[[1]](#footnote-1)

IMAGE QUILTING IMPLEMENTATION

*By – Kshitij Kumar 19BCE1170, Devesh Gupta – 19BCE1579, Utkarsh Tiwari – 19BCE1723*

# **INTRODUCTION**

The demand for high-quality graphics has become enormous in computer graphics and computer vision. Graphic-based rendering plays a major role in video games, for example. Rather creative the whole world visible from the beginning, we choose to select samples from small images and combine facts view of the novel. The beauty of combining texture is that it is fast and requires a small set of input data. Also, texture transfer is a useful tool for computer graphics. Text synthesis is a creative process an unnecessary size image comes from a different image texture. Both sewing a map works directly on a flat surface images also do not require third party information.

In this project we use an image quilting algorithm to combine texture and texture transfer, proposed in the paper SIGGRAPH 2001 by Alexei A. Efros and William T. Freeman. Main the idea is to design a large piece of texture by "joining together" a small attachment that is irregularly shaped

from captivity to input, that is, to close the image. Texture transfer will be used based on sewing to combine. Another report is organized as follows: We begin by analyzing the motive for the release of the image. We then discuss the use of seamless integration and sequence transfer sequences and present simulation results that show how the method meets the required requirements. We discuss the algorithm and present some conclusions.

# **MOTIVATION & PROBLEM STATEMENT**

As discussed in this paper, although old one pixel at a time synthesis algorithm work well, there is a big problem with this algorithm. The fact is that of the many pixels of complex texture there are a few a selection of prices that can be offered. Once the algorithm has started, most pixels are like that cut so completely that it would be a waste of time to search for these pixels. The query solving ability of computer durability and long working hours, the research paper provides a "quilting image", which we carry make a texture map like assembling a jigsaw puzzle, assembling the pieces, making sure they are all Gather together.

This algorithm has two important advantages over the pixel-at-a-time method. The first benefit that is, it greatly reduces the amount of search required, by transferring large patches at a time. This leads to a significant reduction in operating time which makes it easier to use a less powerful computer devices to perform such functions. Second, by transmitting large patches at a time, it ensures visibility consistency within any means of transmission. However, the potential danger is that there are increasing opportunities for visible boundaries texture produced in areas where patches are stitched together. The paper also introduces you solutions to this type of issue.

# **LITERATURE SURVEY**

Text analysis and integration have a long history in psychology, mathematics, and computer theory. Gibson showed the importance of visual acuity, but it was Bela Julesz's pioneer work on textual discrimination that paved the way for field development. Julesz suggested that the two statues of the statues would be seen by human observers as if they were the same. This suggests that the two main functions in compiling mathematical textures are selecting the correct mathematical set to match, and finding the same algorithm. Encouraged by the psychophysical and computational models of human texture discrimination, Heeger and Bergen proposed analyzing the texture following histograms of filtering responses on multiple scales and guidelines. Comparing these histograms repeatedly is enough to produce amazing compositional results for stochastic texture (see for theoretical justification).

However, since histograms assess limits rather than collections, statistics do not capture significant relationships across all scales and orientations, so the algorithm fails to have a more structured texture. Compared to these two-dimensional figures, Portilla and Simoncelli were able to significantly improve the integration results of a structured fabric at the expense of a complex development process.

In the above methods, the texture is combined by taking a random sound image and forcing it to have the exact same figures as in the input image. The opposite approach is to start with the inserted image and do it randomly in such a way that only the figures to be matched are set. De Bonet defines the rough input to a fine, which preserves the conditional distribution of filter extraction over multiple scales (jets). Xu el.al., inspired by the Clone Tool in PHOTOSHOP, suggests a more simple approach that produces similar or better results. The idea is to take random square blocks from the input holder and place them randomly in a composite texture (with alpha stitching to avoid artifact edge).

The data stored here is simply a pixel setting within each block. Although this process will fail at highly structured patterns (e.g. chess board) due to boundary instability, in many stochastic texts it works remarkably well. The related method has been successfully used by Praun et.al. with a semiautomatic text of undeveloped objects. Enforcing statistics worldwide is a daunting task and none of the above algorithms provide a completely satisfactory solution. A simple problem is forcing statistics into place, one pixel at a time. Efros and Leung developed a simple way to “enlarge” the texture using non-parameter samples. The conditional distribution of each pixel given to all its included neighbors to date is measured by searching the sample image and finding all the same locations. (We recently learned that a similar algorithm was proposed in 1981 by Garber but was discarded because of its computer inactivity.) combine all pixels!). Several researchers have suggested a basic methodology that includes Wei and Levoy (based on previous work by Popat and Picard), Harrison, and Ashikhmin. However, all of these improvements still work within the greedy paradigm of one-pixel-time and as a result are in danger of falling into the wrong part of the search site and start “growing garbage”. The approaches are made primarily of dedicated domains that capture the spirit of our intentions in transmitting texture. Our goal is similar to that of work in non-photographic delivery (e.g. [4, 19, 15]).

The main difference is that we want to express the output delivery style by taking samples in the real world. This allows for a wealth of giving styles, characterized by samples of images or drawings. Lots of papers to be published this year, all developed independently, are closely related to our work. The concept of transmitting text-based textures has been suggested by several authors [9, 1, 11] (in particular, see the excellent paper by Hertzmann et.al. on these processes). U-Liang et.al. suggest a real-time patch-based method that is very similar to ours. The student is encouraged to review these activities in order to get a complete picture of the field.

# **PROPOSED WORK**

**Image Quilting Algorithm:**

The complete quilting algorithm is as follows:

Go through the image to be merged with a raster scan in steps of one block (remove spacing).

Everywhere, search the input text to find a set of blocks that meet the overlap (above and left) inside

tolerating a particular mistake. Randomly select one such block.

Calculate the error area between the newly selected block and the old blocks in a spaced area. Find a low-cost route to this area and create that new block border. Attach the block to the seam. Repeat.

Block size is the only user-controlled parameter and depends on the texture structures provided; the block should

be large enough to photograph suitable buildings for captivity, but small enough to leave connections between these buildings

up to the algorithm.

In all of our experiments the width of the horizontal edge (on one side) was 1/6 the size of the block. The error is computerized using the L2 process at pixel values. The error tolerance was set to be within 0.1 times the maximum match block error.

# **EXPERIMENTAL SETUP**

In this section we will use the texture synthesis process. The main idea behind The algorithm is to combine square text "patch" from image input to merge outgoing image. In particular there are three ways to create a composite texture. The first idea is to choose randomly patches from the texture and stitch them together. This stupid approach is actually successful in a a limited set of completely repetitive textures. Many of the semi-structures textures, however, are the result the pictures will have clear boundaries in them where the neighboring leaflets meet.

The best solution is to take patches and put them on the floor with scattered patterns sample patch has a small error scattered over previous (left or top) pars on full region. Neighboring points are selected based on a certain degree of proportional spacing. This will lead to borders that are not very different but that are still visible in most cases. Reducing artifactes on the visible edge, the best solution is to find a small error border cutter path to the local error location with the Seam Carving algorithm and manage this correct path as the boundary of the scattered region. The concept of final development in the algorithm to allow that neighboring tracts have prevailing boundaries. With this algorithm change, patches are neighboring can now be selected by selecting a path in the override that minimizes the error rate. It creates an image quilting algorithm, and greatly improves the quality of the combined texture pictures. The diagram below gives an overview of the algorithm

Random placement of neighboring blocks forced spacing

The algorithm is described below:

(1) in order to produce a large target image that is repeated in the size of the pond.

(2) Everywhere in the picture, we randomly sample from the source image

scattered with pre-selected pads within the tolerance of a particular error. For each candidate

a patch from the source source image calculates a randomly selected error

peaches in a remote area, and select a patch with a small error.

(3) In order to reduce the visible edge, a flexible system is used to obtain small cuts in full region. We calculate the error zone between pre-selected and correct pads a patch in a remote area. Finally, the minimum error path in the error seam is obtained by stitching

carpentry algorithm is used as the boundary for a new episode.

Heart algorithm scanning output image. To achieve the desired quilting behavior, scanning in this algorithm proceeds in pixel steps (patchSize - overlapSize). This is the key to finding a significant decrease in processing time compared to the method of one pixel at a time.

We now discuss three key steps in detail:

Step a: Select a tract

During each new duplication when we need to select the most appropriate patch to paste in the current location, error rate procedure to perform a complete search of the potential power of all possible patches image upload, calculate the error of each clip, keep all the tracts meet the conditions for better fit, and, finally, randomly select one of the saved ports. The first problem to consider is choosing what should be used to measure the error. The paper claims their error rate is calculated based on the L2 pixel values ​​process. With RGB color photos, however, each pixel contains not one value, but three. In our use we calculated the error the number of pixels given to be the L2 value of the weight difference. The error value of the pool is total pixel error values ​​for all pixels in the perimeter of the pond.

Step b: Calculate the error location The paper describes the location of their error in terms of the square difference between the values ​​of something pixel. In our use we also use the L2 process with the same weight as the previous step and get no the main difference between these two error options.

Step c: Find a lower cost limit

If the error location is calculated, all information is required to identify the minimum cost border crossing is at hand. In the paper, the authors describe the cost of the compact method, E, based on the pixel error values, ei, j, in the wrong place. In the straight part of the scattering pond, they calculate the minimum the cost of path, E, from pixel in the first row to the current pixel repeated by line indication, i, and column index, j, as follows:

Ei, j = ei, j + min (ei-1, j-1, ei-1, j, ei-1, j + 1) (i = 2… N)

Minimum error limit cut can be determined by searching the last (Nth) line of a scattered patch, pointing to a small amount, and following. For both tracts direct and horizontal scattering, the authors see that the cost and horizontal cost methods will meet in the middle and the minimum may be selected as a vertical standing position and horizontal trace backs.

There are other parameters that we need to consider in depth, the size of the pool and the spacing size, for example. In this paper, there is only one user parameter that can set, the size of the page. In fact, the penalty is higher for choosing a smaller pool size than for choosing a larger pool size.

The main danger is that the patch size is too large that the image that will appear will be too duplicate. IF the size of the patch is very small, the important texture structures can be completely missed. We also explore different skipping size to see performance.

In the section that analyzes how to transfer texture. In fact, it is used on the basis of texture synthesis method. The difference is that we use a merging algorithm for each patch satisfy the contact map you want and complete the current texture design requirements.

In order to transmit texture, the image created must comply with two independent limits: (a) output is official, integrated examples of source texture, and (b) image effect should match the text image. On paper, the default name for the image quilting algorithm it is converted into a measured amount to meet the needs. The alpha parameter determines the tradeoff between texture design and fidelity to the contact image of the target image.

For an additional limitation, the repetition method is applied over the compiled image several times, and the size of the block is reduced by each repetition. The deceptive part compared to the non-repetitive part that we must maintain the quality of the images included in the previous duplication. This the iterative scheme starts by using large blocks to distribute almost where everything will go and be used

small blocks to make sure the different textures blend well. For direct implementation of texture transfer, the correct sample patch in

source image should also comply with the following two barriers:

(1) The sample patch has a small matching error with pre-selected patch in the scattered area.

(2) The sample package should have a small error linking the target image to what you want

position.

So the error name should cover these two issues and look like this totalError = alpha \* overlapError + (1 - alpha) \* correspondenceError

'OverlapError' represents the difference between a valid patch and a previous merger a patch in a remote area. And 'correspondenceError' represents an error between optimal a patch and a patch from a target image in the desired location. It is a tradeoff to choose alpha from there

balances texture design and reliability on the contact map of the target image. Through the repetition process, we add an item to the equation to maintain production quality image with previous effect during each repetition, Therefore, the full error name looks like this

totalError = alpha \* (overlapError + previousSynthesizedError) + (1 - alpha) \* communicationError

To get the best performance, we need to repeat 3 to 5 times and adjust the alpha according to the formula

Alpha = 0.8 \* (currentIteration - 1) / (totalIterationNumber - 1) + 0.1

# **CONCLUSION**

In this paper we analyzed and used the image quilting algorithm suggested by Efros and Freeman in the SIGGRAPH paper 2001. We have discussed the algorithm in detail, to apply certain stories and show the results of both methods. We also discussed performance according to the results to identify the strengths and weaknesses of the algorithm. All in all, a picture The quilting algorithm works well for a wide class of texture. Produces made of high quality very fast texture images. One good extension might incorporate edge recognition techniques in an algorithm for improving some of the border problems continuously.

There are also some challenges to using this algorithm. First, the error correction process it hurts because we have to cut a lot of variables in a system algorithm that doesn't work solidly control. The choice of texture, block size and tolerance all affect the quality of the result. Second, paper used smaller size images compared to experiments. Computer power plays a big role in us combine or transfer to larger images especially if the recurring number is increased for improvement level.

# **REFERENCES**

1. Zelinka, S., & Garland, M. (2002). Towards Real-Time Texture Synthesis with the Jump Map. Rendering Techniques, 2002, 13th.
2. Zhou, K., Huang, X., Wang, X., Tong, Y., Desbrun, M., Guo, B., & Shum, H. Y. (2006). Mesh quilting for geometric texture synthesis. In ACM SIGGRAPH 2006 Papers (pp. 690-697).
3. Darzi, A., Lang, I., Taklikar, A., Averbuch-Elor, H., & Avidan, S. (2020). Co-occurrence Based Texture Synthesis. arXiv preprint arXiv:2005.08186.
4. Efros, A. A., & Freeman, W. T. (2001, August). Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 341-346).
5. Kolář, M., Debattista, K., & Chalmers, A. (2017, May). A subjective evaluation of texture synthesis methods. In Computer Graphics Forum (Vol. 36, No. 2, pp. 189-198).
6. Copeland, A. C., Ravichandran, G., & Trivedi, M. M. (2001). Texture synthesis using gray-level co-occurrence models: algorithms, experimental analysis, and psychophysical support. Optical Engineering, 40(11), 2655-2673.
7. Lin, W. C., Hays, J., Wu, C., Kwatra, V., & Liu, Y. (2004). A comparison study of four texture synthesis algorithms on regular and near-regular textures. Tech. Rep.
8. Cheng, L., Vishwanathan, S. N., & Zhang, X. (2008, June). Consistent image analogies using semi-supervised learning. In 2008 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

1. [↑](#footnote-ref-1)