International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 16 (2014), pp. 1667-1672

http://www. irphouse.com

Tiled Texture Synthesis

Komal Arora[1], Radha Kathuria[2]

[1]SPGOI College of Engineering, MD University, Rohtak (India)
[2]M.Tech Student, SPGOI College of Engineering, MD University, Rohtak (India)

Abstract

Texture Synthesis main purpose is to construct a large digital image from a small digital sample image by taking advantage of its structural content. It is object of research to computer graphics and is used in many fields, 3D computer graphics and post-production of films. It can be used to fill in holes in images, create large non-repetitive background images and expand small pictures. It is important for rendering synthetic images and animations.

1. Introduction

Texture synthesis is one of the hottest topics in the field of Computer Graphics, Computer Vision and Image Processing. The goal of texture synthesis is as follows: We are provide with a texture sample, we synthesized a new texture that, when seen by a human observer, appears to be generated by the same underlying process. Nowadays neighborhood-based methods are popularly used in texture synthesis, including point-based techniques [1] and patch-based techniques [2]. Hybrid texture synthesis method has also already been developed. These methods attempt to synthesize larger textures by copying selected regions (pixels or patches) of the sample texture and, in some way, obscuring region boundaries. Most of the methods can produce high quality results, but the speeds are very slow. They cannot avoid the heavy searching process in order to get a suited pixel or patch for the output image.

2. Literature Review

Recently, there were many real time texture synthesis algorithms presented. Liang et al. [3] use quad tree pyramid, optimized kd-tree and principal components analysis (PCA) to accelerate the patch placement process, but simply alpha blends the overlap regions (feathers). Zelinka and Garland [4] present a two-phase method towards real-time texture synthesis. They examine the input texture at length first, building a data structure which is called a jump map to store several similar pixels for every pixel in the input texture, and then only a random pixel choice according to the probability in

the jump map is needed in the synthesis phase. Wang etal [5] extend the jump map technique to patch-based maps and get better results than the previous one. Cohen et al. [2] present a new stochastic algorithm to non-periodically tile the plane with a small set of Wang Tiles [6].

Existing approaches of Texture Synthesis:

Science based texture synthesis:

Textures are created by using scientific ideas, models, algorithms and procedures. Any good and powerful scientific model in biology, physics, chemistry or mathematics can be used to create textures. Examples are like reaction-diffusion. Genetic algorithms created in computer science to solve optimization problems using Darwinian concepts of evolution in biology. Genetic algorithms allow objects to evolve with reproduction, selection and fitness. Sims showed that genetic algorithms can be used to create textures

Image based texture synthesis:

Textures are created directly from sample images using digital signal processing techniques. In this case, no prior information about the nature of the textures in the images. These types of textures are based on research on human perception that concludes that two textures are difficult to discriminate when the orientation and distribution of certain characteristics are similar [7].

3. Challenges

Major Challenges faced in Texture Synthesis are:

3.1 Modeling

In this technique the major challenge is How to estimate the texture generation process from a given finite texture sample. The estimated process should be able to model both the structural and stochastic parts of the input texture. The success of modeling is defined by the visual fidelity of the synthesized textures with respect to the given samples.

3.2 Sampling

In Sampling the issue is How to develop an efficient sampling procedure to produce new textures from a given model. The productivity of the sampling procedure will directly determine the computational cost of texture generation.

4. Proposed Methods

The following below methods have been researched or developed for texture synthesis:

Tiling

The very easy and simple way to generate a large image from a sample image is to tile

it. This means that multiple copies of the sample are simply copied and pasted side by side. The result is barely satisfactory. Except in few cases, there will be the seams in between the tiles and the image will be highly repetitive.

Stochastic texture synthesis
Stochastic texture synthesis methods produce an image by randomly choosing color values for each pixel, which only influenced by basic parameters like minimum brightness, average color or maximum contrast. These type of algorithms perform well with stochastic texture; otherwise they produce completely unsatisfactory results as they ignore any kind of structure within the sample image.

Single purpose structured texture synthesis
Algorithms of that family use a fixed procedure to create an output image, i.e. they are fixed to a single kind of structured texture. Thus, these algorithms can both only be applied to structured textures and only to textures with a very similar structure. For example, a single purpose algorithm could produce high quality texture images of stonewalls; yet, it is very different that the algorithm will produce any viable output if given a sample image that shows pebbles.

Chaos mosaic
This method, proposed by the Microsoft group for internet graphics, is a polished version of tiling and performs the following three steps:

1. The output image is filled completely by tiling. The outcome is a repetitive image with visible seams.
2. Randomly selected parts of random size of the sample are copied and pasted randomly onto the output image. The outcome is a rather non-repetitive image with visible seams.
3. The output image is filtered to smooth edges. The outcome is an acceptable texture image, which is not too repetitive and does not contain too many artifacts. Still, this method is not satisfactory because the smoothing in step 3 makes the output image look blurred.

Pixel-based texture synthesis
These methods, such as "Texture synthesis via a non causal nonparametric multi scale Markov random field"[9], "Texture Synthesis by Non-parametric Sampling" [10] and Fast Texture Synthesis using Tree-structured Vector Quantization" [7] are some of the simplest and most successful general texture synthesis algorithms. They generally synthesize a texture in scan-line order by finding and copying pixels with the most similar local neighborhood as the synthetic texture. These methods are very beneficial for image completion. They can be constrained, as in "Image Analogies", to perform many impressive tasks. They are typically accelerated with some form of Approximate Nearest Neighbor method since the exhaustive search for the best pixel is somewhat slow.

Patch-based texture synthesis
Patch-based texture synthesis creates a new texture by copying and stitching together textures at various offsets, identical to the use of the clone tool to manually synthesize a texture. "Image Quilting." is the best known patch based texture synthesis algorithms. These algorithms tend to be more powerful and faster than pixel-based texture synthesis methods.

5. Algorithms

Adjacent Tiling
This is the simplest method of Tile texture synthesis. In this method the tiles are placed in the mesh according to the desired resolution of the image. The tile is repeated for n times horizontally and m times vertically to obtain a image of n*m where the resolution of the obtained pattern if ((n*tileswidth)*(m*tileheight)) where tilewidth and tileheight are the width and height of the tiles respectively.

Overlap Tiling
This is another method of Tile texture synthesis. This method is little more complex than the previous one. In this method the tiles are placed in the mesh according to the desired resolution of the image by overlapping the one pixel at the end of each row. The tile is repeated for n times horizontally and m times vertically to obtain a image of n*m where the resolution of the obtained pattern if ((n*tileswidth)*(m*tileheight)) where tilewidth and tileheight are the width and height of the tiles respectively.

6. Advantages

Various advantages of New Methods over previous methods

Quality
A texture generated by our approach has good visual quality; they are perceived to be very similar to the input sample textures. They and can be of arbitrary size.

Generality
Our algorithm can model a vast variety of textures, despite the versatility of their underlying physical generation process.

Simplicity
Our approach is very easy, simple and can be implemented using standard image processing operations.

Efficiency
Unlike previous approaches, our algorithm is efficient. Typical textures take only seconds to minutes to generate

7. Future Work

Promising future extensions would be to develop a tiled texture with some more techniques like Blending function method and to create a automated tool for Texture quality improvement and synthesis, and to provide more precise control over input image and type. Another extension would be to select a random texture from the given image and creating the wallpaper from that texture and also using the background texture. This idea could be extended further to include texturing for many different shapes including pentagons, hexagons, and octagons. Perhaps seamless textures could even be created for a surface with many different shapes [11]. Animated textures would be another interesting extension. More wallpaper patterns could easily be added as well. Wallpaper with more boundary types and more background images would provide more variation and provide a more a periodic looking image.

8. Conclusion

In this paper a simple method to create a tiled texture from any image has been created. The program facilitates easy texture creation with Adjacent tiling method and Overlap tiling method. The method allows the user to choose any resolution of choice as well as required numbers of rows and columns. Adjacent tiling method is good for stochastic textures which do not have specified boundaries; hence seam problem will not occur whereas overlapping tiling method is good for all types of textures most likely for regular and stochastic. The mentioned methods also allows for creation of some simple wallpapers because Adjacent method for specific types of textures and Overlapping method creates seamless 2D textures it proves to be very useful for creating a periodic textures. The methods are easy to understand and implement.

8. References

[1] A. A. Efros and T. K. Leung.Texture synthesis by nonparametric sampling. In ICCV ’99: Proceedings of the International Conference on Computer Vision-Volume 2, page1033, Washington, DC, USA, 1999. IEEE Computer Society.

[2] Q.Wu and Y. Yu. Feature matching and deformation for texture synthesis. ACM Trans. Graph., 23(3):364–367, 2004.

[3] L. Liang, C. Liu, Y.-Q.Xu, B. Guo, and H.-Y.Shum.Real-time texture synthesis by patch-based sampling.ACMTrans.Graph, 20(3):127–150, 2001.

[4] S. Zelinka and M. Garland. Towards real-time texture synthesis with the jump map. In EGRW ’02: Proceedings of the13th Euro graphics workshop on Rendering, pages 99–104,Aire-la-Ville, Switzerland, Switzerland, 2002. EurographicsAssociation.

[5] B. Wang, J. Yong, and J. Sun. Real-time texture synthesis with patch jump maps. In International Symposium on Computational and Information Sciences (CIS’04), pages 1155–1160, 2004.

[6] Li-Yi Wei, Texture Synthesis by Fixed Neighborhood Searching, A

dissertation report, November 2001.

[7] L.-Y. Wei. Tile-based texture mapping on graphics hardware. In HWWS ’04: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware, pages 55–63, New York, NY, USA, 2004. ACM Press.

[8] Weiming Dong Ning Zhou Jean-Claude Paul, Optimized Tile-Based Texture Synthesis, published in Graphics Interface, Montreal, Canada: 2007.

[9] Paget and Longstaff,"Texture synthesis via a non causal nonparametric multiscale Markov random field.", IEEE Trans. on Image Processing, 1998.

[10] Efros and Leung,"Texture Synthesis by Non-parametric Sampling.", ICCV, 1999,

[11] Li-Yi Wei Marc Levoy, Texture Synthesis over Arbitrary Manifold Surfaces, A thesis.