Digital Image Processing
School of Information Science and Technology
ShanghaiTech University

Assignment 3
Due time: 23:59, May 5th, 2023

---

1 Notes

This homework has 80 points in total.

Please submit your homework to blackboard with a zip file named as DIP2023_ID_Name_hw3.zip. The zip file should contain three things: a folder named 'codes' storing your codes, a folder named 'images' storing the original images, and your report named as report_ID_Name_hw3.pdf. The names of your codes should look like 'p1a.m' (for (a) part of Problem 1), so that we can easily match your answer to the question. Make sure all paths in your codes are relative path and we can get the result directly after running the code. Please answer in English.

Please complete all the coding assignments using MATLAB. All core codes are required to be implemented by yourself (without using relevant built-in functions). Make sure your results in the report are the same with the results of your codes. Please explain with notes at least at the key steps of your code.

2 Policy on plagiarism

This is an individual homework. You can discuss the ideas and algorithms, but you can neither read, modify, and submit the codes of other students, nor allow other students to read, modify, and submit your codes. Do not directly copy ready-made or automatically generated codes, or your score will be seriously affected. We will check plagiarism using automated tools and any violations will result in a zero score for this assignment.

Problem 1 (30 pts)

(a) Please implement the Basic global thresholding on ”flower.tif”. (start with T = 0.001) (10pts)

(b) Please implement the Otsus method on ”caster_stand.tif”. (10 pts)

(c) Please implement the edge tracking using Canny Edge Detector on ”fingerprint.tif”. (10 pts)

Solution:

(a) The binarized image is shown in Figure 1.

(b) The binary image processed by Otsus method is shown in Figure 2.

(c) The edge tracking result using Canny Edge Detector is shown in Figure 3. The gaussian filter has size 30×30 and $\sigma = 5$. The low and high thresholding is 0.2 and 0.48.

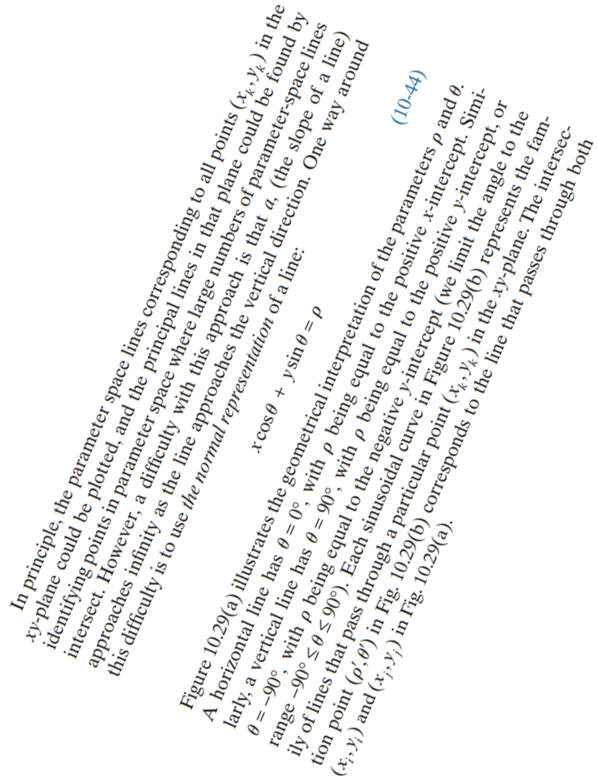
Figure 1: Basic global thresholding on ”flower.tif

Figure 2: Otsus method on ”caster_stand.tif

Figure 3: Edge tracking using Canny Edge Detector

Problem 2 (20 pts)

Figure 1 shows an image of some texts titled with an unknown angle. For better readability, the titling angle must be found and used to correct the orientation of the image. Hough transform is a good choice for this application. 'tilted.png' is one of the texts, your tasks are:


In principle, the parameter space lines corresponding to all points (x_k, y_k) in the xy-plane could be plotted, and the principal lines in that plane could be found by identifying points in parameter space where large numbers of parameter-space lines intersect. However, a difficulty with this approach is that a, (the slope of a line) approaches infinity as the line approaches the vertical direction. One way around this difficulty is to use *the normal representation* of a line:

$$x\cos\theta + y\sin\theta = \rho \qquad (10\text{-}44)$$

Figure 10.29(a) illustrates the geometrical interpretation of the parameters ρ and θ. A horizontal line has $\theta = 0°$, with ρ being equal to the positive x-intercept. Similarly, a vertical line has $\theta = 90°$, with ρ being equal to the positive y-intercept, or $\theta = -90°$, with ρ being equal to the negative y-intercept (we limit the angle to the range $-90° \le \theta \le 90°$). Each sinusoidal curve in Figure 10.29(b) represents the family of lines that pass through a particular point (x_k, y_k) in the xy-plane. The intersection point (ρ', θ') in Fig. 10.29(b) corresponds to the line that passes through both (x_i, y_i) and (x_j, y_j) in Fig. 10.29(a).

Figure 4: tilted text

(a) Implement Hough transform and perform it on 'tilted.png', show its $\rho\theta$ -plane image. The input of Hough transform is a binary image, so you should do some thresholding before performing Hough transform. (15 pts)

(b) Using result obtained by Hough transform, correct the tilt angle, show the corrected image. (5 pts)

Solution:

(a) The $\rho\theta$ -plane image is shown in Figure 5. The binary thresholding is 0.7 (most words need to be set to 1). The $\Delta\rho$ is 2.5 and the $\Delta\theta$ is 0.125.

(b) The corrected image is shown in Figure 6. Since the angle that perpendicular to the text line should contains largest values in accumulator, so I select the θ that has maximial value in accumulator.

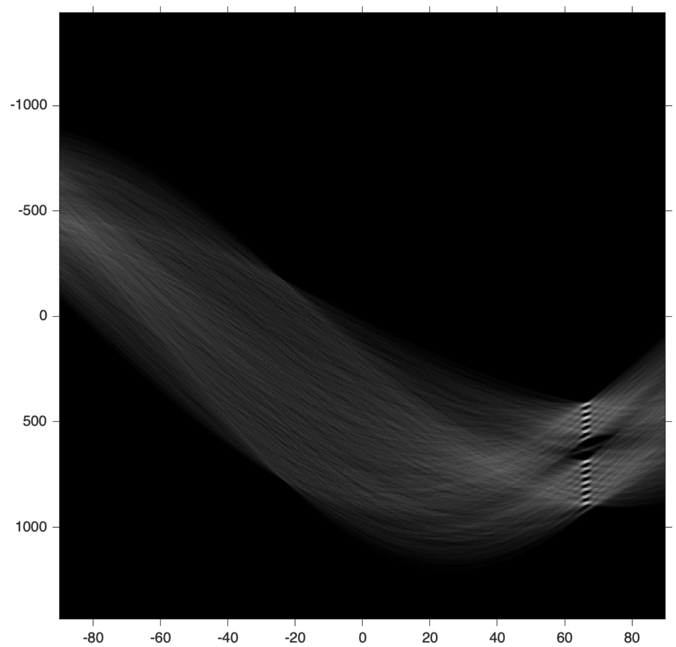
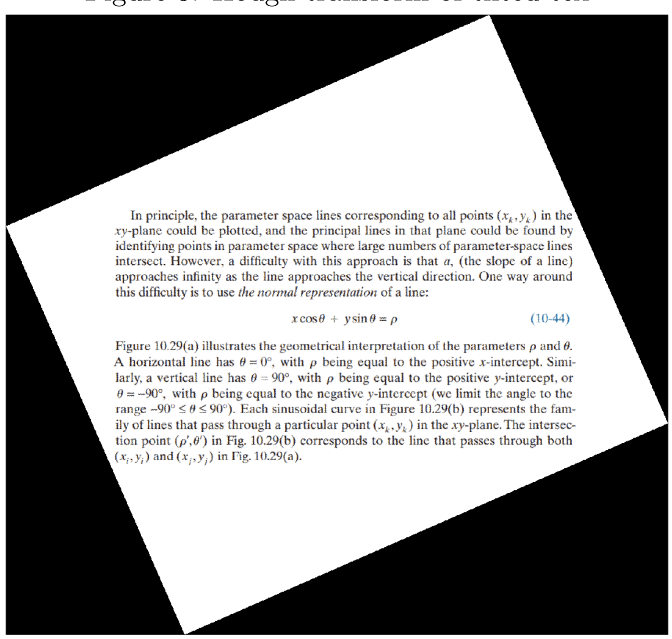


Figure 5: Hough transform of tilted tex



In principle, the parameter space lines corresponding to all points (x_k, y_k) in the xy-plane could be plotted, and the principal lines in that plane could be found by identifying points in parameter space where large numbers of parameter-space lines intersect. However, a difficulty with this approach is that a, (the slope of a line) approaches infinity as the line approaches the vertical direction. One way around this difficulty is to use *the normal representation* of a line:

$$x\cos\theta + y\sin\theta = \rho \tag{10-44}$$

Figure 10.29(a) illustrates the geometrical interpretation of the parameters ρ and θ. A horizontal line has $\theta = 0°$, with ρ being equal to the positive x-intercept. Similarly, a vertical line has $\theta = 90°$, with ρ being equal to the positive y-intercept, or $\theta = -90°$, with ρ being equal to the negative y-intercept (we limit the angle to the range $-90° \leq \theta \leq 90°$). Each sinusoidal curve in Figure 10.29(b) represents the family of lines that pass through a particular point (x_k, y_k) in the xy-plane. The intersection point (ρ', θ') in Fig. 10.29(b) corresponds to the line that passes through both (x_i, y_i) and (x_j, y_j) in Fig. 10.29(a).

Figure 6: Corrected image

Problem 3 (30 pts)

Super pixel is a method that turns a pixel-level picture into district-level picture, which is an abstraction of basic information elements. A super pixel is a small area composed of a series of adjacent pixels with similar character-istics such as color, brightness, and texture. Most of these small areas retain effective information for further image segmentation, and generally do not destroy the boundary information of objects in the image.

In this problem, you need to turn 'sea house.jpg' to super pixel style using SLIC algorithm with 100, 500 and 1000 cluster centers (In practice, number of cluster center can be slightly different from these values) and show the result images.

Reference, doi:10.1109/TPAMI.2012.120

Algorithm 1 SLIC superpixel segmentation

> /* *Initialization* */
> Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S.
> Move cluster centers to the lowest gradient position in a 3×3 neighborhood.
> Set label $l(i) = -1$ for each pixel i.
> Set distance $d(i) = \infty$ for each pixel i.
> **repeat**
>   /* *Assignment* */
>   **for** each cluster center C_k **do**
>   **for** each pixel i in a $2S \times 2S$ region around C_k **do**
>   Compute the distance D between C_k and i.
>   **if** $D < d(i)$ **then**
>   set $d(i) = D$
>   set $l(i) = k$
>   **end if**
>   **end for**
>   **end for**
>   /* *Update* */
>   Compute new cluster centers.
>   Compute residual error E.
> **until** $E \leq$ threshold

Figure 7: SLIC Algorithm

Figure 8: sea house.jpg

Solution: The results with 100, 500 and 1000 cluster centers are shwon in Figure 9, Figure 10 and Figure 11.

Figure 9: cluster=100,threshold=0.5,m=40

Figure 10: cluster=500,threshold=0.5,m=40

Figure 11: cluster=1000,threshold=0.5,m=40