray level threshold that provide almost the same number of black and white pixels and generate a black and white image using that threshold.
c) Remove small sparse spots using a median filter (3x3).
d) Transform again the image from black and white pixels into gray-scale images (just substituting 0 to black pixels and 255 to white ones)
e) Apply an average 3x3 uniform filter and show the result.

Matlab
List of possible functions

```
figure
im2double
im2bw
rgb2gray
imread
medfilt2
imfilter
imshow
zeros
find
imhist
```

Solutions

Ex.1

Ex.2

Ex.3

```
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);
```