Department of Computer Science and Software Engineering
Comp 6771 Image Processing
Assignment 2

Mandana Samiei, ID: 40059116 *14/10/2017*

Theoretical Questions

1- Image binarization and thinning process to repair gaps in OCR outputs:

a) Since the gaps are ranging from 1 to 5 pixels, in order to cover all gaps with different size, we need to use at least 7*7 mask (the size of the mask should be odd). If we use 5*5 mask, in case of locating the filter on the center of 5-pixel gaps, it cannot fill the gaps since it is just including blank pixels. I am going to exemplify my definition by the following example: if we have an image such as the below and we need to fill the zero pixel at the center of 5*5 window (blue one), we won't be able to do this because all the other neighbor pixels are zero, so the result of averaging would be zero for these specific pixels. (5-pixels gaps are highlighted yellow)

b)

0	0	0	1	0	0	0
0	0	1	0	1	0	0
1	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	0	0	0	0	0	1
0	0	0	0	0	0	0

1- An example of an image

c) Thinning is an image processing operation in which binary valued image regions are reduced to lines that approximate the center lines. In the other word, we just have the essential information of the image. After thinning process most of the 1's are connected. Generally, each 1 in the thinned line should always have the minimum number of pixels that maintain eight-connectedness. so if we had just one 1 in a 3*3 block, it means that the eight-connectedness is confirmed so we should change the value of the central pixel into 1. But, in this problem we are facing gaps with the maximum size of 5 and the mask should be 7*7 to be able to fill the gaps. So, in this case we need the 49-connectedness neighborhood. As a result, the threshold should be equal or more than 1/49 to preserver the connectivity. This connectivity guarantees an equal number of thinned connected lines as the number of connected regions in the original image.

Th_min >= 1/49

d) We can use akind of structuring element in order to fill the gaps. For this purpose, we need to design a structuring element according to the size of the gaps, and try to associate it with different parts of the image and in various directions, to find connected regions. Shape and size of the element depend on the size of the image and pattern of 1's and 0's. As the maximum size of the gaps are 5 pixels, we require to have a 7*7 structuring element. Because in structuring elements, we are looking for the correspondence 1's and if we use 5*5 element maybe we don't find any 1 in that window. Additionally, for the shape of the element, we need to consider the shape and structure of characters, since we are designing these elements for correcting character recognition errors. Also, we should consider a thin line of 1's in our elements because after thinning mostly we just have a connected thinned line. I think the following shape of error are more common in OCR systems than square shaped or other kind of geometry shape. But it still depends on their application.

0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
1	1	1	1	1	1	1
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0

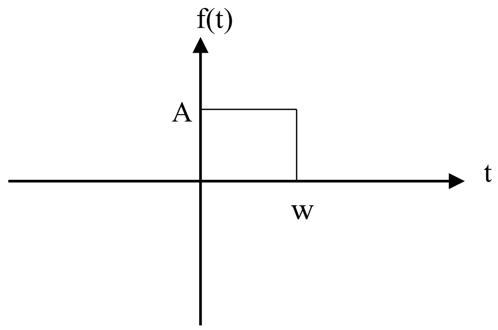
Cross-shaped Element

0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	1	1	1	0	0
1	1	1	1	1	1	1
0	0	1	1	1	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0

Diamond-shaped Element

3-

$$f(t) = \begin{cases} A, & 0 \le t \le w \\ 0, & otherwise \end{cases}$$



a)
In this section, we aim to calculate the Fourier transform of f(t):
At first, we write the Fourier transform formula:

$$\mathcal{F}\ (\mu) = \int_{-\infty}^{+\infty} f(t) e^{-j2\pi\mu t} dt$$

This function has value between 0 to w, so we need to compute the transform just for this interval:

$$\mathcal{F}\ (\mu) = \int_0^w A e^{-j2\pi\mu t} dt = \mathrm{A} \int_0^w e^{-j2\pi\mu t} dt = \frac{A}{-j2\pi\mu} \left(e^{-j2\pi\mu t} \Big|_0^w \right) = \frac{A}{-j2\pi\mu} \left(e^{-j\pi\mu w} - 1 \right) =$$

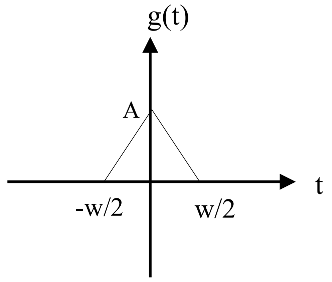
$$\frac{A}{-j2\pi\mu} \left(e^{-j\pi\mu w} - e^{j\pi\mu w} \right) \times e^{-j\pi\mu w} = \frac{A}{\pi\mu} \left(\frac{e^{j\pi\mu w} - e^{-j\pi\mu w}}{2j} \right) \times e^{-j\pi\mu w} = = \frac{A}{\pi\mu} \sin \pi\mu w \times e^{-j\pi\mu w} =$$

$\frac{Aw}{\pi\mu w}$ $(\sin\pi\mu w)\times e^{-j\pi\mu w}$= $Aw\left(sinc(\pi\mu w)\right)\times e^{-j\pi\mu w}$

The only difference between this result and the other in the example 4.1 is $e^{-j\pi\mu w}$.
This is an exponential term and related to the phase of the signal. Also, it can be described according the diagram of f(t). In this example, f(t) is shifted w/2 to the right direction.

If A= w = 1:
$\mathcal{F}(\mu) = sinc(\pi\mu)\times e^{-j\pi\mu}$

b) Fourier Transform of the tent function g(t):



"box" function $f(t) = \begin{cases} A, & -w/2 \le t \le w/2 \\ 0, & otherwise \end{cases}$

g(t) = f(t)∗f(t) = $\int_{-\infty}^{+\infty} f(\tau)f(t-\tau)d\tau$

Fourier Transform of g(t):

$G(\mu) = \int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty} f(\tau)f(t-\tau)d\tau\right]e^{-j2\pi\mu t}dt = \int_{-\infty}^{+\infty} f(\tau)\left[\int_{-\infty}^{+\infty} f(t-\tau)e^{-j2\pi\mu t}\,\mathrm{dt}\right]\mathrm{d}\tau$=

$\int_{-\infty}^{+\infty} f(\tau)\left[F(\mu)e^{-j2\pi\mu\tau}\right]d\tau = F(\mu)\int_{-\infty}^{+\infty} f(\tau)e^{-j2\pi\mu\tau}d\tau = F(\mu)\times F(\mu) =$

$Awsinc(\pi\mu w)\times Awsinc(\pi\mu w)$= $(Aw)^2\times sinc^2(\pi\mu w)$

if A= w= 1:
G(μ) = $sinc^2(\pi\mu)$

Programming Questions

2- In this question we aim to implement the given algorithm to calculate the threshold T to binarize the image.

In this algorithm, entropy thresholding method is used to binarize the image. Entropy is a measure of the information in the image expressed as the average number of bits required to represent the information. Here, the entropy for the two classes is calculated for each potential threshold, and the threshold where the sum of the two entropies is largest is chosen as the best threshold.

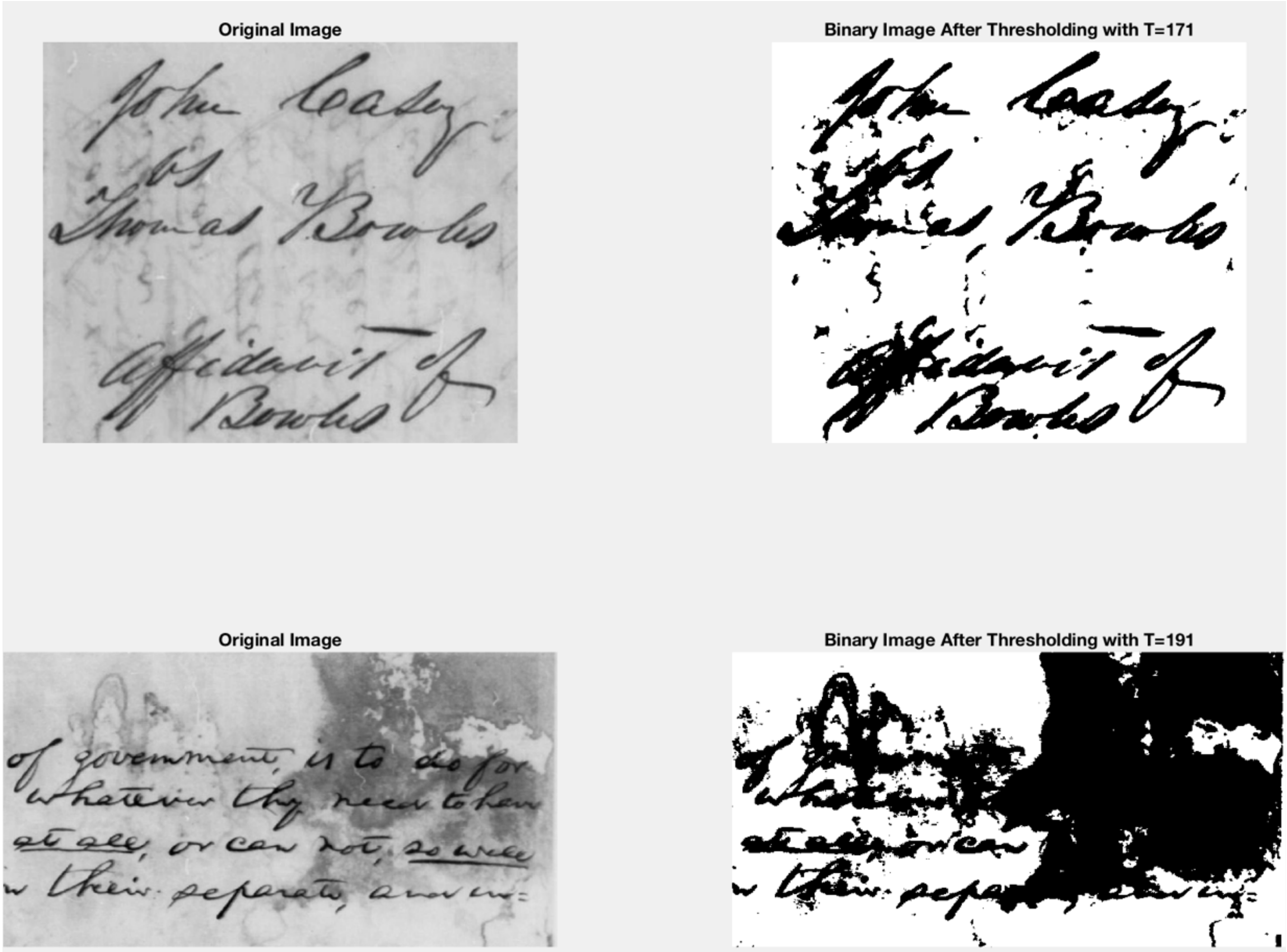
```
function [ T ] = calculateT( img_in )

I = rgb2gray(img_in);
```

```
r = size(I,1);
c = size(I,2);
N = r*c;
img = I(:);

MAX = 0;
T = 0;
L = 256;
p = zeros(256,1); % Probability
cum = zeros(256,1); % Cumulative Probability (Similar to P_t in the Shannon
Enthropy Formula)
hb = zeros(256,1);% Black pixels entropy
hw = zeros(256,1);% White pixels entropy
frequency = zeros(256,1);

% Calculating Probability
for i1= 1:size(img)
    frequency(img(i1)+1)= frequency(img(i1)+1) + 1;
    p(img(i1)+1) = frequency(img(i1)+1)/N;
end

%Calculating CDF(P_t)
for j=1:size(p)
    cum(j) = cum(j) + p(j);
end

%Temp Values
t1_value = 0;
t2_value = 0;
% Total Entropy
h=0;

for t=1:L

    for i2 = 1:t % Calculating the first series of pixels that are assumed
black
        t1_value = -(p(i2)*log(p(i2)/cum(t)))/cum(t);
        if(~isnan(t1_value) && ~isinf(t1_value))
            hb(t)= hb(t) + t1_value;
        end
    end

    for i3 = t+1:L % Calculating the second series of pixels that are
assumed white
        t2_value=-(p(i3)*log(p(i3)/1-cum(t)))/(1-cum(t));
        if(~isnan(t2_value) && ~isinf(t2_value))
            hw(t) = hw(t) + t2_value;
        end
    end

    h = hb(t) + hw(t);

    if h > MAX
        MAX = h;
        T=t;
    end
end
end
```

Testing the function:

```
%%----------Question 2--------------
%%--------- Mandana Samiei----------
%%--------- Student Id: 40059116 ---
%% Test Function

I1 = imread('H03.bmp');
I2 = imread('H04.bmp');
threshhold1 = calculateT(I1);
threshhold2 = calculateT(I2);
normalized_th1 = double(threshhold1/256);
normalized_th2 = double(threshhold2/256);

BW1 = im2bw(I1,normalized_th1);
BW2 = im2bw(I2,normalized_th2);

figure;
subplot(2,2,1);
imshow(I1); title('Original Image');

subplot(2,2,2);
imshow(BW1); title(['Binary Image After Thresholding with T='
num2str(threshhold1)]);

subplot(2,2,3);
imshow(I2); title('Original Image');

subplot(2,2,4);
imshow(BW2); title(['Binary Image After Thresholding with T='
num2str(threshhold2)]);
```

I also plotted the result. Although I know, the result is not satisfying specially for the second image that many backgrounds' shaded parts have darkened. But I put the images and also my program here, I checked it with lot of my friends, some of them had the same result as mine and another guy got the better one with lower thresholds, also I checked the steps of algorithm along details with this guy to understand the problem but the way of our thinking was almost the same, if there is any problem with mine, I would appreciate it if we can discuss it further.


Original Image
Binary Image After Thresholding with T=171
Original Image
Binary Image After Thresholding with T=191