COMP 478 - Image Processing

Assignment 02

Name	ID
Vaansh Vikas Lakhwara	40114764

Part I: Theoretical questions

1. It is given that the thinning processes could result in broken strings of 1s with gaps ranging from 1 to 5 pixels, which means the longest possible gap is 5.

We need to choose a filter with dims. 3x3, 5x5, 7x7, ...

The maximum size of the gap is 5, so the next largest odd numbered filter size is chosen, i.e, 7.

∴ the minimum size of the blurring mask to accomplish the task is 7.

The value of the threshold can be calculated for a weighted filter where only two pixels are considered. So, the minimum threshold value would be 1/7 x 1/7.

∴ the minimum value of the threshold to accomplish the task is $1/49$.

2.
We know that a Laplacian filter represents:

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$$

and that

$$s = \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} = 0$$

where s = sum of all elements in matrix and m = matrix size

It is given that:

$$w_{22} = -16$$

Rules that you should follow to build such a filter:

- extreme intensity changes are noted in the result
- generally sum(all elements) - sum(center) = 0

Therefore,

$$s = \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij:i!=j} = -16$$

Example of 3x3 filter

0	-1	0
-1	4	-1
0	-1	0

Example of 5x5 filter

1	1	1	1	1
1	1	1	1	1
1	1	-24	1	1
1	1	1	1	1
1	1	1	1	1

∴ 5-by-5 Laplacian-like filter with the center element equal to -16:

0	0	1	0	0
0	1	2	1	0
1	2	-16	2	1
0	1	2	1	0
0	0	1	0	0

3. a)

The difference is that my result contains the term $e^{-j\pi\mu W}$, whereas it is not in `Example 4.1`.

By Euler's Equation:

$$e^{ix} = cos(x) + isin(x)$$

$$\begin{aligned} F(\mu) &= \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t}dt = \int_{0}^{W} Ae^{-j2\pi\mu t}dt \\ &= \frac{-A}{j2\pi\mu}\left[e^{-j2\pi\mu t}\right]_0^W = \frac{-A}{j2\pi\mu}\left[e^{-j2\pi\mu W} - e^{0}\right] \\ &= \frac{A}{j2\pi\mu}e^{-j\pi\mu W}\left[e^{j\pi\mu W} - e^{-j\pi\mu W}\right] \\ &= AW\frac{\sin(\pi\mu W)}{(\pi\mu W)}e^{-j\pi\mu W} \end{aligned}$$

When A = W = 1:

$$F(\mu) = AW\frac{\sin(\pi\mu W)}{(\pi\mu W)}e^{-j\pi\mu W} = \frac{\sin(\pi\mu)}{(\pi\mu)}e^{-j\pi\mu}$$

b) Let box function be denoted by $b(x)$ and define it as follows:

$$b(x) = \begin{cases} \mathrm{A}, & if -W/2 \leq \mathrm{x} \leq W/2 \\ 0, & \text{otherwise} \end{cases}$$

Finding the fourier transform for the box function:

$$\begin{aligned} F(b(x)) &= \int_{-\infty}^{\infty} b(x)e^{-j2\pi\mu t}dt = \int_{-1/2}^{1/2} e^{-j2\pi\mu t}dt \\ &= \frac{-1}{j2\pi\mu}\left[e^{-j2\pi\mu t}\right]_{-1/2}^{1/2} = \frac{-1}{j2\pi\mu}\left[e^{-j\pi\mu} - e^{j\pi\mu}\right] \\ &= \frac{\sin(\pi\mu)}{(\pi\mu)} = sinc(\mu) \end{aligned}$$

Let tent function be denoted by $t(x)$. It is a convolution of $b(x)$.
$\therefore F(t(x)) = F(b(x) * b(x)) = F(b(x)) * F(b(x)) = sinc^2(\mu)$

Part II: Programming questions

1. Code:

```
function q1()
    % read
    I = imread("img/Doc.tiff");

    % initialize parameters and
    % call custom function
    [m, c] = deal(9, 5.5);
    O = adaptive_thresholding(I, m, c);

    % initialize parameters and
    % call built-in function
    s = 0.7;
    T = adaptthresh(I, s);
    M = imbinarize(I, T);

    % compare and save
    imshowpair(O, M, "montage")
    imwrite(O, "img/prog/output.tiff");
end

function [O] = adaptive_thresholding(I, m, c)
    % get size
    [row, col, x] = size(I);

    % initialize empty output
    O = zeros(row, col);

    % create a new image with padded boundary
    boundary_len = floor(m / 2);
    K = padarray(I, [boundary_len, boundary_len]);

    % loop through every pixel
    for i= 1 : row
        for j = 1 : col
            % get average of local window
            win = K(i : i + m - 1, j : j + m - 1);
            avg = mean(win(:));
```

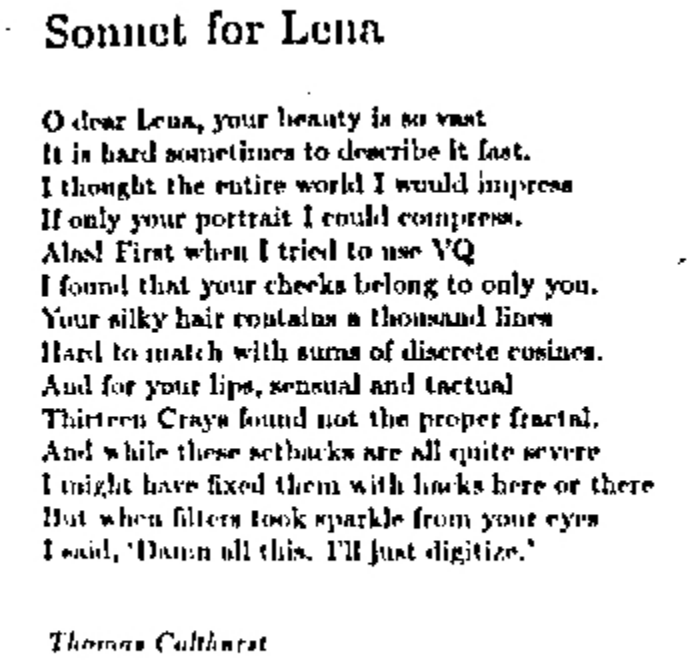
```
            % compute threshold
            thr = avg - c;

            % set output pixel accordingly
            if (I(i, j) >= thr)
                O(i, j) = 1;
            end
        end
    end
end
```

Output:

Window size (m) = 9 x 9

Constant (c) = 5.5



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when fifteen took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Discussion:

- `parameter(s)` : The input arguments for the method are image `I` , window size `m` , and constant `c` as that is all that is required to calculate the threshold transfer function value.
- `filter` : Averaging filter used locally was a simple mean based filter.
- `built-in` : Parameters were tweaked as both functions require different types of arguments. Overall, the results are similar but the text appears darker in the built-in method. This could be due to the set sensitivity level in the built-in `addaptivethresh` function call. Below are the results in tabular form.

My Result	Built-in Matlab
Window size (m) = 9 x 9 Constant (c) = 5.5	sensitivity (s) = 0.7