Image Quilting for Texture Synthesis and Transfer

Project Report
Neelkamal Bhuyan, 180020058
Samyak Shah, 18D070062

CS663: Fundamentals of Digital Image Processing

Indian Institute of Technology, Bombay

06 December 2020

Project Abstract

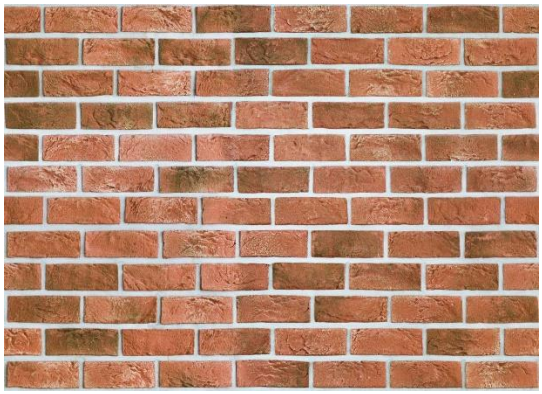
Through this course project, we intend to analyse and implement the work of *Efros and Freeman* on Image Quilting for Texture Synthesis and Transfer. Image Quilting is a novel image processing-based approach to generate new visual appearances in given images.
The paper discusses a brief history of the importance of textures in visual perception. The authors have introduced the *'quilting algorithm'* which can generate textures by stitching small patches of existing images. The method is extended to texture transfer which is quite similar to the very popular Neural Style Transfer in Deep Learning; the difference is that quilting has significantly low computational cost than its deep learning counterpart.

For reproducing the results of the paper, we will use the images that can be downloaded from http://graphics.cs.cmu.edu/people/efros/research/quilting.html. We have used images from different sources to analyse if the algorithm fails to perform well on specific types of images or textures.

Textures- a primer

Though the word texture doesn't carry an unambiguous meaning, in context of this project (and image processing in general) texture means a certain kind of repetition in a digital image. Although, in common day-to-day language, it means the structure of a surface and the word usage is often with respect to the sense of touch.
Textures can be classified into:

1) Regular Textures: They contain a certain regular pattern in the textured image.



Regular Pattern
(source: Pinterest)

Irregular Pattern
(source: Shuttertock)

2) Stochastic Texture: Textures of this kind look somewhat like a noise, for example a collection of dots scattered on a monochromatic background.

Stochastic Texture
(Source: 123RF.com)

Near Stochastic Texture
(Source: bestqwallpapers.com)

Patch Based Texture Synthesis

Patch based texture synthesis techniques create a new and larger texture by copying and stitching together texture patches. Image Quilting is one of the best-known patch-based texture synthesis algorithms. Pixel Based texture synthesis takes more time than patch-based texture synthesis since a lot of time is wasted on dealing with pixels which already know what their value should be (in the former). For the batch-based approach, information about the texture is all contained within the particular patches. Thus choosing the patch of right size plays a pivotal role.
Image Quilting deals with texture synthesis similar to fitting pieces of a jigsaw puzzle. This will further be explained in the next few sections of the report.

The Image Quilting Algorithm

As mentioned in the parent paper of this project, the image quilting algorithm is as follows:

- Going through the image to be synthesised in raster scan order in steps of one block (not including the overlap)
- For every location, we search for input texture from the texture image which satisfy the overlap constraints by the already rendered image above and on the left of it. The constraint needs to be satisfied within an error tolerance. Pick one such texture block randomly.
- Compute the error surface between the newly chosen block and the old blocks at the overlap region. Now, find the minimum cost path along this surface and make this the boundary of the new block. Paste the block onto the new texture image (the one to be synthesised). Repeat the process till the entire image is generated.

Minimum Error Boundary Cut

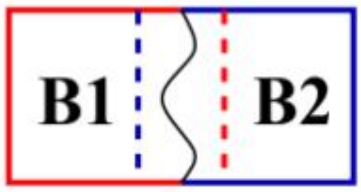
We wish to make the cut between two overlapping blocks where the two texture patterns match best (i.e. where the overlap error is low).
This can easily be achieved using Dynamic Programming (DP).

If B_1 and B_2 are two blocks that overlap over a vertical edge, With the regions of overlap called B_1^{ov} and B_2^{ov} respectively.

Let e be equal to $(B_1^{ov} - B_2^{ov})^2$. To compute the minimum error vertical cut through the surface, we calculate the cumulative minimum error E for all the paths.

$$E_{i,j} = e_{i,j} + \min\left(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}\right)$$

In the end of this procedure, the minimum value of the last row of the E matrix will indicate the end of the desired cut. We can trace back to find the desired cut fully.
A similar procedure is applied for horizonal overlaps too. In case of both horizontal and vertical overlaps, the minimal paths from the horizontal and vertical cases intersect and the overall minimum is chosen.



minimum error boundary cut

Texture Transfer

We augment the texture synthesis algorithm by adding an additional constraint that each patch satisfies a desired **correspondence map C** (this is an additional constraint imposed to the constraints already put in by the synthesis algorithm). The correspondence maps a particular quality of the target image, like its illumination. Experimenting with different correspondence maps may result in better or worse results for a particular use case.
For texture transfer, image being synthesized must respect two independent constraints:

- The outputs are legitimate, synthesised examples of the source textures.
- The correspondence mapping is strictly respected.

We introduce an additional parameter, alpha, and vary the error function by including ‘alpha’ in it. Alpha determines the trade-off between the texture synthesis and the fidelity to the target image correspondence map

Conclusions drawn from the outputs

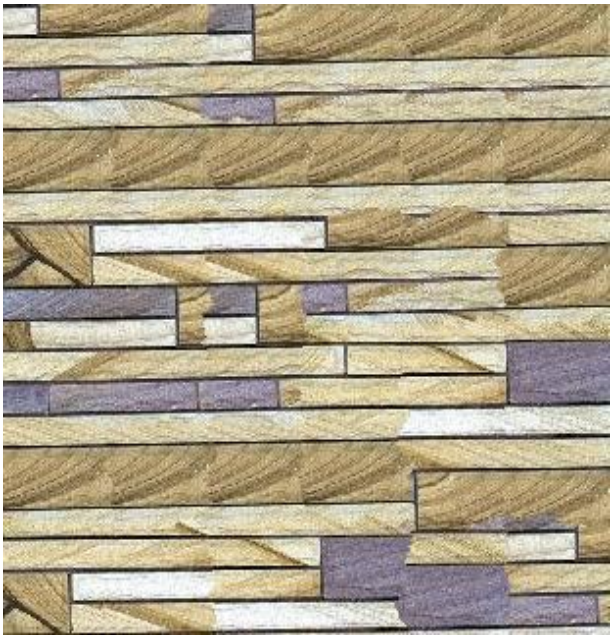
Like most contemporary algorithms for texture synthesis and texture transfer, Image quilting is not a perfect algorithm and has a few minor drawbacks.

- Image Quilting works very well for stochastic textures, but it does not work well in case of very regular textures (like a chess board or a geometric texture which has a global image logic behind its pattern)
- Texture Synthesis takes comparatively low time, but texture transfer requires a lot of time to generate an output (sometimes even around 15 to 20 minutes). This huge time makes this algorithm, in its present form, unsuitable to be deployed for customer use.
- We used a single pass over the image for texture transfer, better results can be generated if we perform multiple passes over the output image with diminishing window size with each pass.

- Major problems for the texture synthesis part were observed for the olive images, where we got distorted boundaries in the output. In case of the berries image we observed excessive repetitive patterns in the output image which look unpleasant on appearance. The patterns of the input may not always have been there on purpose!
- The time to generate the output depends on the type of image and the patch size considered by the user.
- Overall, we can conclude that Image quilting is quite an elegant approach to generate textures and transfer them. It produces magnificent results, more than what one would expect from a method of such simplicity!

Texture Synthesis Results (reproducing the results for images used in the paper)

Patch Size: 60



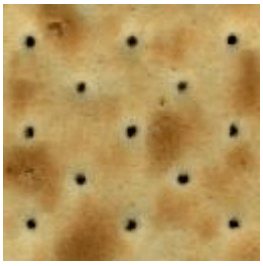
Patch Size: 60

Patch Size: 60

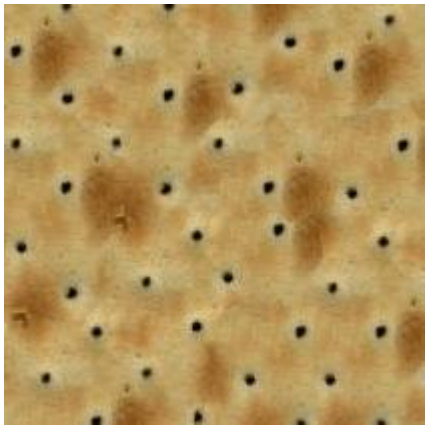
Patch Size: 60
The synthesis result for this image is quite unsatisfactory

Patch Size: 60

Patch Size: 60



Patch Size: 60



Patch Size: 60

Patch Size: 60

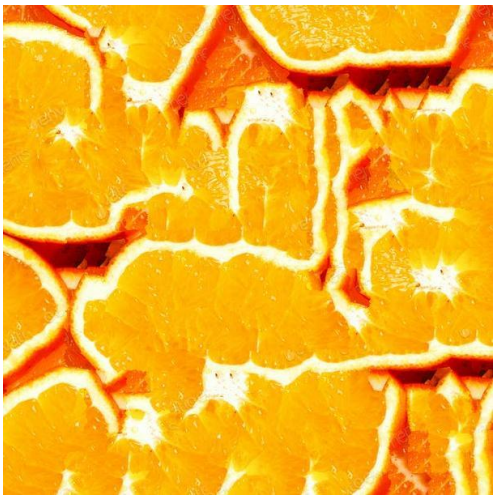
Patch Size: 60

Note: For more results please refer the folder in the zip file submission, we didn't wish to increase the length of this report unnecessarily.

Texture Synthesis Results (on our own data)

Source: elements.envato.com

This image was specifically chosen to show the limitations of the algorithm on certain type of images
(here the pattern shown is not enough to be replicated into a good output)



Source: istockphots.com

This image was specifically chosen to show the limitations of the algorithm on certain type of images. (This image is very regular, which is a difficult input for the algorithm.)

Source: libreshot.com

This image has a near stochastic pattern. Excellent results are observed!

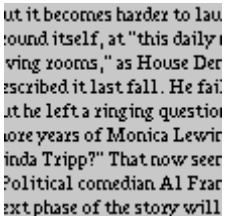
Texture Synthesis Results (variation with patch size)

Input Image:

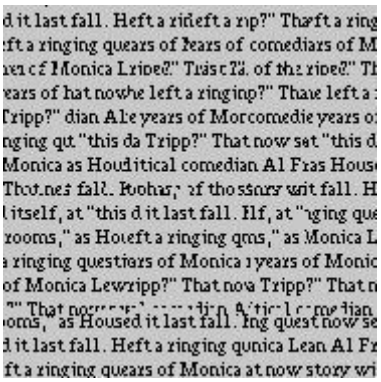
Synthesised Texture with path size = 48

Synthesised Texture with path size = 60

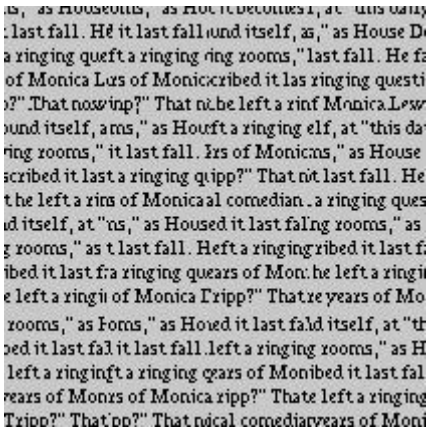
Input Image:



Synthesised Texture with path size = 36



Synthesised Texture with path size = 60



Note: For more comparisons (on other images too) please refer the folder in the zip file submission, we didn't wish to increase the length of this report unnecessarily. A very extensive image output set is present.

Texture Transfer Results (reproducing the results for images used in the paper)

Input Image:

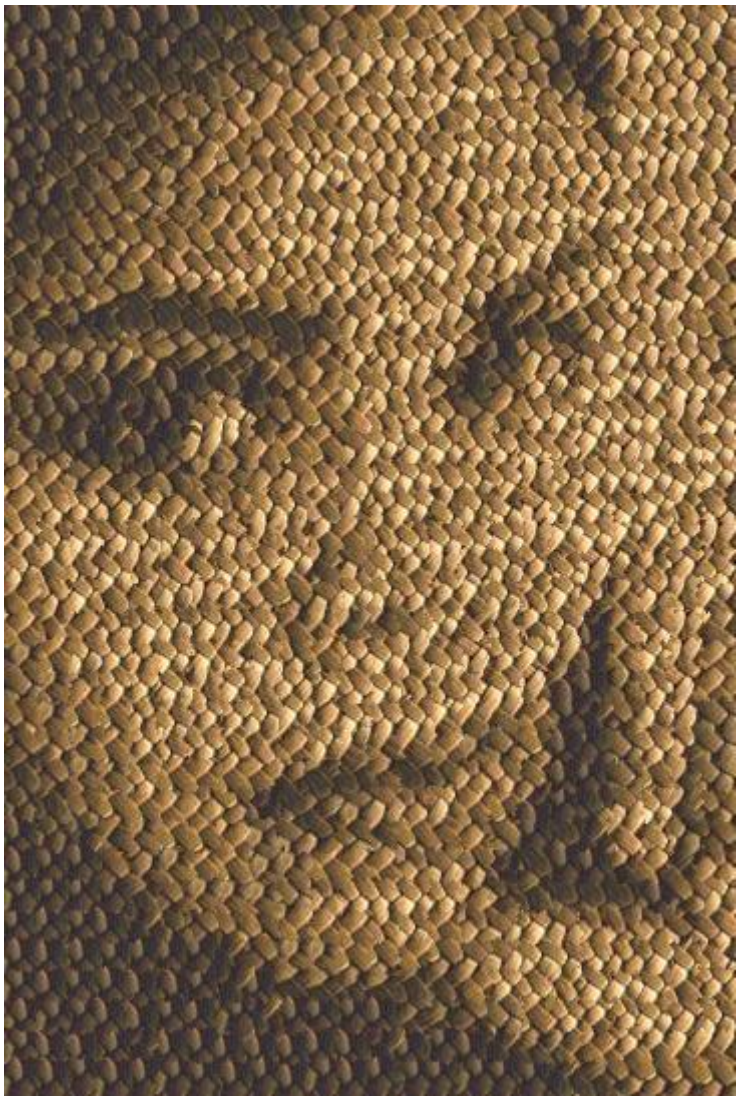
Source Texture:

Output Image with Transferred Texture:

Input Image:

Source Texture:

Output Image with Transferred Texture:



Texture Synthesis Results (on our own data)

Input Image: (source: istockphoto.com)

Source Texture: (source: shutterstock.com)

Output Image with Transferred Texture:

Input Image: (source: treeandwoodland.co.uk)

Source Texture: (source: toddcshea.com)

Output Image with Transferred Texture:

Seeing the above result we can conclude that as the source texture is pretty complicated in itself, the output image doesn’t resemble the input image much.

Input Image: (source: treeandwoodland.co.uk)

Source Texture: (source: hoodlynn.de)

Output Image with Transferred Texture:

Note: For more texture transfer results, please see the corresponding subfolder in the zip file.
