**HomeWork2 Balaji Suresh**

Running the code

For texture synthesis run “ImageSynthesis.m” , for image inpainting run “ImageInpainting.m” and for object removal run “ImageFilling.m”.

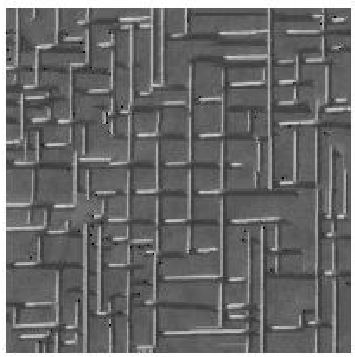
Assign the values for parameters **Window Size, Sample Image, Neighbourhood Size** for running each of these functions

**Image Synthesis**

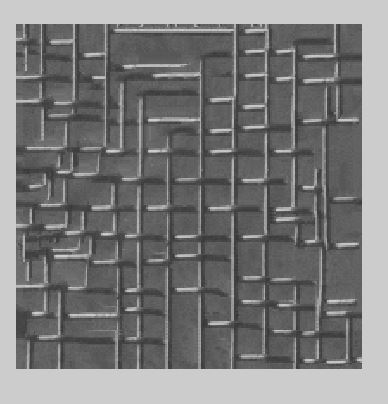
Below are the results that are obtained by modifying each parameter

**T1**

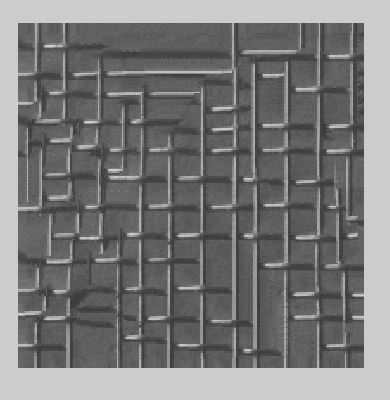
Window size =5



Window size=9

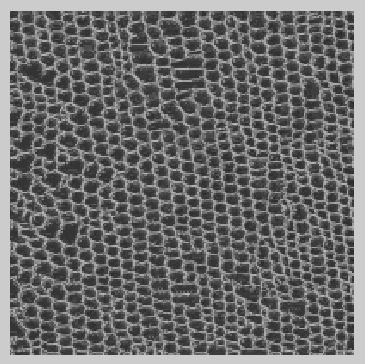


Window size=11

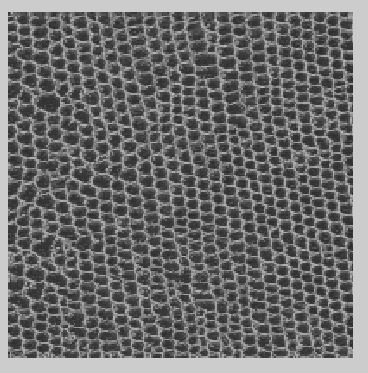


**T2**

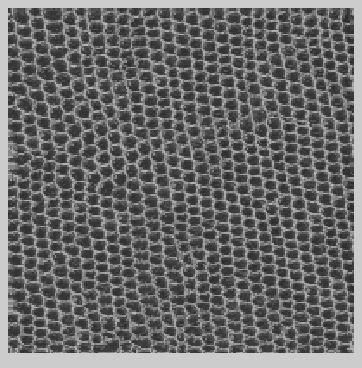
Window size=5



Window size=9

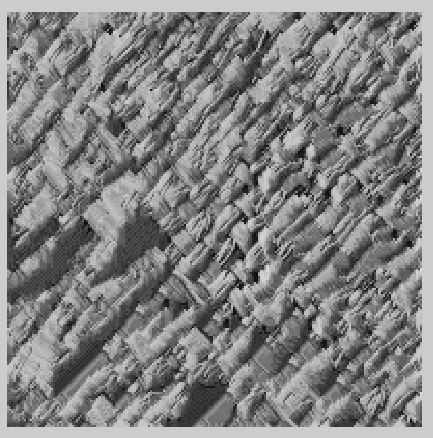


Window size=11

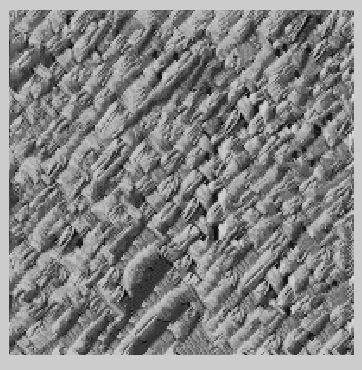


**T3**

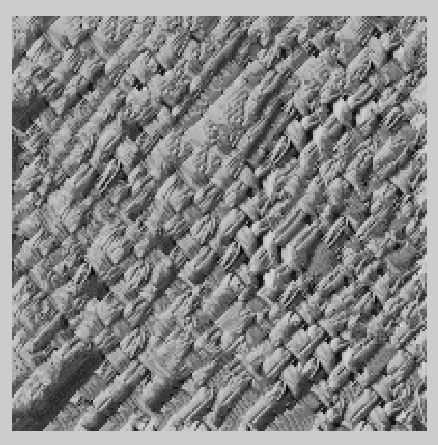
Window size =5



Window size=9

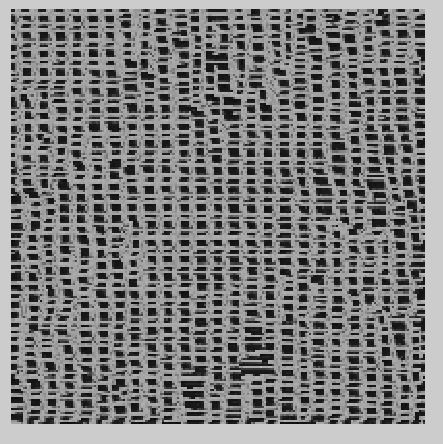


Window size =11

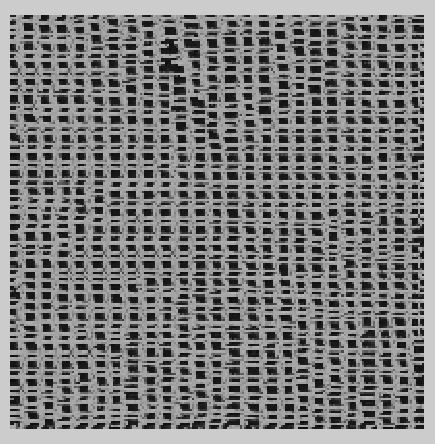


**T4**

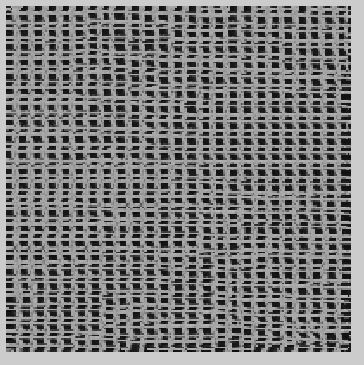
Window size=5



Window size=9

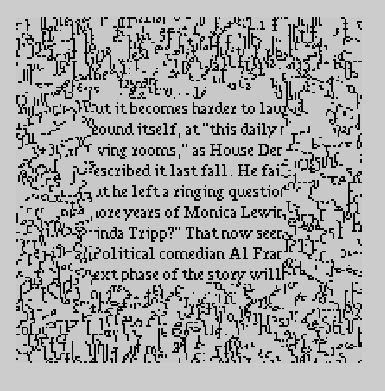


Window size=11

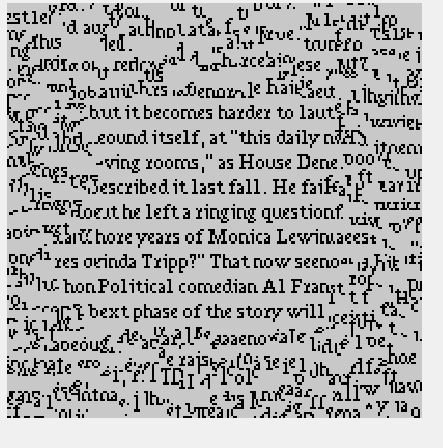


**T5**

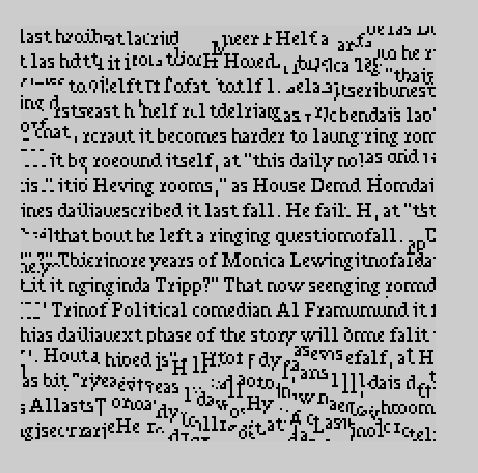
Window size =5



Window size=9



Window size=11



From the above images we can observe that the image quality keeps increasing as the window size increases. The reason for this is that Is as the size increases the pixel has extra values to acess which also increases the computation time.

Image Inpainting

Test Image 1

Window size 11



Window size 9



Window size 5



Test image 2

Window Size 5



Window size 9



Window size 11



**Obejct Removal**

The sign



The man

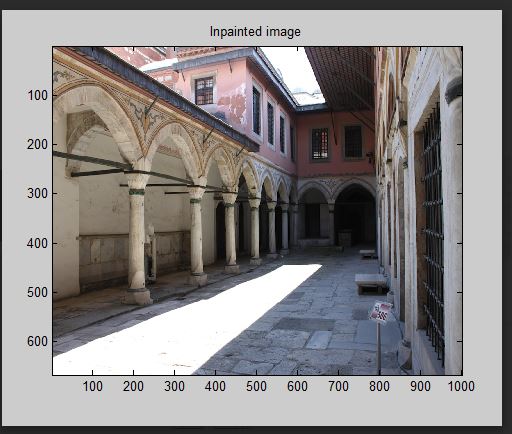


Bright area

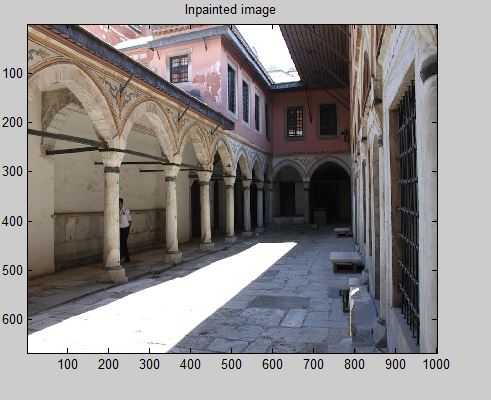


**Criminsi,Perez and Toyama Approach**

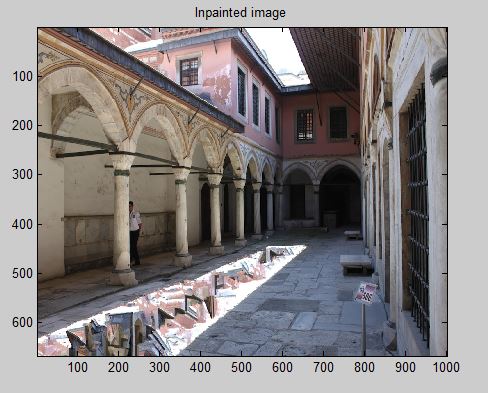
**Man on the left**

****

**The sign**

****

**Bright area**

****

**Image quilting**

Image quilting takes one block at a time for systhesis whereas Efro and Leung’s approach takes only one pixel at a time time thus taking more time. In image quleting for every pixel a bunch of blocks around it are searched for the constraint and one of them is randomly chosen. The error surface between the new and old block is chosen and the overlap region Is computed. This process is repeated till the image grows. Efro and Leung’s approach uses non parametric sampling to do texture synthesis. In this scenario the pixels are filled one at a time depending upon the sample values obtained by searching the neighbours of a pixel. Among the matches obtained in the search one of them is randomly chosen and assigned to the current pixel