Image Quilting for Texture Synthesis and Transfer

Aman Kansal 170050027
Ansh Khurana 170050035
Kushagra Juneja 170050041

1. Introduction

We present a simple method which given a texture image, generates new images having the same texture. We do this by stitching together small patches of the texture image to generate a new image having same texture. We also extend the algorithm to perform texture transfer i.e. rendering an object with the texture taken from a different object. Unlike other methods which use one-pixel-at-a-time synthesis, our method uses patches of the image at a time and hence provides a great computational advantage while not compromising with the results since even in complicated textures as well the degree of freedom is only of a few pixels and hence patches are an intuitive choice.

2. Method

2.1. Quilting

The patch-based synthesis algorithm in this approach picks small blocks from the texture image and stitches them together to form a larger image that follows the same texture. This algorithm is similar to the quilting procedure used in textile where pieces of fabric are stitched together using a needle and thread.

2.1.1 Algorithm

- We construct the image to be synthesized in units of a block.
- For every location, we search the input texture and find the set of blocks which match the above and the left block in the overlapping region. Precisely, we choose those blocks whose error (using L2 norm) with the already placed blocks in the overlapping region are within some tolerance (set to 0.1) of the minimum error. Among these blocks we choose a block randomly.
- Now, we need to stitch this block with the already placed blocks. To do that, we find the optimal boundary path which minimizes the sum of errors along the path. This is done by dynamic programming. This boundary is visualized in figure 1. To the right and bottom of this boundary, we choose the pixels of the new block, and in the other region, we choose the pixels of the old blocks.

One hyperparameter is the block size B, that has to be tuned separately for all images. In all experiments, the width of the overlapping region was taken to be 1/3 of block size.

Figure 1: Boundary Cuts in Synthesis of Apple image

2.2. Texture Transfer

In the task of texture transfer, the input is taken in form of small piece of texture and a target image. The aim is to reconstruct the target image from small patches of the given texture so that the high level information in the image is preserved while giving it a textured appearance similar to the input texture.

2.2.1 Algorithm

The algorithm for this task is similar to the quilting algorithm, except that the final image is constructed in multiple iterations, each producing a better output image than the previous one. The output image is of the same dimensions as the input target image. Thus, each patch in the output image corresponds to a patch in the target image. After placing the first patch (at the top-left corner say) in the output image, we progressively place the next patch, stochastically minimizing the summation of following two losses -

- SSE (sum of squared errors) between the pixels in the new patch and those in the surrounding patches, in the overlap region. Call it SSE1
- SSE between the pixels in the new patch and the pixels in the corresponding patch of the target image. Call it SSE2

Mathematically the minimized loss is -

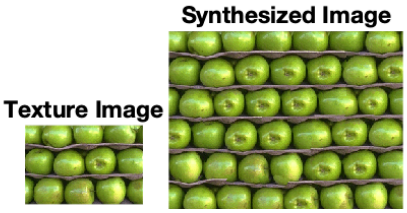
$$L = \alpha * SSE1 + (1 - \alpha) * SSE2$$

The paramter α, in i th iteration is heuristically kept as $\alpha_i = 0.8 * \frac{i-1}{N-1} + 0.1$ where N is the total number of iterations and is ideally kept between 3 to 5. The size of the patch is also reduced in each successive iteration in a geometric fashion, i.e., $B_{i+1} = B_i * d$ where d is the decay rate for patch size and a hyperparameter for the algorithm. In the successive iteration we replace the target image by the image obtained in the previous iteration.

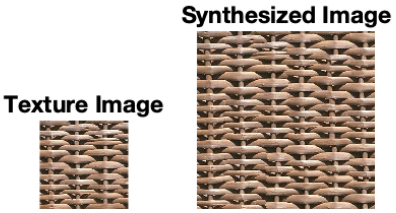
Intuition behind iterative algorithm We can observe that in the successive iterations, in addition to reducing the patch size, we alter the relative weights for the two SSE's adding more weight to maintaining continuity accross edges. The explanation behind it is as follows - We choose larger patches first (although still small enough to be able to replicate target image), so as to preserve the important components in the texture. This might reflect some discountinuities at the edges of the block. In the subsequent iterations, when smaller patches are being chosen then the patches inside the previously larger patch would tend to remain same (as they match the target imageperfectly and are inherently continuous), but the patches at the boundary would be chosen so as the remain closer to the previously chosen patch, while enhancing continuity accross the boundary.

3. Results

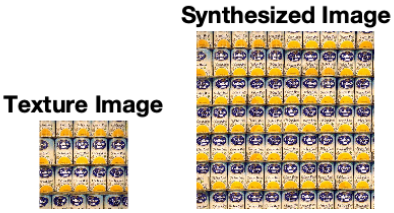
We show some amazing synthesis results below followed by transfer results. The results demonstrate the effectiveness of our implementation. The images produced are high quality and realistic for both synthesis and transfer tasks. We are able to reproduce results similar in quality when compared to the ones presented in the paper.



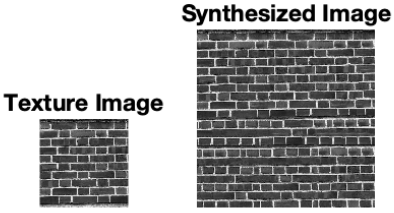
(a) Apples



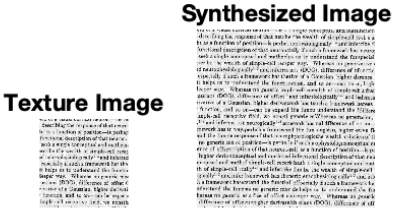
(b) Jute



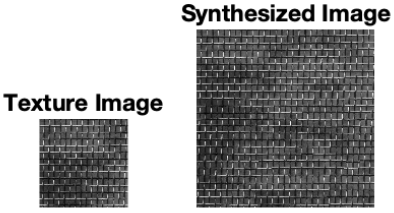
(c) Cans



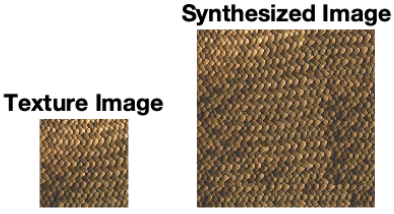
(d) D1



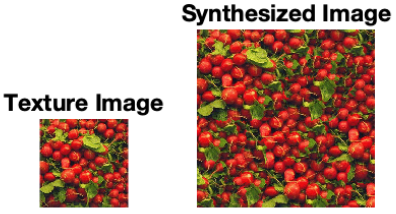
(e) text



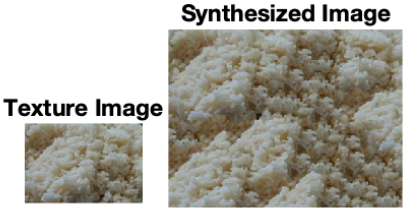
(f) Bricks B/W



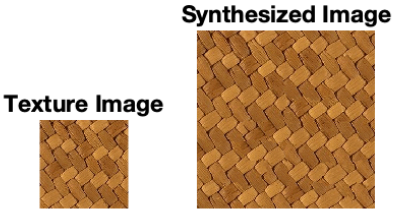
(g) Fabric



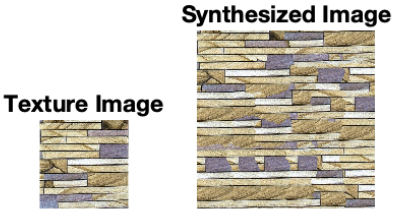
(h) Radishes



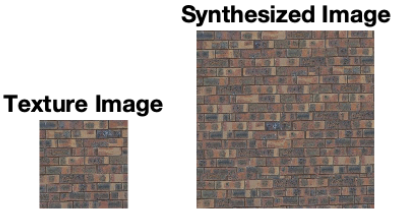
(i) Rice



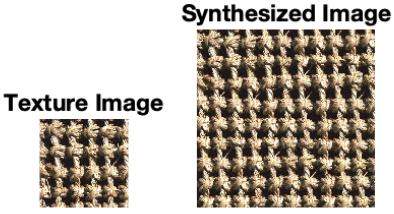
(j) Wood



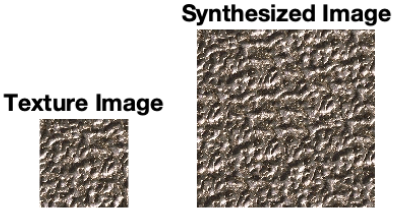
(k) Tiles



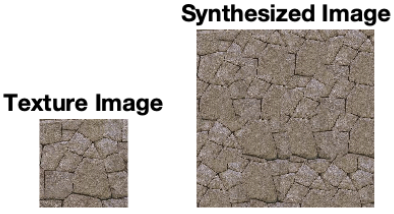
(l) Bricks



(m) Rope



(n) Pattern



(o) Stone

Figure 2: Quilting Results

(a) Input

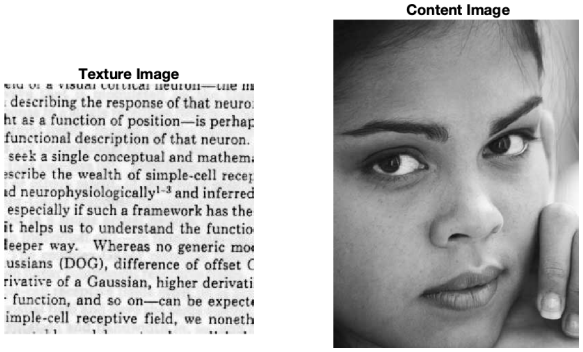
(b) Output

Figure 3: Result for texture Rice trasferred to Bill image

(a) Input

(b) Output

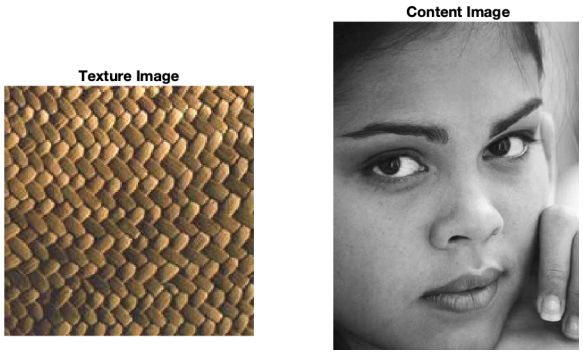
Figure 4: Result for texture Rice transferred to Girl image



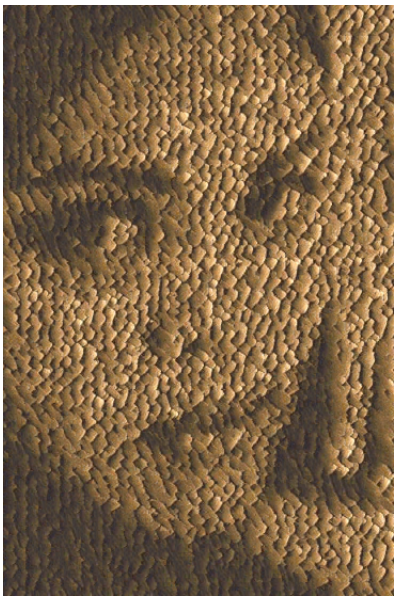
(a) Input

(b) Output

Figure 5: Result for texture Text transferred to Girl image

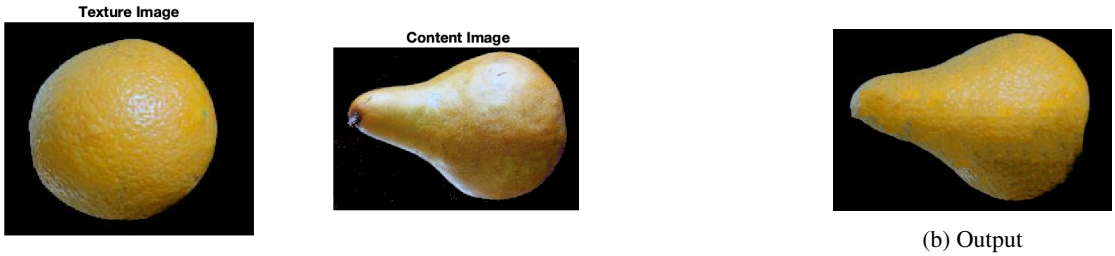


(a) Input



(b) Output

Figure 6: Result for texture Fabric transferred to Girl image



(a) Input

(b) Output

Figure 7: Result for texture Orange transferred to Pear image

(a) Input

(b) Output

Figure 8: Result for texture Pattern transferred to Girl image

4. Ablation Studies

4.1. Quilting

The effect of block size can be seen in figures 9, 10 and 11. As we can see, lower block sizes are not enough to capture the complete texture, hence lead to distorted boundaries and merging of the objects. E.g. consider the apples image, here in the case of B=20, the block size is not large enough to capture a complete apple in the overlapping region, hence in the overlapping region, we can see a lot of apples merging with each other. A very large block size is also not preferable since it reduces the number of patches we select from which means that there is less chance of adjacent patches matching with each other, hence at the boundaries of patches we will observe discontinuity.

(a) B=20

(b) B=30

(c) B=40

(d) B=50

Figure 9: Effect of block size on Apples

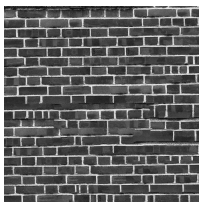
(a) B=20

(b) B=30

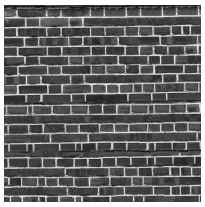
(c) B=40

(d) B=50

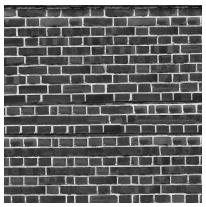
Figure 10: Effect of block size on Cans



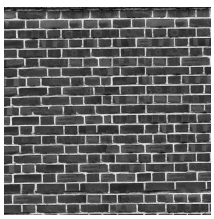
(a) B=20



(b) B=30



(c) B=40



(d) B=50

Figure 11: Effect of block size on Bricks

4.2. Transfer

Here, we study the effect of changing block size on the texture transfer result. When the block size is small, content preservation is more because the overlapping region more accurately represents the approximate intensity of the complete block hence matching that leads to a result which closely resembles the target image content. However, a small block size is not enough to capture the complete texture of the image as we saw in texture synthesis, hence the texture of the image is better preserved when the block size is larger. Hence, there is a tradeoff between content preservation and texture preservation and hence the block size must be tuned appropriately to get visually pleasing results.

We also study the output images obtained after each iteration of the algorithm. In the earlier iterations, a small value of α_i is used which means that the content preservation will be more but there will be some artifacts in the boundary of the patches and the texture will not be very clear. This is because the error weight given to matching the overlapping region with the previously places blocks is low, hence we will observe discontinuity in the edges of the patches. As we reach the later iterations, this quantity α_i is gradually increased which removes these boundary artifacts and leads to a better image texture and continuity while sacrificing a bit on the content preservation. Hence, the number of iterations must also be tuned properly for superior results.

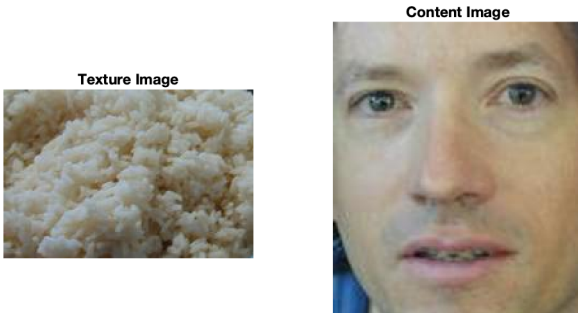


Figure 12: Rice-Bill input

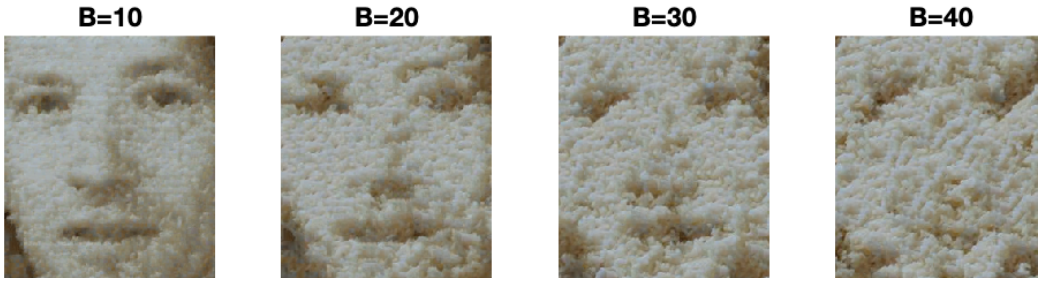


Figure 13: Rice-Bill, Block Size, B decay rate = 0.8

Figure 14: Rice-Bill, Iterations, Block size=20, B decay rate = 0.8

Figure 15: Rice-Girl input

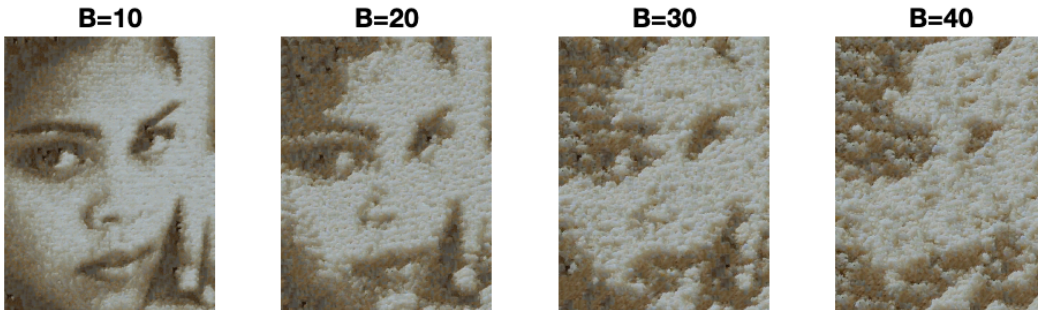


Figure 16: Rice-Girl, Block Size, B decay rate = 0.7

Figure 17: Rice-Girl, Iterations, Block size=20, B decay rate = 0.8

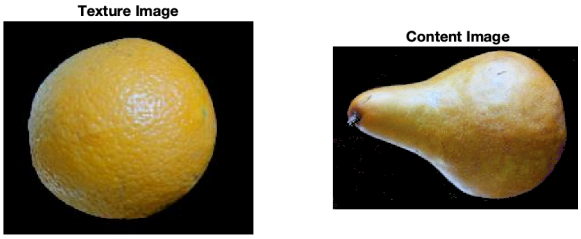


Figure 18: Orange-Pear input

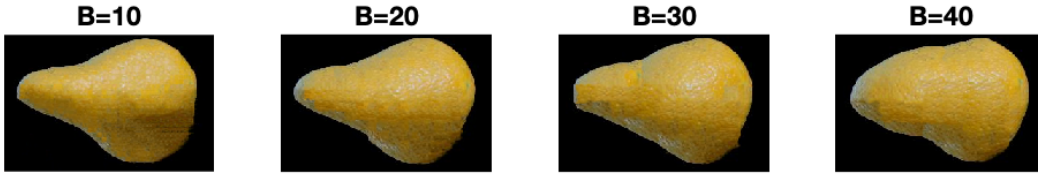


Figure 19: Orange-Pear, Block Size, B decay rate = 0.9

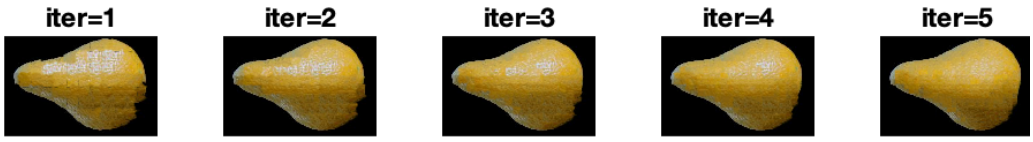


Figure 20: Orange-Pear, Iterations, Block size=20, B decay rate = 0.8