DIP HW3 Report

R07922036 邱能賢

April 10, 2019

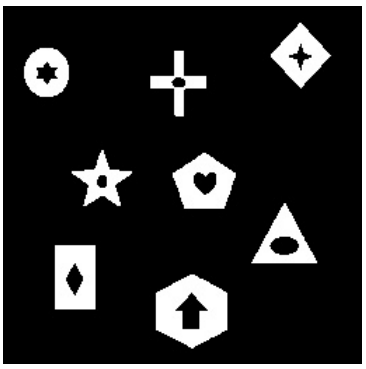
Problem 1: Morphological Processing

(a) Perform boundary extraction on I_1 to extract the objects' boundaries and output the result as an image B.

I tried to extract the boundaries with the following two methods, where H is the 3-by-3 structuring element:

$$\beta_{inner}(I_1) = I_1 - (I_1 \ominus H)$$

$$\beta_{outer}(I_1) = (I_1 \oplus H) - I_1$$



I_1, sample1.raw

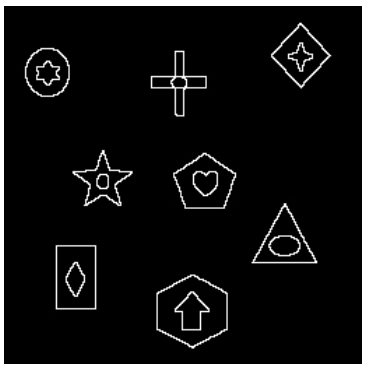


image B - Inner boundary

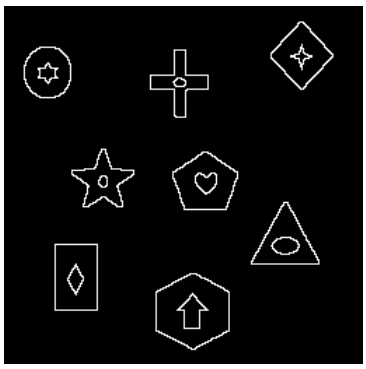


image B - Outer boundary

(b) Perform connected component labeling on I_1 to obtain an image C where different objects are labeled with different colors.

To perform connected component labeling, I iterate the following operation until there is no change in G, starting with G_1 containing only one first-found white pixel:

$$G_{i+1} = (G_i \oplus H) \cap I_1$$

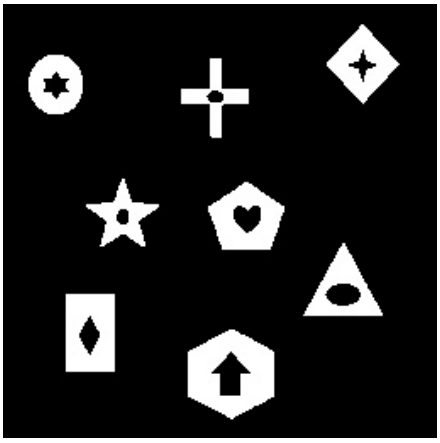
Mark G on image C, then erase G from I_1, and continue with the next connected component.

(c) Perform thinning and skeletonizing on I_1 and output the results as image $D1$ and D_2.

Here I used the two-stage algorithm developed by Pratt and Kabir. The operation can be expressed as

$$D = I_1 \cap [\bar{M} \cup P]$$

where M and P are the images generated by the two 3×3 hit-or-miss transformation stages.



I_1, sample1.raw

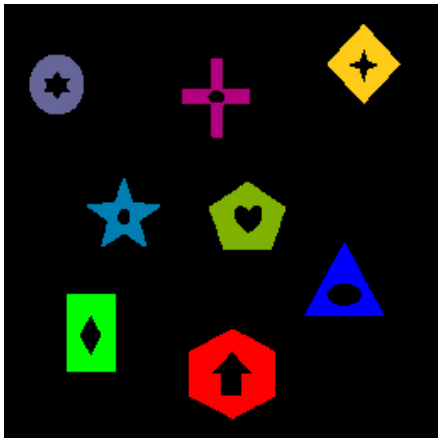
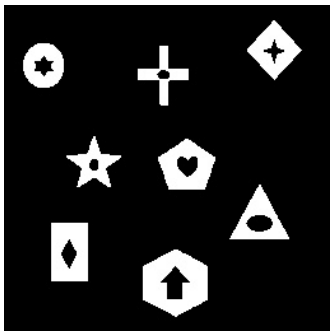


image C



I_1, sample1.raw

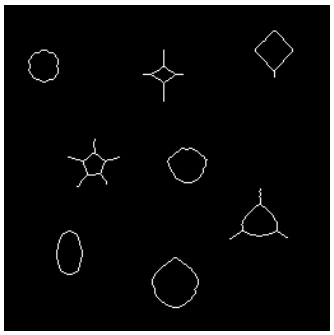


image D_1

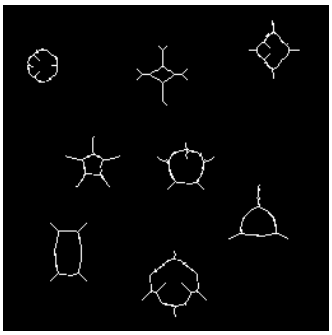
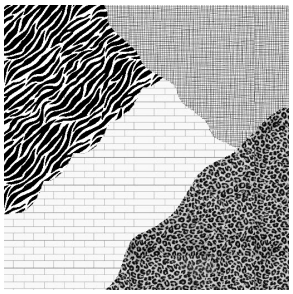


image D_2

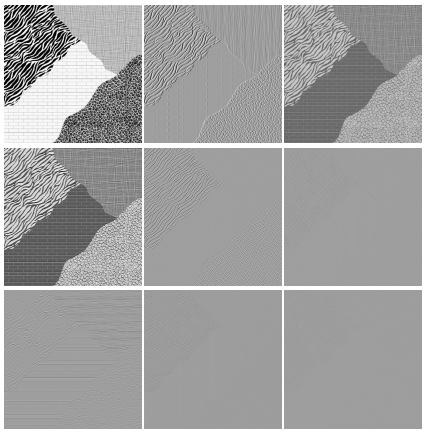
Problem 2: Texture Analysis

(a) Perform Law's method on I_2 to obtain the feature vector of each pixel.

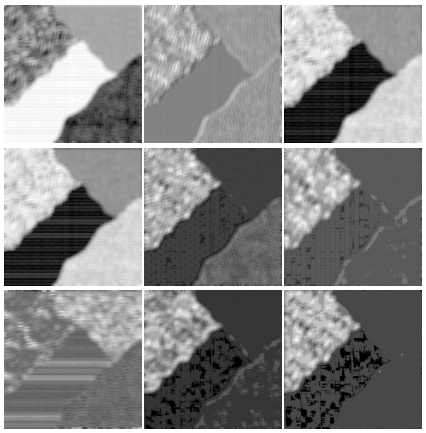
At the second stage, the feature is the energy in a 19×19 window.



I_2, sample2.raw



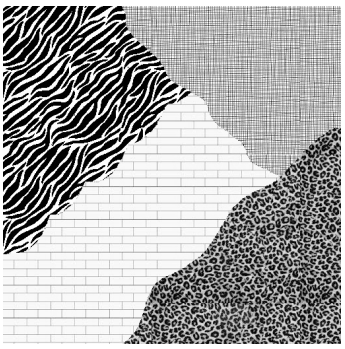
9 Micro-structure response



Feature vectors (Energy)

(b) Use k-means to classify each pixel and label same kind of texture with same gray-level intensity and output the result as image E.

The following is the resulting image E using k-means to classify each pixel. The centroid gets stable after 8 iterations. As the image shows, there are lots of holes in the top left texture, also the classification is poor at texture boundaries. When the energy window increases, the holes become less but the mis-classified boundary gets wider.



I_2, sample2.raw

Image E

So, I use the following method to improve the texture segmentation.

(a) First extract the 4 texture to its own binarized image.

Texture 1. Texture 2. Texture 3. Texture 4.

(b) Perform the close operator followed by the open operator on each texture.

Texture 1. Texture 2. Texture 3. Texture 4.

(c) Perform connected component labeling and remove components whose area is less than 1000 pixels.

Texture 1. Texture 2. Texture 3. Texture 4.

Here is the merged image:

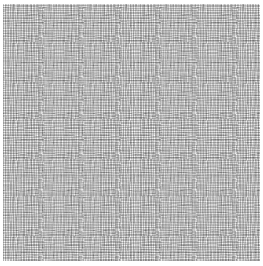
Original image E. Optimized image E.

There are still gaps at texture boundaries, so I assign unclassified pixels to its nearest texture to fill the gap.

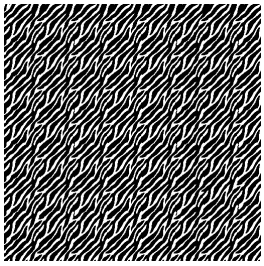
Optimized image E. Image E after gap filling.

(c) Based on E, try to generate another texture image by exchanging the types of different texture patterns as image G.

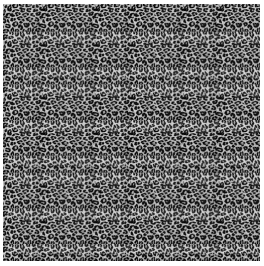
I tried implementing the image quilting [1] algorithm. I cropped an 84×84 block from I_2 and generated the 4 texture image.



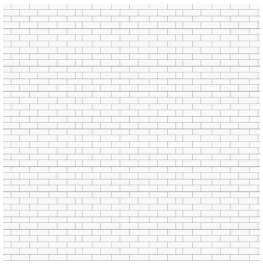
Texture 1.



Texture 2.

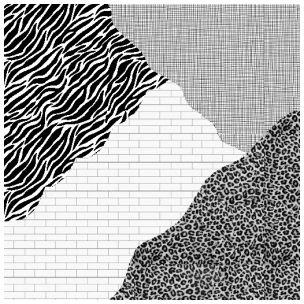


Texture 3.



Texture 4.

Then use the segmentation from image E to merge the textures in different orders.



I_2, sample2.raw

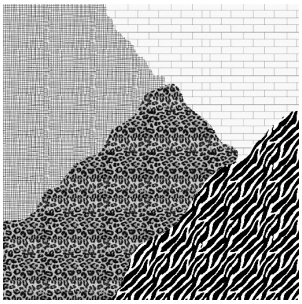


Image G

References

[1] "Image Quilting." Efros and Freeman. SIGGRAPH 2001