**Image Quilting**

**Adhithya.C, Dinesh kumar.N, Diyanesh.M.S**

UG Scholar Amrita School of Computing, Amrita Vishwa Vidyapeetham – Chennai

**Abstract**

The current improvements in computer graphics, VR and AR applications have resulted in the problem of texture synthesis receiving considerable attention from the computer animations and image processing research groups. In this paper we reproposed the well-known texture synthesis method, image quilting, first proposed by Efros and Freeman and tried to produce some improvements to its algorithms and implementation. Further, we outline its application to the framework suited for modern imaging applications associated with compressed image data. We also made some experiments whose results are prove the effectiveness of the algorithm

1. **Introduction**

Driven by the stringent requirements of future multimedia industry, especially interactive games, movie animations and VR applications, Secure Image Transmission texture synthesis has become an active research area within computer graphics and image processing. It is expected to be widely used in texture mapping aimed at improving the realism of computer generated images, whereby texture details are added by wrapping a given texture image around the original surface.

The demand for high-quality images has been greater in computer graphics and computer vision. Image-based rendering is playing bigger roles in video games, for instance. Rather than creating the whole physical world from scratch, we prefer pick samples from small images and synthesize realistic novel views. The advantage of texture synthesis is that it is faster and requires a smaller input dataset. Also, texture transfer is a useful tool in computer graphics. Texture synthesis is the process of creating an image of arbitrary size from a different image texture. Both texture mapping work directly on flat images and require no three dimensional information.

A texture synthesis method starts from a sample image and attempts to produce a texture with a visual appearance similar to that sample, by repeated placement of micro patterns of texture elements on a surface so that when perceived by a human observer, it appears to be generated by the same underlying stochastic process. Unfortunately, creating a robust and general texture synthesis algorithm has been proven difficult.

In this project we implement the image quilting algorithm for texture synthesis and texture transfer. The main idea is to compose a large patch of texture by “stitching together” arbitrarily shaped smaller patched from the input texture, that is, image quilting. Texture transfer would be implemented based on texture synthesis.

**Keywords :** Image Quilting, Texture Transfer, Texture Synthesis, Image based rendering, Texture Mapping

**2. Literature Review & Problem Description**

The problem of synthesizing textures has been studied extensively and numerous solutions/approaches have already been proposed. {2.1-2.4} involves the previous methods proposed and the drawbacks and {2.5} gives image quilting approach to solve the below mentioned drawbacks.

**2.1. Markov Random Fields**

One approach that has been successful in producing good quality textures uses Markov Random Fields [3,4]. This approach has proven to synthesize textures, which are good approximations to a broad range of textures. However the main drawback of these algorithms is their computational intensiveness that prevents them being used in real time texture synthesizing applications.

**2.2. Physical Simulation of Texture**

Another common approach is the physical simulation of texture. In this method texture generation is done by directly simulating the physical generation process of certain textures such as corrosion, weathering etc. Certain patterns such as fur, scale and skin are modelled using reaction diffusion [5] and cellular texturing [6]. The main disadvantage of these algorithms is that they cannot be applied to general categories.

**2.3 Statistical Feauture Matching**

Another commonly used approach is statistical feature matching. In this method certain features of the input texture are matched in constructing the resulting texture. These algorithms are more efficient than Markov Random Field algorithms. Heeger and Bergen [7] proposed a method for modelling textures by matching marginal histograms of image pyramids. This algorithm fails to give good results on structured textures. Simoncelli and Portilla [8] were able to improve the synthesis results on structures by using a complicated optimisation procedure.

**2.4 Multi Resolution Filter Based Approach**

De Bonet [10] also uses a multi-resolution filter-based approach in which a texture patch at a finer scale is conditioned on its “parents” at the coarser

scales. The algorithm works by taking the input texture sample and randomizing

it in such a way as to preserve these inter-scale dependencies. This method can successfully synthesize a wide range of textures although the randomness parameter seems to exhibit perceptually correct behavior only on largely stochastic textures. Another drawback of this method is the way texture images larger than the input are generated. The input texture sample is simply replicated to fill the desired dimensions before the synthesis process, implicitly assuming that all textures are tilable which is clearly not correct.

**2.5 Image Quilting and Texture Synthesis**

The inspiration for our work comes from the recent algorithm by Efros and Freeman [1], which has subsequently become the de-facto standard for image synthesis. The approach is simple and works well with most kind of textures.

In 1999, Efros and Leung [2] proposed a much simpler model that sampled pixels directly from similar neighborhoods in the input image. This was an important advancement in the field and several extensions/improvements have been developed since.

One such algorithm is Image Quilting [1] as proposed by Efros and Freeman. Image Quilting provides very high quality results in very manageable time, mainly by sampling patches of texture instead of individual pixels. By using patches, speed is increased by doing matching once per block instead of once per pixel and quality is increased by increasing the amount of structure captured. However an extensive experimental analysis done by us has proved that it has some basic algorithmic problems, which result in the synthesised texture being not optimal at times.

**3. Methodology**

Basically, the project involves two different methodologies to perform image quilting. “Texture Synthesis” involves the simple idea of taking random square blocks from the input texture and place them randomly onto the synthesized texture (with alpha blending to avoid edge artifacts). In “Texture transfer” we take the texture from one image and transfer it onto different objects. The result has the texture of the source image and the correspondence map values of the target image.

**3.1 Texture Synthesis**

* Go through the image to be synthesized in raster scan order insteps of one block (minus the overlap).
* For every location, search the input texture for a set of blocks that satisfy the overlap constraints (above and left) within some error tolerance. Randomly pick one such block.
* Compute the error surface between the newly chosen block and the old blocks at the overlap region. Find the minimum cost path along this surface and make that the boundary of the new block. Paste the block onto the texture. Repeat.



**Figure 1:** Texture Synthesis.

Square blocks from the input texture are patched together to synthesize a new texture sample:

 (a) Blocks are chosen randomly

(b) The blocks overlap and each new block is chosen so as to “agree”

with its neighbours in the region of overlap,

(c) To reduce blockiness the boundary between blocks is computed as a

minimum cost path through the error surface at the overlap.

**3.2 Texture Transfer**

For texture transfer, image being synthesized must respect two independent constraints:

1. the output are legitimate, synthesized examples of the source texture, and
2. that the correspondence image mapping is respected. We modify the error term of the image quilting algorithm to be the weighted sum, n times the block over-lap matching error plus (1-n) times the squared error between the correspondence map pixels within the source texture block and those at the current target image position.

Ei,j = ei,j + min(Ei-1,j-1 , Ei-1,,j , Ei+1,j+1)

The parameter determines the trade-off between the texture synthesis and the ﬁdelity to the target image correspondence map. Because of the added constraint, sometimes one synthesis passthrough the image is not enough to produce a visually pleasing result. In such cases, we iterate over the synthesized image several times, reducing the block size with each iteration. The only change from the non-iterative version is that in satisfying the local texture constraint the blocks are matched not just with their neighbour blocks on the overlap regions, but also with whatever was synthesized at this block in the previous iteration. This iterative scheme works surprisingly well: it starts out using large blocks to roughly assign where everything will go and then uses smaller blocks to make sure the different textures ﬁt well together.

**2.3. Efros and Freeman Image Quilting Algorithm**

Initialization, patch selection, computation of the minimum error border, and blending procedure along the boundary are all steps in the proposed sequential patch-based approach. It consists of producing output texture images that are both aesthetically comparable to and pixel-wise distinct from the input, and perhaps greater in size, given an input texture image. The approach for stitching a fresh patch in the sequentially created output texture to eliminate discontinuities as much as feasible is the method's key novelty. This is accomplished by determining the best cut between the patch and the synthesis area. The input sample and output texture are denoted by

I0 and Is, respectively. The output image is constructed patch by patch in a raster scan order. The goal of each iteration is to fill a patch Pold of Is that is only partially defined on a region called overlap region. To do so a patch Pin of I0 that matches Pold on the overlap region is randomly selected. Within the overlap zone, an ideal boundary cut between Pold and Pin is determined. This optimal boundary cut is used to construct the new patch Pnew by blending Pold and Pin along the cut.

• The initialization of Is the first step in the algorithm. A random patch Pin of size wp\*wp is taken from I0 and placed at the top-left corner of Is for this purpose.

• The square distance between the overlap region of the patch Pold of Is and the equivalent regions of all the patches of I0 is computed to choose a patch Pin of the input image I0. The minimum distance dmin is calculated, and Pin is chosen at random from all patches with a distance to Pold of less than (1 +ϵ) dmin, where is the tolerance value.

• In the overlap region, the Patch Search step returns a patch Pin of I0 with coordinates (m, n) that is identical to the partially specified current patch Pold. There are three sorts of overlap zones that appear in the raster scan order: vertical, horizontal, and L-shaped overlap. The patches Pold and Pin must be combined to create the final patch P.

1. **Experimental Results and Analysis**

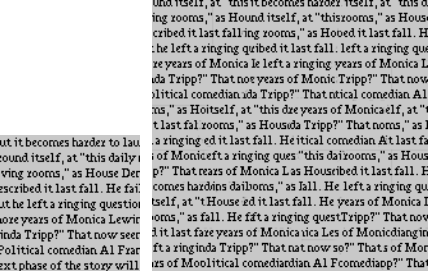
The below sections provides us the results of experiments of both the techniques involved in this project.

* 1. **Texture Synthesis**

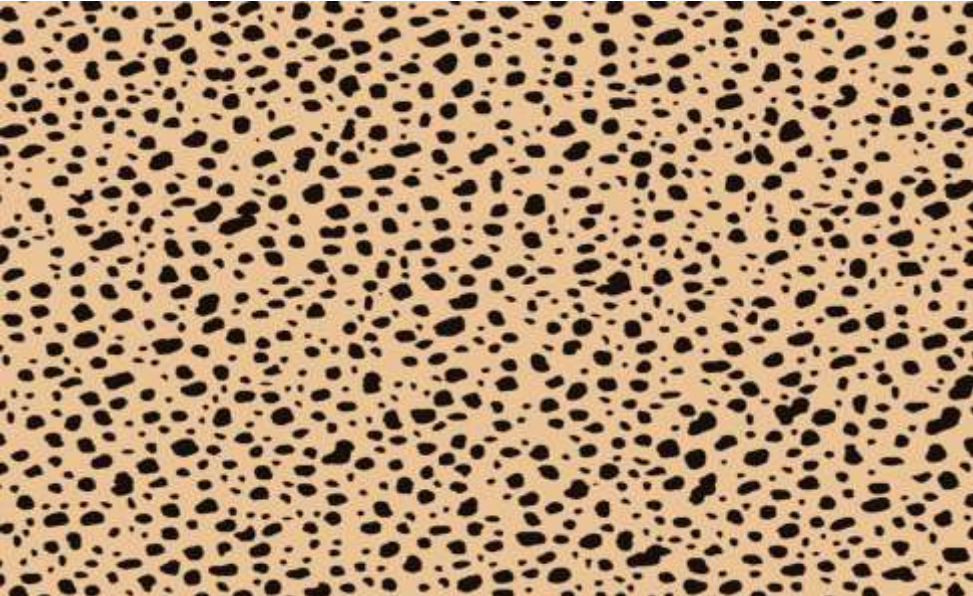
lts The results of the synthesis process for a wide range of input textures are shown on Figures 2-4. While the algorithm is particularly effective for semi-structured textures (which were always the hardest for statistical texture synthesis), the performance is quite good on stochastic textures as well. The two most typical problems are excessive repetition (e.g. the berries image), and mismatched or distorted boundaries (e.g. the mutant olives image). Both are mostly due to the input texture not containing enough variability. Figure 6 shows a comparison of quilting with other texture synthesis algorithms.

****

**Figure 2:** Windows

****

**Figure 3:** Text

** **

**Figure 4:** Jute

* 1. **Texture Transfer**

Our texture transfer method can be applied to render a photograph using the line drawing texture of a particular source drawing; or to transfer material surface texture onto a new image which is cleary shown in Figures [5-7].

For texture transfer, image being synthesized must respect two independent constraints: (a) the output are legitimate, synthesized examples of the source texture, and (b) that the correspondence image mapping is respected. We modify the error term of the image quilting algorithm to be the weighted sum, α times the block overlap matching error plus (1 - α ) times the squared error between the correspondence map pixels within the source texture block and those at the current target image position. The parameter determines the tradeoff between the texture synthesis and the fidelity to the target image correspondence map.

Because of the added constraint, sometimes one synthesis pass through the image is not enough to produce a visually pleasing result. In such cases, we iterate over the synthesized image several times, reducing the block size with each iteration. The only change from the non-iterative version is that in satisfying the local texture constraint the blocks are matched not just with their neighbour blocks on the overlap regions, but also with whatever was synthesized at this block in the previous iteration. This iterative scheme works surprisingly well: it starts out using large blocks to roughly assign where everything will go and then uses smaller blocks to make sure the different textures fit well together.

**Figure 5 :** Texture of rice transferred to man’s face

**Figure 6 :** Texture of fabric transferred to girl’s face

**Figure 7 :** Texture of orange peel transferred to banana peel

The algorithm is not only trivial to implement but is also quite fast: the unoptimized MATLAB code used to generate these results ran for between 15 seconds and several minutes per image depending on the sizes of the input and output and the block size used. Because the constraint region is always the same it’s very easy to optimize the search process without compromising the quality of the results

**4. Conclusion and Future work**

In this paper we have introduced a novel approach to synthesizing textures under a multi resolution framework. We have provided experimental results and an in-depth analysis, proving that the proposed method works remarkably well producing equivalent (or better) output texture quality as compared to the method proposed in papers [3]-[10], as a much less computational cost. The multiresolution nature of the proposed framework also makes it easily applicable to modern imaging applications needing progressive transmission capabilities.

At present we are extending the idea further in two directions, namely, increasing the implementation efficiency and preventing seam visibility using adaptive edge cutting techniques. Finally we aim to test the proposed algorithm in mapping real texture onto the surfaces of modelled 3D objects as a with the interest in contributing to the growth of VR/AR applications.

1. **References**

[1] A.A. Efros and W. T. Freeman. "Image Quilting for Texture Synthesis and

Transfer", Proceedings of SIGGRAPH el, Los Angeles, California,

August, 2001.

[2] A.A. Efros and T.K Leung. "Texture Synthesis by Nonparametric Sampling".

In Proc. IEEE International Conference on Computer Vision, 1999.

[3] K.Popat and R.Picard. Novel cluster based probability model for texture

synthesis, classification and compression. In Visual Communication and

Image Processing. Pp. 756- 768, 1993.

[4] R.Paget and I Longstaff. Texture synthesis via noncasual nonparametric

multi-scale Markov Random Field. IEEE Transactions on Image Processing,

7(6): 925-931, June 1998.

[5] A.Witkin and M.Kass. Reaction-diffusion textures. In Computer Graphics

(SIGGRAPH ’91 Proceedings), July 1991.

[6] S.P.Worly. A cellular texture basis function. In H. Rushmeirer, editor,

SIGGRAPH ’96 conf. Proc., Annual conf. series, pp. 291-294, Aug 1996.

[7] D.J.Heeger and J.R.Bergen. Pyramid-Based texture analysis/synthesis. In

SIGGRAPH ’95, pages 229- 238, August 1995.

[8] E.Simonceli and J.Portilla. Texture characterization via joint statistics of

wavelet coefficient magnitudes. In Fifth International conference on image

processing, Volume 1, pages 62-66, October 1998.

[9] D.S.Wickramanayake, Eran.A.Edirisinghe, Helmut Bez. Image Quilting and

Texture Synthesis : A revisit and Variation. January 2004

[10] J. S. D. Bonet. Multiresolution sampling procedure for analysis and

synthesis of texture images. In SIGGRAPH ’97, pages 361–368, 1997.

[11] Afrin Fathima and Shoby Sunny. Image quilting Algorithm for Texture

Synthesis. IRJMETS Volume:03/Issue:09/September-2021