Computer Vision CS534

Homework 2:

Texture Synthesis and Image Inpainting

Hao Luo

RUID: 162006265

Main Task

1.1. Texture Synthesis:

Efros and Leung's approach is implemented in Matlab language.

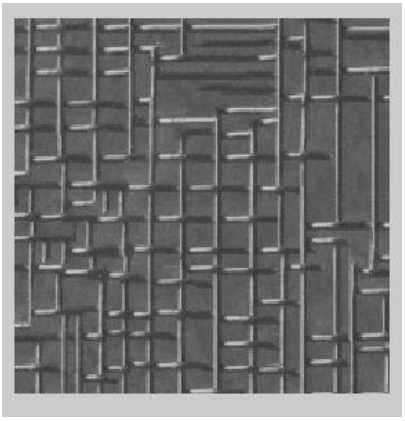
I think the WindowSize is 2*parameter+1.

So I use WindowSize (11*11, 19*19, 23*23) to compute the final result.

We can see that as n goes larger, the outcome becomes smoother.

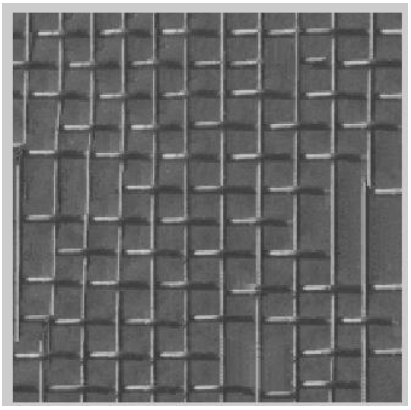
T1.gif

Windowsize=11



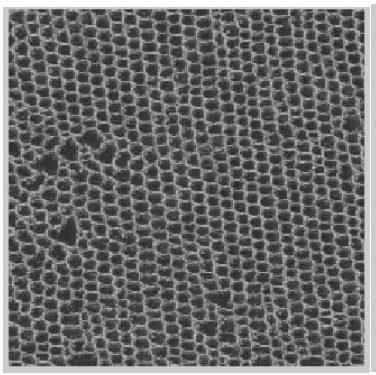
Windowsize=19

Windowsize=23

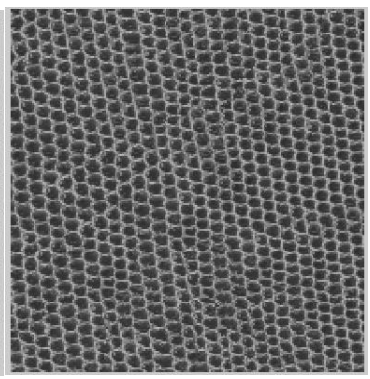


T2.gif

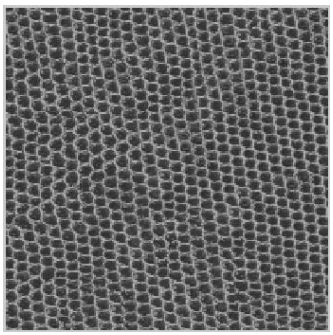
Windowsize=11



Windowsize=19

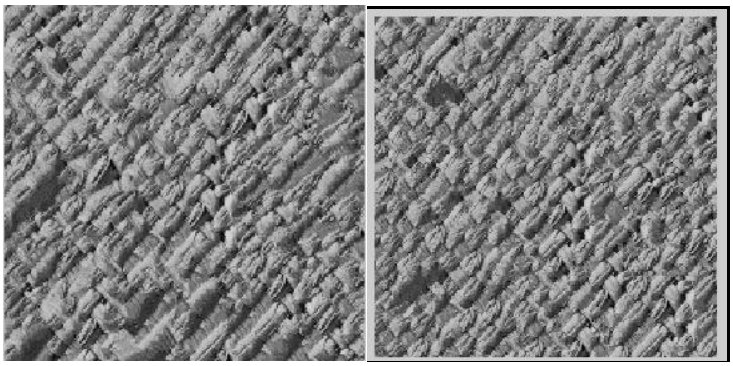


Windowsize=23

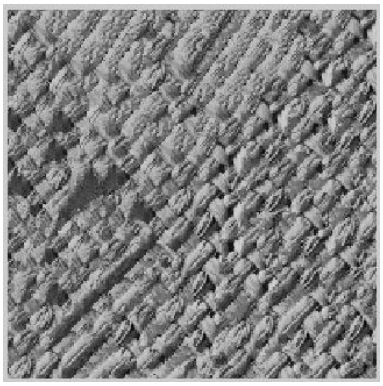


T3.gif

Windowsize=11 Windowsize=19

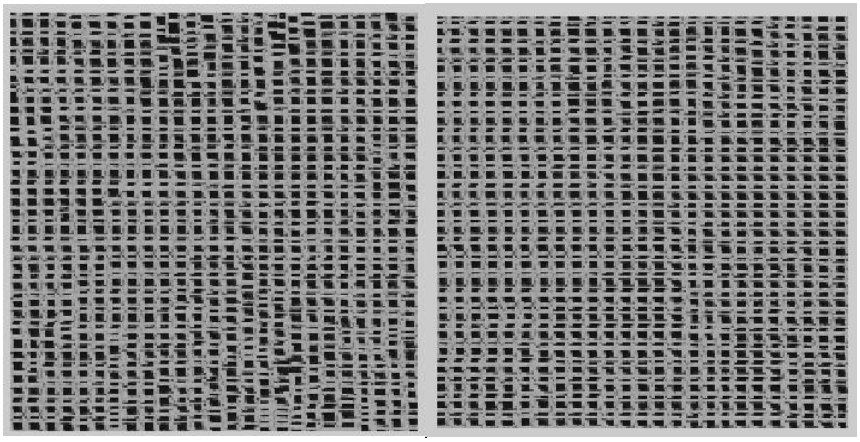


Windowsize=23

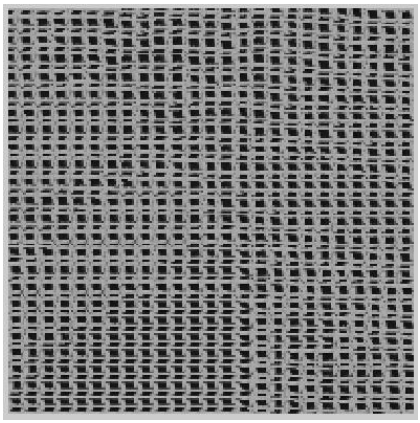


T4.gif

Windowsize=11 Windowsize=19

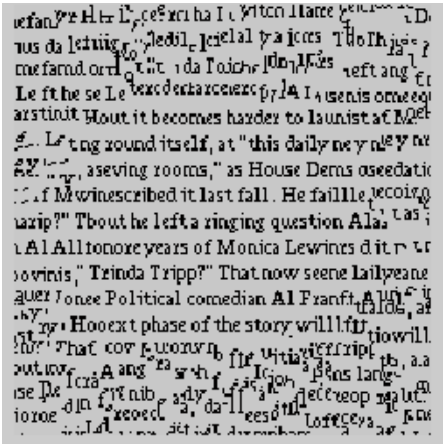


Windowsize=23

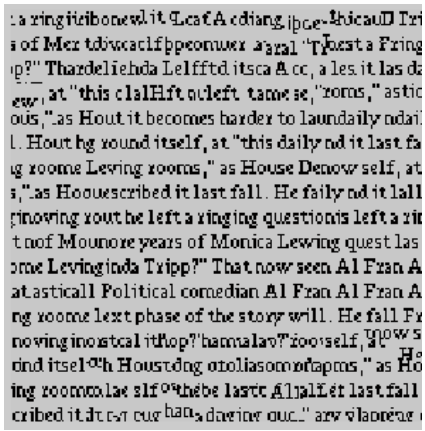


T5.gif

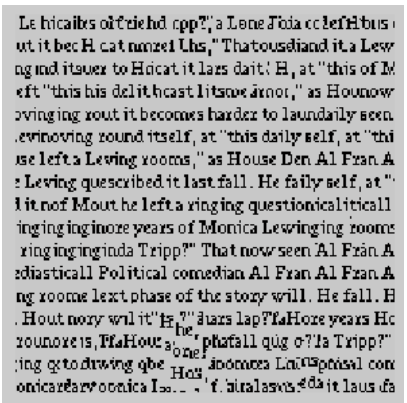
Windowsize=11



Windowsize=19



Windowsize=23



1.2 Image Inpainting:

In this part, I set the same window size as before.

test_im1.bmp

Windowsize=11

Windowsize=19

Windowsize=23

test_im2.bmp

Windowsize=11

Windowsize=19

Windowsize=23

We can see that as the windowsize goes larger, the outcome becomes more clear and more smooth.
At first, I used the same function as I used in Texture Synthesis. But the result is not good. Some part of the filling area sometime looks different from surrounding. Then I realized that I have to take the sample area from neighborhood. And the outcome looks 'normal'.

1.3 Object Removal

Efros's method of remove man

Efros's method of remove sign

Efros's method of remove tile

Criminisi's method of remove man

Criminisi's method of remove sign

Criminisi's method of remove tile

I set the windowsize of these two methods is 11.
Although the result of the Criminisi's method is color picture and the Efros's result is the grayscale image, we can still get the conclusion of these two methods roughly. Among the 3 removements, the results of sign are the best and the results of tile have some flaws.
I think one reason is that the area of sign which needs to be filled is close to background of the picture. And the small area also makes the result looks better.
And the large filling region of tile makes some small exceptions changing into flaws possible.
Although the Efros's result looks a little better than the Criminisi's one, Criminisi's method take less time to complete the work. (In fact, the time of Efros's method is more than twice of the Criminisi's one). And considering that the Color pictures has more complexities than the grayscale image, the Criminisi's method is more effective.

1.4 Image Quilting

1.4.1 Summary of Image Quilting

Unlike Efros's method, Image Quilting deals with patch instead of pixel.
Image Quilting combines patches of current pictures to generate the new image. The patches are randomly chosen from the sample image. And the best n input texture patches will be chosen according to the value of SSD. The SSD value depends on the overlap area of every promising texture patches. Also threshold is set to avoid bad matching results, error is the difference between the existing patches and the new one. And by determining the minimum pixel difference, quilt the best texture patch from the existing part outwardly.

1.4.2 Similarities and Differences

They all use the sample image from neighborhood by computing the SSD value.
The main difference is that Image Quilting using patches, while Efros's method using Pixels.
And Image Quilting is much faster than the Efros's one, usually the work which needs a few minutes of Image Quilting Method will take hours for Efros's method to finish.
What's more, Image Quilting usually achieves better result than the Efros's method.
With boundary removing approach, Image Quilting is more visible and produce less flaws.
Also Image Quilting can be used in Texture Transfer while Efros's is only designed for the Texture Transfer.

Appendix: How to run the code :

Make the codes and pictures in the same directory.

In Texture Synthesis:
Use the function Texture_Synthesis.m (this function will use the ExtensionImage.m and FindMatch.m)
Input the path of the image and the window size:

```
result_img = Texture_Synthesis('T1.gif',11);
```

In Image Inpainting:
Use the function Image_Inpainting.m (this function will use the ExtensionImage2.m and FindMatch.m)
Input the path of the image and the window size:

```
result_img=Image_Inpainting('test_im1.bmp',11);
```

In Object removal:
Use the function Object_Remove.m (this function will use the FindMatch.m)
Just run this function and follow the instruction to set the area needs to be removed.