Digital Image Processing, Spring 2018

Homework 3

DUE DATE: May 2, 2018

Student ID: B03902129 Department: CSIE Name: Peng-Yu Chen

README

To run my program, simply type **README** in the Command Window of MATLAB application, then it'll run all .m files and output the .raw images.

Listing 1: README.m

```
% DIP Homework Assignment #3
% May 2, 2018
% Name: Jay Chen
% ID #: B03902129
% email: b03902129@ntu.edu.tw

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Add path first
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('Add path ./prob1, ./prob2, ./bonus and ./readwriter');
addpath('./prob1');
addpath('./prob2');
addpath('./bonus');
addpath('./readwriter');

disp('Make a parent folder ./outputs');
mkdir . outputs

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem 1: MORPHOLOGICAL PROCESSING
% Implementation 1: Boundary extraction, I1 -> B
% Implementation 2: Count the number of objects based on morphological
%                   processing
% Implementation 3: Skeletonizing, I1 -> S
% M-files: prob1.m, boundaryExtract.m, countObjects.m, dilateBinary.m,
%                   label.m and skeletonize.m
% Output: B.raw and S.raw
% Usage: run prob1 to call other .m files
% Parameters:
%       * Boundary Extraction: window size = 3 x 3
%       * Dilation: window size = 9 x 9
%       * Skeletonizing: 8-neighbors
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('-----------------------------------------\n');
fprintf('Running prob1\n-----------------------------------------\n');
prob1();
```

```
%########################################################################
% Problem 2: TEXTURE ANALYSIS
% Implementation 1: Perform Law's method, I2 -> K
% Implementation 2: Generate another texture image, K -> exchanged
% M-files: prob2.m, law.m, computeEnergy.m, kmeans.m, findTextures.m,
%          crossMedianFilter.m and exchange.m
% Usage: run prob2 to call other .m files
% Output: K.raw
% Parameters:
%       * Energy Computation: window size = 13
%       * kmeans: initial centroids: (128, 128), (256, 256) and (384, 384)
%       * Texture finding:
%           * T1(i, j) = 13520
%           * T1(i, j) > 20000 && T2(i, j) > 3500
%       * Cross Median Filter: window size = 61
%########################################################################

fprintf('------------------------------------------\n');
fprintf('Running prob2\n------------------------------------------\n');
prob2();

%########################################################################
% Bonus
% Implementation 1: Produce an image by appropriate morphological
%                   processing
% M-files: bonus.m and dilate.m
% Usage: run bonus to call other .m files
% Output: None
% Parameters:
%       * Cross Median Filter: window size = 25
%       * Dilation: window size = 11
%########################################################################

fprintf('------------------------------------------\n');
fprintf('Running bonus\n------------------------------------------\n');
bonus();
```

PROBLEM 1: Morphological Processing

Given a binary image I_1 as shown in Fig. 1. White pixels represent the objects and black pixels represent the background. Please follow the instructions below to create several new images and describe the method in detail for each case.

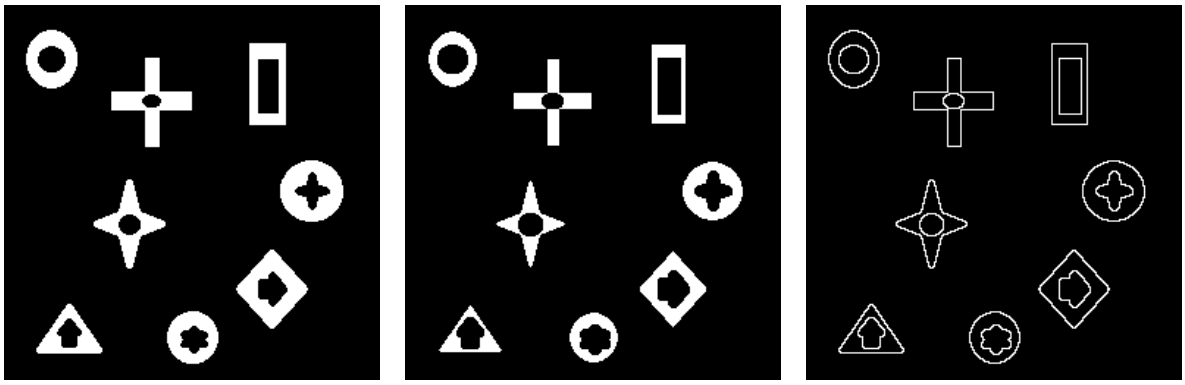
(a) Perform boundary extraction on I_1 to extract the objects' boundaries and output the result as image B. Please provide some discussions about image B.

I perform boundary extraction by the following formula:

$$B(I_1(j,k)) = I_1(j,k) - (I_1(j,k) \ominus H(j,k)).$$

(i) Calculate $(I_1(j,k) \ominus H(j,k))$.
Initialize a zeros array I_1H to denote $(I_1(j,k) \ominus H(j,k))$ first.
For each point, we create a 3×3 window, if there isn't any 0 in the window, we turn the $I_1H(j,k)$ to be 255 (white); otherwise, it remains 0 (black).

(ii) Calculate $B(I_1(j,k)) = I_1(j,k) - (I_1(j,k) \ominus H(j,k))$ directly.



(a) I1.raw (b) $I_1 \ominus H$ (c) B.raw

Figure 1: Boundary Extraction

(b) Please design an algorithm to count the number of objects in I_1 based on morphological processing.

(i) To find the number of objects, I notice that there are some spaces in the objects, so I perform *dilation* with a 9×9 structure element with 1s first.

$$Dilated(j,k) = I_1(j,k) \oplus H(j,k)$$

(ii) After perform dilation, we initialize an array *Labeled* with zeros and a variable $labelNum = 1$.

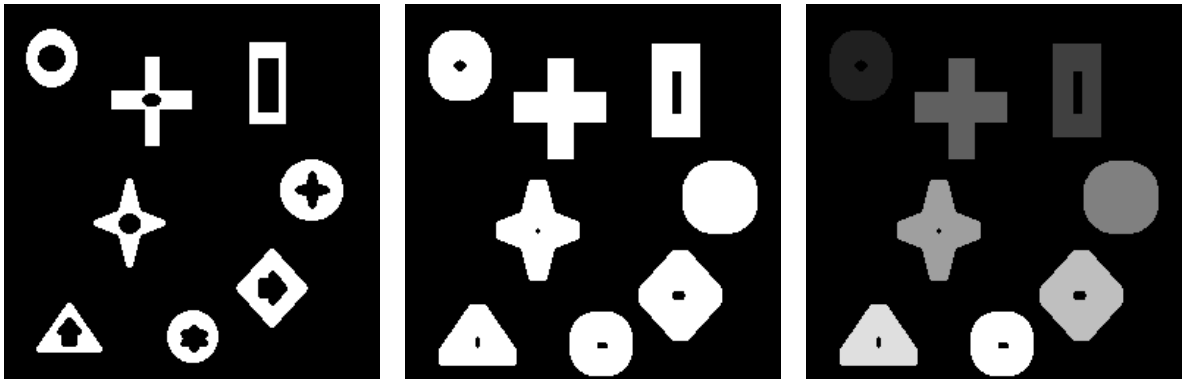
(iii) For each entry, we labeled the *Labeled* array with 4-connectivity by depth first search.

```
if Dilated(i, j) == 1 && Labeled(i, j) == 0
    Labeled(i, j) = labelNum;
    Labeled = label(Dilated, Labeled, labelNum, i, j, h, w);
    labelNum = labelNum + 1;
end
```

There are four cases:

i. $Dilated(i+1, j) == 1$ and $L(i+1, j) == 0$
ii. $Dilated(i-1, j) == 1$ and $L(i-1, j) == 0$
iii. $Dilated(i, j+1) == 1$ and $L(i, j+1) == 0$
iv. $Dilated(i, j-1) == 1$ and $L(i, j-1) == 0$

Each case will recursively call the *label* function with the same orientation.
The program will finally output the $max(Labeled) = 8$.



(a) I1.raw (b) *Dilated* (c) *Labeled* by different intensities

Figure 2: Count Objects and Label

(c) Perform skeletonizing on I_1 and output the result as image S. Please provide some discussions about image S.

Here I implement the Zhang Suen Thinning Algorithm found in the web.

(i) Convert the I_1 to $logical(I_1)$ array with only 0s and 1s.

(ii) Initialize variables $isChange = true$ and an array *deletion* with 1s.

(iii) While $isChange == true$, we first set the $isChange = false$, then we determine the *deletion* array in two steps for each while-loop.

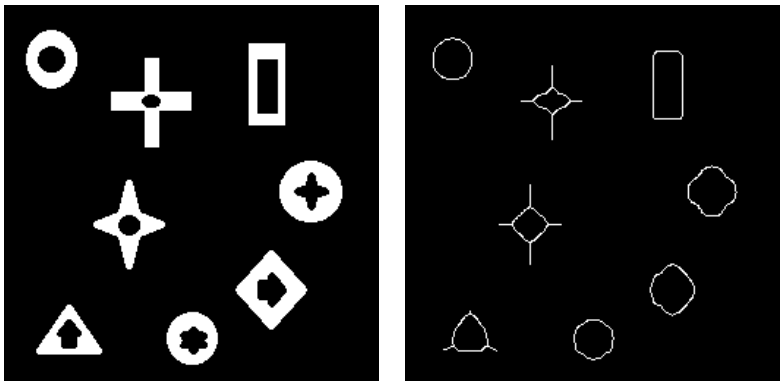
(iv) For each entry, we let P equals to the element of $I_1(j, k)$ + its 8 neighbors (clockwise from $I_1(j-1, k)$) and we calculate the boolean variable *cond* based on which step we take:

```
if step == 1
    cond = (S(i, j) == 1 && sum(P(2: end − 1)) <= 6 && sum(P(2: end − 1)) >=2
        && P(2) * P(4) * P(6) == 0 && P(4) * P(6) * P(8) == 0);
elseif step == 2
    cond = (S(i, j) == 1 && sum(P(2: end − 1)) <= 6 && sum(P(2: end − 1)) >=2
        && P(2) * P(4) * P(8) == 0 && P(2) * P(6) * P(8) == 0);
end
```

(v) We then determine the *deletion* array by

```
if cond
    A = 0;
    for k = 2: size(P, 2) − 1
```

```
            if P(k) == 0 && P(k + 1) == 1
                A = A + 1;
            end
        end
        if A == 1
            deletion(i, j) = 0;
            isChange = true;
        end
    end
```



(a) I1.raw (b) S.raw

Figure 3: Count Objects and Label

The algorithm is to thin the input image I_1 until there is no change. I calculate the skeletonized image with 8-neighbors. The skeletonized image is a thinnest image without changing the topological order but it'll sacrifice some details of the original image.

PROBLEM 2: Texture Analysis

An image I_2 which is composed of several different textures is given in Fig. 2.

(a) Perform Law's method on I_2 to segment the image into 3 different texture groups. Label the pixels of the same texture group with the same intensity values. Please detail the method you choose, specify all the parameters and output the result as K.

After performing Law's method on I_2, we'll obtain

$$M_i = I_2 \otimes H_i, \text{where } H_i \text{ is laws' masks and } i = 1, 2, \ldots, 9.$$

We then compute the entropy for each mask,

$$T_i(j,k) = \sum\sum |M_i(j-m, k+m)|, \text{where } i = 1, 2, \ldots, 9 \text{ and } m = 6(window.size = 13).$$

Then run k-means algorithm with 9 features: $M_1, M_2, \ldots, M_9$ to classify 512^2 entries by 1, 2 or 3.

The algorithm of k-means is as follows:

(i) Initialize centroids of k-clusters randomly.

(ii) Assign each sample to the nearest centroid.

(iii) Calculate centroids (means) of k-clusters.

(iv) If centroids are unchanged, done. Othrewise, go to step (ii).

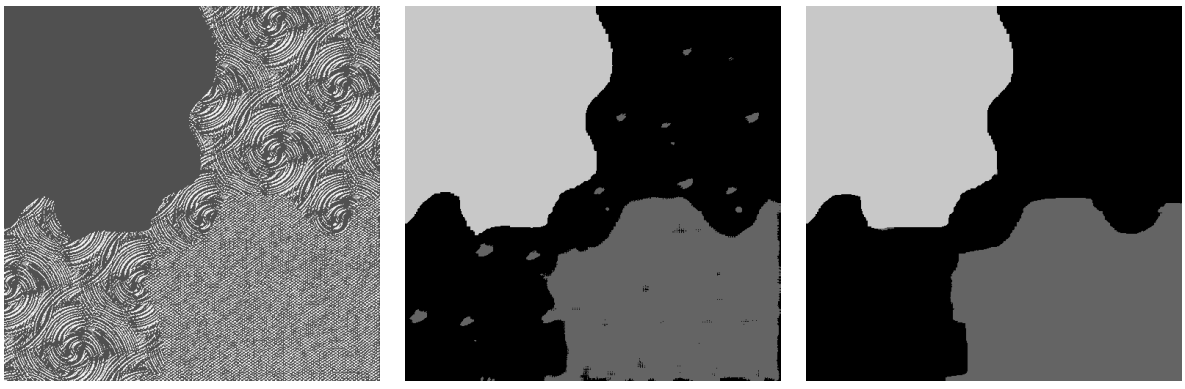
Then I use thresholding method with

(i) $KMEANS$ (computed by k-means algorithm) and

(ii) T_8 (a good feature picked by eye)

to label the 3 different textures with intensities 0, 100, 200.

Finally, using Cross Median Filter with $cross.size = 61$ to filter out the noises and obtain K.raw.

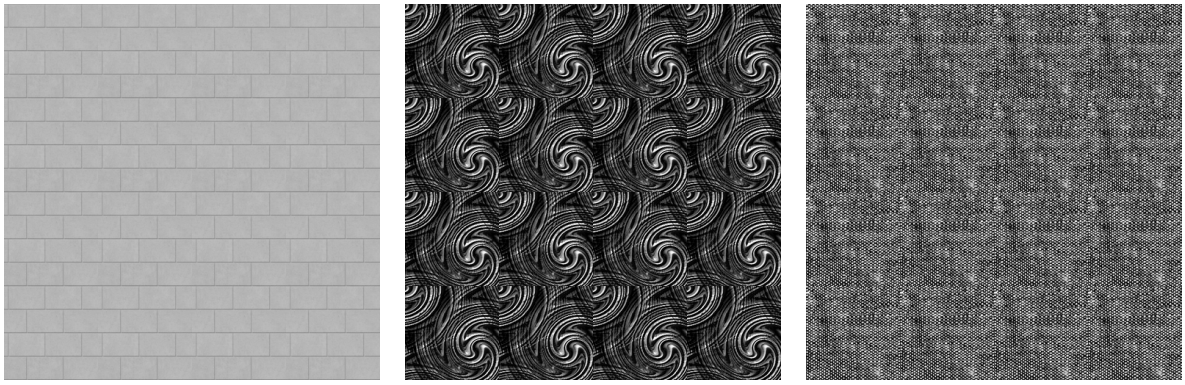


(a) $KMEANS$ (b) $noiseK$ (c) K.raw

Figure 4: Texture Analysis

(b) Based on K, try to generate another texture image by exchange the types of different texture patterns.

To exchange texture patterns, we first extract a 128×128 area for each texture patterns. Then we repeat each pattern for 16 times to get a full size 512×512 texture map.



(a) *map*1 (b) *map*2 (c) *map*3

Figure 5: 3 different texture maps

For each entry in the K, there are three cases:

(i) intensity = 0, change it to the corresponding entries of *map*3.

(ii) intensity = 100, change it to the corresponding entries of *map*1.

(iii) intensity = 200, change it to the corresponding entries of *map*2.

(a) K with labeld intensities (b) I_2 (c) *exchanged*

Figure 6: Texture Exchanging

BONUS

Given an image I_3 shown in Fig. 3, please try to produce an image as illustrated in Fig. 4 by adopting appropriate morphological processing. Please describe the designed algorithm in detail and provide some discussions.

I first apply Cross Median Filter with $window.size = 25$, thne use dilation method with $window.size = 11$ to enhance the given image.

(a) I3.raw (b) $crossMedianFilter(I_3, 25)$ (c) Blurred image

Figure 7: Blurred Image