Image Quilting

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

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

***Abstract:- The issue of texture synthesis has drawn a lot of attention from the computer graphics and image processing research community due to the current trends in computer animation, virtual reality, and augmented reality applications. In this study, we explore the popular picture quilting texture generation technique, first put out by Efros and Freeman, and suggest enhancements to its implementation and algorithms. We also describe how it can be applied inside a multiresolution framework, which is ideal for contemporary imaging applications that work with compressed picture data. To demonstrate the efficiency of the suggested alterations and variations, we present experimental findings.***

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

1. INTRODUCTION

Secure Image Transmission texture synthesis is currently a hot topic of research in the fields of computer graphics and image processing due to the demanding specifications of the future multimedia sector, particularly interactive games, movie animations, and VR applications. It is anticipated to be widely utilised in texture mapping, which adds texture features by wrapping a specified texture image around the original surface, to enhance the realism of computer-generated images.In computer graphics and vision, there has been an increasing demand for high-quality images. Rendering using images is crucial in video games. We choose to select samples from small photographs and create realistic perspectives rather than starting from scratch to create the full actual image The speed and limited input dataset needed for texture creation make it advantageous. A useful tool for computer animations and visuals is texture transfer. The process of producing a picture of any size from a sample image texture is known as texture synthesis. Both types of texture mapping can be used on flat images without the need for 3-D information. By repeatedly placing micro patterns of texture elements on a surface, a texture synthesis method attempts to create a texture with a visual appearance similar to that sample, giving the impression to a human observer that it was created by the same underlying stochastic process. Unfortunately, it has turned out to be challenging to develop a reliable texture synthesis algorithm.

In this project, we put the image quilting algorithm which is used to  create and transfer textures. The fundamental goal of image quilting is to create a large texture patch by "stitching together" several smaller patches of the input texture that have been randomly formed. The implementation of texture transfer would be based on texture synthesis.

1. LITERATURE REVIEW & PROBLEM

DESCRIPTION

Numerous approaches and solutions have already been put forth for the difficulty of synthesizing textures, which has been the subject of extensive study. {A-D} summarizes the previously suggested solutions and their shortcomings, and {E} offers an image quilting strategy produced by us to address the problems listed above.

*A. Markov Random Fields*

Markov Random Fields are one of the approaches that has been effective in creating textures of high quality [3,4]. It has been demonstrated that this technology can create textures that are accurate approximations of a wide variety of textures. However, this method has a significant drawback in that it is computationally expensive, making it impossible to use it for real-time texture synthesising applications.

*B. Physical Simulation Of Texture*

The physical simulation of textures is another popular technique. In this approach, textures are physically created by simulating the creation of textures like corrosion and weathering. Reaction diffusion [5] and cellular texturing [6] are used to model several patterns, including hair, scales, and skin. These algorithms fundamental drawback is that they can't be used for all computer-based applications due hardware intensive processes.

*C. Statistical Feauture Matching*

Statistical feature matching is another strategy that is commonly employed. This method matches some input texture attributes to create the output texture. Compared to techniques using Markov Random Field, these algorithms achieved effective outcomes. A technique for modelling textures by matching the marginal histograms of image pyramids was put out by Heeger and Bergen [7]. On structured textures, this approach does not produce satisfactory results. By using a some optimisation techniques, E. Simoncelli and J. Portilla [8] were able to produce superior synthesis outcomes for structures, but the techniques were complicated.

*D. Multi Resolution Filter Based Approach*

Using a multi-resolution filter-based technique, De Bonet [10] compared a texture patch at a smoother scale to its progenitors at a rougher scale. These inter-scale dependencies are kept when the input texture is randomly generated by the method. Despite the randomness parameter appearing to only display perceptual behaviour on textures that are largely stochastic, this method can successfully synthesise a wide variety of textures. The fact that the created texture pictures are larger than the input is another disadvantage of this method. Internally, the synthesis procedure simply begins by duplicating the input texture sample in the necessary dimensions. This is obviously erroneous because it assumes that all textures are tilable.

*E. Image Quilting*

The recent approach by Efros and Freeman [1], which has since evolved into the de-facto standard for image synthesis, served as the basis for our study. The strategy is straightforward and suitable for most textures. A far more straightforward model that sampled pixels directly from similar neighbourhoods in the input image was put out by Efros and Leung [2] in 1999. Numerous expansions and enhancements have been created since this significant advancement in the industry.Image Quilting [1], an algorithm by Efros and Freeman, is one example. Image quilting, which samples texture patches rather than individual pixels, produces extremely high-quality results in a very manageable amount of time. By employing patches, quality is improved by increasing the amount of structure collected and speed is raised by performing matching once per block rather than once per pixel. However, a thorough experimental analysis carried out by us has demonstrated that it has some fundamental algorithmic issues, which occasionally cause the synthesized texture to be subpar.

III. METHODOLOGY

In essence, the project uses two separate image quilting approaches. The basic concept behind "Texture Synthesis" is to randomly insert square blocks from the input texture onto the synthesized texture while using alpha blending to prevent edge artefacts. We use "Texture transfer" to apply the texture from one image to various objects. The final product includes both the target image's correspondence map values and the source image's texture.

*A. Texture Synthesis*

Raster scan the image to be synthesised in steps of one block (with the overlap removed).Find a group of blocks that satisfy the overlap constraints (above and left) at each place by searching the input texture within a certain error tolerance. Choose a block like this at random.

Calculate the error surface at the overlap zone between the newly selected block and the previous blocks. Make the boundary of the new block the lowest cost path along this surface. the texture, then paste the block there. Repeat.



**Figure 1:** Texture Synthesis.

To create a new texture sample, square blocks from the input texture are patched together:

(a) Blocks are picked at random

(b) The blocks overlap, and each new block is chosen to be in agreement with its neighbours in the zone of overlap.

(c) The shortest cost path through the error surface at the overlap is used to determine the block

boundary to reduce blockiness. The following

formula calculates minimal error boundary cut.

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

*B. Texture Transfer*

To transfer a texture, an image must adhere to two

separate requirements:

(a) The output must be accurate synthesised representations of the source texture.

(b) Corresponding image mapping must be adhered to. We change the image quilting algorithm's error term to the weighted sum of (1-n) times the squared error between the correspondence map pixels within the source texture block and those at the current target image position, together with n times the block over- lap matching error.

The parameter affects the trade-off between the fidelity to the target picture correspondence map and the texture synthesis. One synthesis run across the image may not always be enough to give an aesthetically attractive outcome due to the added limitation. In these situations, we repeatedly iterate over the synthesized image, lowering the block size as we go. The blocks are not just only matched with their neighbour blocks on the overlap regions, but also with textures synthesized at this block in the before iteration, in order to satisfy the local texture constraint, which is the only difference from the non-iterative version. Surprisingly, this iterative approach works well. It begins by using giant blocks to approximately determine where everything will go and then utilises smaller blocks to make sure the various textures blend in nicely.

*C. Efros and Freeman Image Quilting*

*Algorithm*

The proposed sequential patch-based approach consists of the following steps: initialization, patch selection, computation of the minimum error border, and blending along the boundary. It entails creating output texture images that, given an input texture image, are perhaps larger in size and are both aesthetically similar to and pixel-for-pixel different from the input. The main originality of the method is the procedure for sewing a new patch into the output texture that is produced sequentially in order to minimise discontinuities. Finding the ideal cut between the patch and the synthesis area allows for this. I0 and Is stand for the input sample and output texture, respectively. The output image is created in a raster scan order, patch by patch. Filling a patch Pold of Is that is only partially specified on a region known as the overlap region is the aim of each iteration. To do this, a randomly chosen patch Pin of I0 that matches Pold on the overlap region is chosen. An optimal boundary cut between Pold and Pin is established within the overlap zone. By combining Pold and Pin along the cut, the new patch Pnew is created using this ideal boundary cut.

* The algorithm's first step is to initialise I. For this, a random patch pin of size wp\*wp is taken from I0 and positioned at the top-left corner of Is.• To select a patch Pin from the input picture I0.
* The square distance between the overlap region of the patch Pold of Is and the equivalent regions of all the patches of I0 is calculated. Pin is selected at random from all patches having a distance to Pold of less than (1 + ϵ) dmin, where dmin is the minimum distance, where is the tolerance parameter.
* The Patch Search step provides a patch Pin of I0 with (m, n) coordinates that is identical to the partially given current patch Pold in the overlap region. Vertical, horizontal, and L-shaped overlap zones are the three types of overlap zones that can be seen in the raster scan order. The final patch P must be made by combining the patches Pold and Pin.

1. EXPERIMENTAL RESULTS AND ANALYSIS

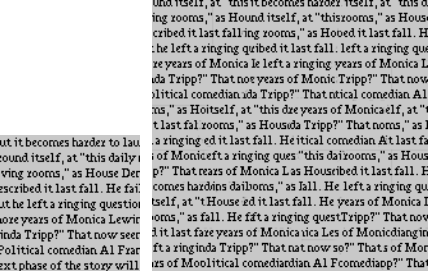
The below sections provides us the results of experiments of both the techniques involved in this project.

1. *Texture Synthesis*

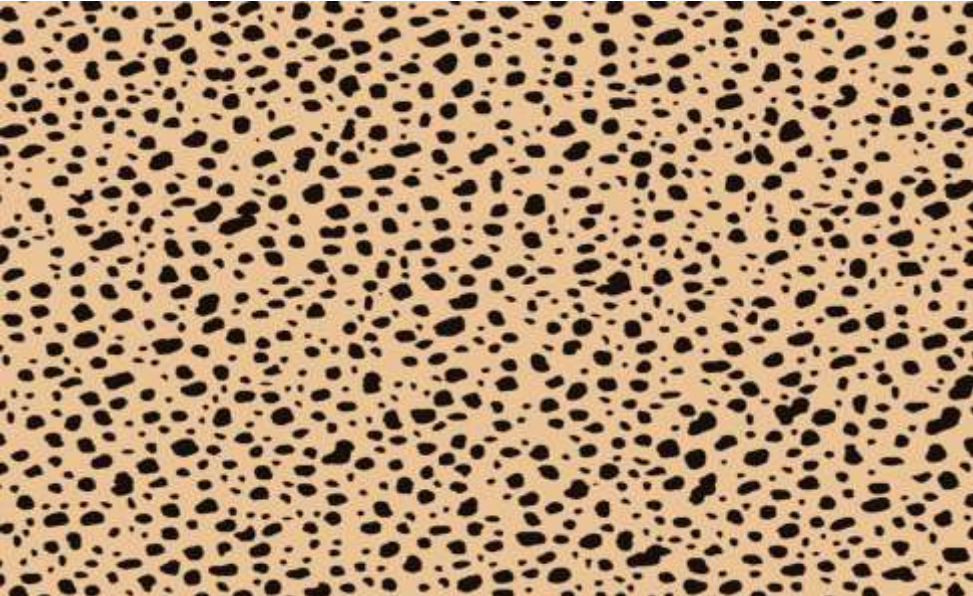
lts Figures 2-4 display the outcomes of the synthesis process for a variety of input textures. While stochastic textures also perform well, semi-structured textures—which have always been the most challenging for statistical texture synthesis—benefit most from the algorithm's effectiveness. Excessive repetition and misaligned or distorted boundaries are the two most common issues. Both are mostly caused by the input texture's lack of sufficient variety.

****

**Figure 2:** Windows

****

**Figure 3:** Text

** **

**Figure 4:** Jute

1. *.Texture Transfer*

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.

One synthesis run across the image may not always be sufficient to generate an aesthetically attractive outcome due to the added limitation. In these situations, we repeatedly iterate over the synthesised image, lowering the block size as we go.The blocks are not just only matched with their neighbour blocks on the overlap regions, but also with textures synthesized at this block in the before iteration, in order to satisfy the local texture constraint, which is the only difference from the non-iterative version. Surprisingly, this iterative approach works well. It begins by using giant blocks to approximately determine where everything will go and then utilises smaller blocks to make sure the various textures blend in nicely.

**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 optimised MATLAB code used to produce 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 utilised. The approach is not only simple to develop but also rather quick. It is quite simple to optimise the search process without sacrificing the quality of the results as the constraint region is always the same.

V. CONCLUSION AND FUTURE WORK

In this paper, we present a unique method for texture synthesis in a multi-resolution framework. The suggested method works amazingly well, producing output texture quality that is equivalent to (or better than) that of the method proposed in articles [3]–[10], but requiring significantly less computer resources. We have provided experimental findings and a thorough analysis to support this claim. The proposed framework's multiresolution features also make it simple to adapt to contemporary imaging applications that require progressive transmission capabilities.

Using adaptive edge cutting techniques, we are currently trying to extend the concept further in two directions: improving implementation effectiveness and reducing seam visibility.. 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 Non-

parametric 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