**EE4212 Part 2 CA 1 Report**

Ananya Das, A0126632L

*Question 1: Texture Results*

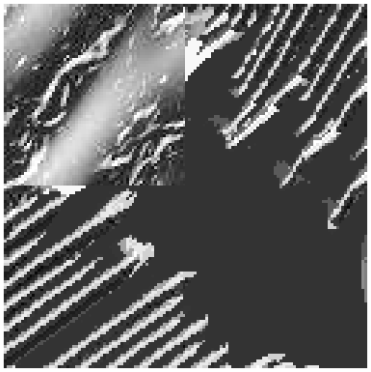

*Figure 1 Texture 1 result, window size 41*

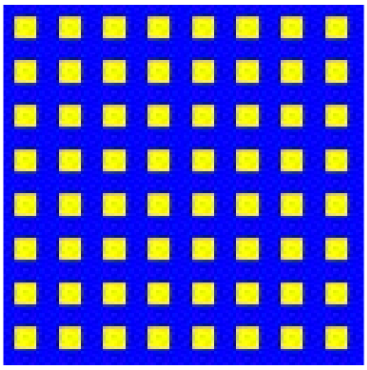

*Figure 4 Texture 4 result, window size 31*

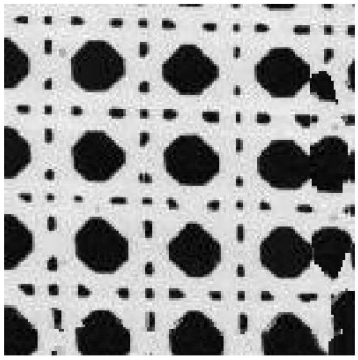

*Figure 7 Texture 7 result, window size 61*

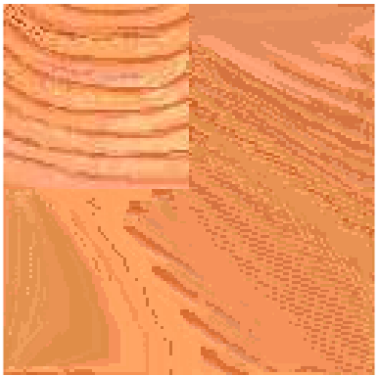

*Figure 2 Texture 2 result, window size 29*

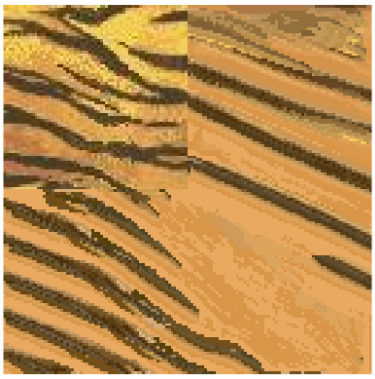

*Figure 5 Texture 5 result, window size 31*

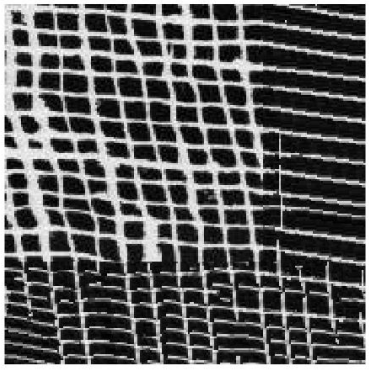

*Figure 8 Texture 8 result, window size 31*

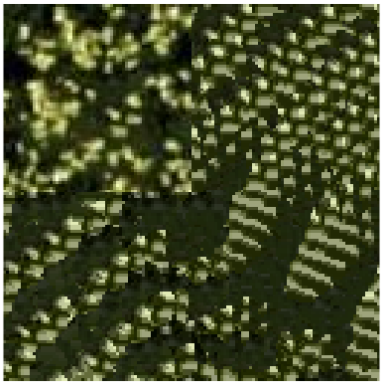

*Figure 3 Texture 3 result, window size 41*

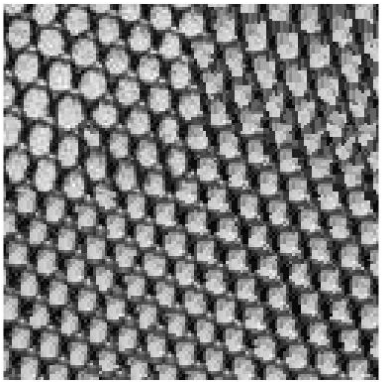

*Figure 6 Texture 6 result, window size 31*

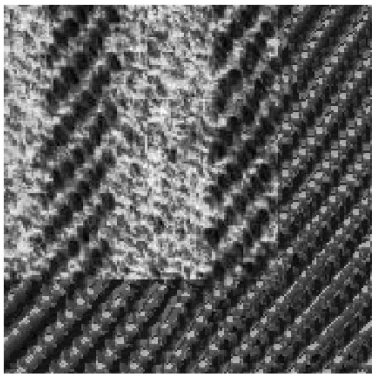

*Figure 9 Texture 9 result, window size 31*

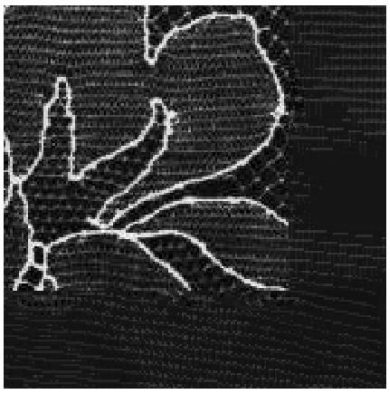

*Figure 10 Texture 10 result, window size 21*

*Figure 11 Texture 11 result, window size 41*

*Question 1 : Discussion*

A larger neighborhood or window size increases the time required for synthesis. However, it generally can help get better results for larger scale patterns if we use a larger window size compared to a small one. Comparing texture 9 and texture 7, specifically the window size used, we can see that for faithfully replicating large general patterns a larger window size is beneficial.

The window size of 31 used in texture 9 would have been too small, hence only one of the two general patterns was replicated (though this one replication is quite accurate, without significant boundary artifacts except at the boundary with the missing general texture). For texture 7, a much larger window size of 61 was used; the texture was replicated nicely when replicating downwards: the horizontal boundary is not obvious, the synthesized texture looks homogeneous along the vertical direction. However, the synthesized texture 7 suffered from a very strong and noticeable boundary artifact along the vertical boundary with the original image. Thus it would appear an adequately large window size alone may not be a sufficient solution to avoid boundary artifacts consistently and reliably.

Regarding the type of data that works best, in general, with the simple algorithm implemented in this assignment, the more uniform textures give better synthesis results.

Texture 4 could be synthesized almost perfectly, and it looks natural, because the pattern is perfectly uniform, and a large enough window size was chosen such that the spacing of the yellow squares on the blue background is maintained.

Texture 6 is also a relatively uniform texture, even if it has curves; note how the synthesized larger pattern based on texture 6 does not show significant boundary artifacts, and looks more natural compared to the other synthesized textures. The main issue observed is blurring and distortion at the upper right corner of the synthesized texture (for texture 6).

The results of the attempts to synthesize the less uniform patterns contain artifacts which make the boundary between the original texture image and the synthesized one visible or noticeable.

For texture 1, the synthesis does not capture the whole pattern, its missing the light gray gradients and the true shape of the white marks; in spite of the large window size. For texture 2, the noisy colors from the top corner of the original image got replicated into the larger synthesized image. With smaller window size, the dark lines running along the lighter background were getting too thick.

Texture 3 and texture 8 are a little bit less uniform patterns, where the spacing and shape of the yellow dots contains some irregular looking variation in texture 3, and the thickness of the white lines varies in texture 8. The results of the synthesized portion of the texture are much more uniform than the original textures; the naturalness could not be

replicated. Using a larger window size may have potentially been helpful for texture 3, but looking at the non-uniformity in texture 8 I would guess that larger window size alone would not be able to improve the results. For texture 5 the original image pattern can also be observed to be quite non-uniform, and the synthesized portion taken in parts looks somewhat natural enough; but as a whole combined with the original image , due to some boundary artifacts and color issues, overall it looks unnatural.

For texture 10 , the synthesized portion is a somewhat good replication of the main background texture of the image. The second texture in the image was too scarce and irregular to be picked up well by this synthesis algorithm. It appears it was picked up a for a very small area in the synthesized result, but the algorithm is not sophisticated to the extent to be able to detect and continue the patterns of the white lines for example.

For texture 11, the pattern is also non-uniform from the point of view of using windows…the spacing of bottom row of bricks is not replicated anywhere else in the original image, so it could not be picked up by the algorithm and hence the strange and unnatural looking synthesis result. Using a larger window size may have improved the horizontal direction synthesis, but probably would not have helped the vertical direction (the unnatural pattern on the bottom part of the synthesized image from the last row of bricks in the original image)

To improve efficiency of the algorithm, block level synthesis could be tried instead of the pixel by pixel method implemented here. The minimum error boundary cut block level could be tried, but I feel it would not work for all the textures. Alternatively, like suggested in the question itself, patch match could be used for faster synthesis, copying a whole patch into the synthesized image at a time instead of a single pixel per match.

*Question 2 : Results*

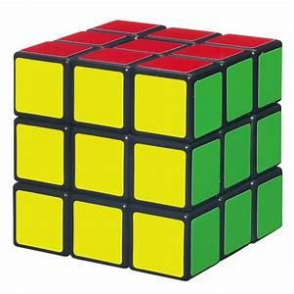

*Figure 12 Image A*

*Figure 13 Image B*

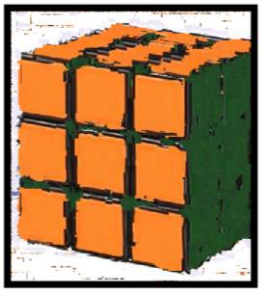

*Figure 14 Reconstructed Image*

*Figure 15 Image C*

*Figure 16 Image D*

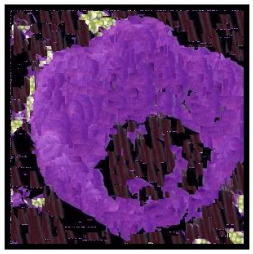

*Figure 17 Reconstructed Image C*