CSE 152 Assignment 2
Spring 2006
Due Tuesday, May 09, in class

Instructions :

- Attempt all questions. Questions marked bonus will be graded more subjectively.
- Please comment all your Matlab code adequately.
- Try to minimize use of loops in Matlab!

1 Morphological Image Processing

We will limit our interest to binary images in this section. Recall that binary images have only two intensity levels, one corresponding to the foreground and the other background.

(a) Consider the image in *figure1.png*. Based on the histogram of this image, write Matlab code to automatically determine an intensity level at which you would threshold the image to segment out the background from the foreground. Create a binary image B of the same size as the input image, but where each background pixel has value 0, while the foreground is assigned an intensity of 1. Show the image B. **[4 points]**

*For all further questions 1***(b)** *to 1***(f)***, the input image must be converted to binary as in question 1***(a)***.*

Morphology refers to application of set operations such as union and intersection on an image. Morphological algorithms take as input an image and a structuring element, which encodes the shape characteristics based on which the input image must be processed. Usually, the structuring element is a $(2k+1) \times (2k+1)$ array, where k is a small number (say, between 1 and 5).

In general, a morphological operation is performed by making a copy of the input image. The structuring element, S, is compared to a $(2k+1) \times (2k+1)$ window neighborhood, W, of each pixel of the input image. If W exactly matches S, then the corresponding pixel of the copy image is set to a particular value (0 or 1, depending on particular operation), else that pixel in the copy is left unchanged.

(b) **Erosion:** The desired effect in erosion is that any foreground pixel which has a background pixel as a neighbor is set to background. Consider the 3×3 structuring element :

$$S_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The 3×3 neighborhood of each pixel in the input image is compared to this structuring element S_1 and the corresponding pixel in the copy image is set to background if they do not

match, else it is left unchanged. Thus, it causes the boundary of the foreground to shrink. Write Matlab code to perform erosion using the element S_1 on image *hole.png*. (Remember to convert the input to a binary image first.) Display your output after the image has been eroded by 1, 4 and 10 passes of the erosion algorithm. **[4 points]**

(c) Now repeat question **(b)** with a structuring element S_1' of the same form as S_1, but of size 7×7. Also, give a qualitative description of the effect of increase in size of the structuring element. **[3 points]**

(d) Dilation: The dual problem to erosion is called dilation. Here the aim is to use a morphological operation to expand the boundary of the foreground region of an image. One way to do this is the following:

```
for each background pixel p
    Let N = neighborhood of p
    if N contains only background pixels
        leave p unchanged
    else
        set p to foreground
    end
end
```

Describe the process and structuring element, S_2, you would use to perform dilation on an input image. Write Matlab code that takes as input the figure *hole.png*, the structuring element you have devised and desired number of passes. The output should be the dilated image after the specified number of dilation passes. Display the dilated image after 1, 4 and 10 passes. **[5 points]**

(e) Skeletonization: We will now compute the skeleton of the image *fingerprint.jpg* using morphological operations. The idea here is to remove any pixel on the boundary of the foreground which has more than one foreground neighbor, provided its removal does not split the connected component of which it is a member into two.

Consider the structuring elements:

$$S_3 = \begin{bmatrix} 0 & 0 & 0 \\ * & 1 & * \\ 1 & 1 & 1 \end{bmatrix} \qquad S_4 = \begin{bmatrix} * & 0 & 0 \\ 1 & 1 & 0 \\ * & 1 & * \end{bmatrix}$$

Here, 1 stands for foreground, 0 stands for background and $*$ stands for "don't care". That is, $*$ can be either foreground or background. Again, the structuring element will be compared to the 3×3 neighborhood of each pixel and if they match, then the corresponding pixel in the output image is set to background.

Let $I_1 \xrightarrow{S} I_2$ denote a single pass over image I_1 using the structuring element S to result in the image I_2. Let $S(\theta)$ denote the structuring element S rotated by angle θ. Then, a single pass of the skeletonization algorithm can be expressed as

$$I_1 \xrightarrow{S3(0^\circ)} I_2 \xrightarrow{S4(0^\circ)} I_3 \xrightarrow{S3(90^\circ)} I_4 \xrightarrow{S4(90^\circ)} I_5 \xrightarrow{S3(180^\circ)} I_6 \xrightarrow{S4(180^\circ)} I_7 \xrightarrow{S3(270^\circ)} I_8 \xrightarrow{S4(270^\circ)} I_9$$

To compute the skeleton of the input image, repeat the above sequence of operations until convergence (that is, when further operation on the image by a structuring element leaves it unchanged). Write Matlab code for the same and display the computed skeleton. **[5 points]**

(f) Bonus question: Can you use one of the morphological operations discussed in parts **(b)-(e)** to detect edges in the image *hole.png*? Write Matlab code and display the edge map.

Why do you think filtering mechanisms are preferred over morphological ones in edge detection in real world imaging situations? **[4 points]**

2 Counting Objects

(a) Labeling: Write Matlab code to implement the connected component labeling algorithm discussed in class, assuming 8-connectedness. Use your code to count the number of components in the images *components1.jpg* and *components2.jpg*. Display a grayscale image as output, where you assign a distinct gray-level to each connected component. **[5 points]**

(b) How many components do you think are in the image *coins.png*? What does your connected component algorithm give as output for this image?

Can you use one of the morphological operations in Question 1 to "separate out" the individual coins in this image? Write Matlab code for this separation and display any relevant figures for output. **[5 points]**

(c) Use your connected components labeling algorithm to count the number of coins in the output image of 2**(b)**. **[3 points]**

(d) Bonus question: Use some criterion to automatically distinguish between the kinds of coins in the image *coins.png*. You can assume that there are just two kinds of coins here - dimes and quarters. Use your program to count the amount of change in this image. The number of points you score is equal to the correct amount of change. **[4 points]**