International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 4, April - 2013

The Novel Approach To Enhance Image Using Texture Synthesis

Amruta G. Chakkarwar
Computer Science & Engg Department
Sipna C.O.E.T, Amravati.

Abstract

Image Enhancement is found to be a very effective technique useful in today's digital image processing applications. Generating novel photo-realistic imagery from smaller examples has been widely recognized as significant in computer graphics. A wide number of applications require realistic textures to be synthesized for object decoration in virtual scenes. Basically, Texture synthesis is used to create large non-repetitive background images and expand small pictures by removing noise and also to fill in holes in images. The Primary goal of a Texture Synthesis process is to synthesize a new texture in such a way that when it is been perceived by some human observer, it appears to be generated by same underlying process.

Texture synthesis is the process of algorithmically constructing a large digital image from a small digital sample image by taking advantage of its structural content. Texture synthesis can be used to fill holes in images, create large non-repetitive background images and expand small pictures. Texture synthesis is the method of cleaning the image by using patches or pixels for making the image resolution higher and better than the original image.

This dissertation addresses two parts. First part addresses brief introduction about texture, texture synthesis, and various types of Texture Synthesis and finally different application of Texture Synthesis. While second part of dissertation addresses methods for cleaning gray scale image, these two methods are Pixel Based Texture Synthesis and Patch Based Texture Synthesis.

General Terms

Pixel Based Texture Synthesis and Patch Based Texture Synthesis.

Keywords - Image enhancement techniques, Texture Synthesis, Patch Based Texture Synthesis, Pixel Based Texture Synthesis, Texture Synthesis.

1. Introduction

Texture is a ubiquitous experience. It can describe a variety of natural phenomena with repetition, such as sound (background noise in a machine room), motion (animal running), visual appearance (surface color and geometry), and human activities (our daily lives). Since reproducing the realism of the physical world is a major goal for computer graphics, textures are important for rendering synthetic images and animations. However, because textures are so diverse it is difficult to describe and reproduce them under a common framework.

1.1 What is a Texture?

Reproducing detailed surface appearance is important to achieve visual realism in computer rendered images. To map an image, either synthetic or digitized, onto the object surface, a technique called *texture mapping* .The mapped image, usually rectangular, is called a *texture map* or *texture*. A texture can be used to modulate various surface properties, including color, reflection, transparency, or displacements. In computer graphics the content of a texture can be very general; in mapping a color texture, for example, the texture can be an image containing arbitrary drawings or patterns.

1.2 What is Texture Synthesis?

Texture synthesis is an alternative way to create textures. Because synthetic textures can be made any size, visual repetition is avoided. Texture synthesis can also produce tile able images by properly handling the boundary conditions. It is the method of cleaning the image by using patches or pixels for making the image resolution higher and better than the original image .The process of constructing a large digital image from a small digital sample image algorithmically is known as Texture Synthesis.

Basically, Texture synthesis is used to create large non repetitive background images and expand small pictures by removing noise and also to fill in holes in images. The Primary goal of a Texture Synthesis process is to synthesize a new texture in such a way that when it is been perceived by some human observer, it appears to be generated by same underlying process.

The goal of texture synthesis can be stated as follows: Given a texture sample, synthesize a new texture that, when perceived by a human observer, appears to be generated by the same underlying process (Figure 1).

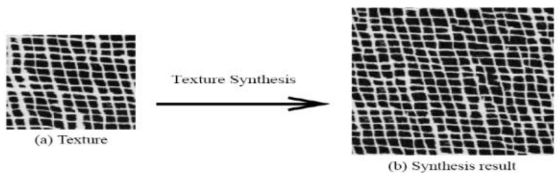

Figure 1.Texture sample synthesized in to new texture

2. Applications of Texture Synthesis

Texture synthesis can be useful in a lot of applications in computer graphics, image processing, and computer vision. Some of them are listed below

2.1 Rendering-

In rendering, textures can mimic the surface details of real objects, ranging from varying the surface's color, perturbing the surface normal's (bump mapping), to actually deforming the surface geometry (displacement mapping).

2.2. Animation-

Computer generated animations often contain scripted events and random motions. Scripted events are non repetitive actions such as opening a door or picking up an object, and are usually rendered under direct control. On the contrary, random motions are repetitive background movements such as ocean waves, rising smoke, or a burning fire. These kinds of motions have indeterminate extent both in space and time, and are often referred as temporal textures.

2.3 Compression-

Images depicting natural scenes often contain large textures regions, such as a grass land, a forest, or a sand beach. Because textures often contain significant high frequency information, they are not well compressed by transform based techniques such as JPEG. By segmenting out these textured regions in a pre processing step, they might be compressible by a texture synthesis technique. In addition to image compression, texture synthesis can also be employed for synthetic scenes containing large amounts of textures.

2.4 Computer Vision-

Several computer vision tasks use textures, such as segmentation, recognition, and classification. These tasks can benefit from a texture model, which could be derived from a successful texture synthesis algorithm.

3. Methods of cleaning Gray Scale Image

For cleaning gray scale image two methods are used .These methods are Pixel Based Texture Synthesis and Patch Based Texture Synthesis.

3.1. Patch Based Texture Synthesis

This set of methods construct the output texture map by taking image patch samples from the input image and then stitching them together so that they look like a continuous texture. This technical report explores several methods that use different ways for selecting the patches from the input texture image and several techniques for stitching them together so that their appearance seems continuous.

Patch-based texture synthesis creates a new texture by copying and stitching together textures at various offsets. These algorithms tend to be more effective and faster than Pixel-based texture synthesis methods. Synthesization: Synthesizing high quality textures as they can maintain global structure of the texture. But slow when the synthesized image is large. Time: Compare to pixel based texture synthesis this is faster .Execution: Execution is faster than pixel based texture synthesis. Seams: Here seams are present but we overlapping the patch so remove the seams. The method synthesis's a new image by stitching together small patches from the sample image. In this method synthesis's a result image block by block in raster order. Square blocks are used to capture the primary pattern in the sample texture. First, a block is randomly selected from the sample image and pasted into the new image beginning at the first row and the first column. Then another block is selected as a candidate neighbour. It is placed next to the first block so that they overlap one another.

3.1.1 Chaos mosaic-

The chaos mosaic technique proposed by Xu et al. relies on using randomness and a deterministic stochastic function to model the stochastic nature of the given sample image in the process of patch selection and placement .In order to generate a chaos mosaic texture the first step is to construct an initial output texture by repeating the input image sample and therefore constructing a tiled texture. Then a random patch is selected from each of those tiles. The selected random patch should be of size less than the size of each tile itself and the paper suggests a size between 0.5 and 0.75 times each tile, where each tile has the same dimensions as the input image sample.

They also mention that more than one random patch might be selected from each tile but they used only one random patch per tile in their experiments. The next step is placing those random patches on top of the tiled texture.

3.1.2 Image Quilting Algorithm-

The Image Quilting algorithm proposed by Efros & Freeman introduces the notion of minimum error cut boundaries between tiles to solve the weakness of the blending method described in Chaos Mosaic. It also presents another algorithm for selecting and placing the patch samples based on Euclidean distances between neighboring regions. The first part of the algorithm starts by choosing an initial random patch from the input texture and placing this patch in the top left corner of the output texture.

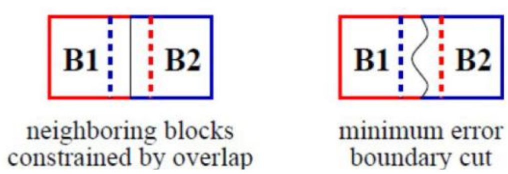

Figure 2.First step is choosing the consecutive patches based on the overlap constraint and then Calculating the minimum error boundary cut between consecutive patches.

3.1.3. Graphcut Textures-

The Graphcut Textures proposed by Kwatra et al. represents the image or portions of the image as a weighted graph, where each pixel is a node in the graph and the edge weight or cost between two adjacent nodes is given by some measure of error between adjacent pixels.

They initially propose a measure using a simple difference between pixel intensities and then refine this cost function by adding local frequency information into account. In its most simple form, the graph cut method can be used to find the cut between two adjacent patches instead of using the minimum error cut boundary proposed by Efros & Freeman.

3.2. Pixel Based Texture Synthesis

The basic framework for this type of methods is defining the model that will better represent the process that produced the given input image and then simulate this process to generate the output texture of arbitrary size one pixel at a time. This section presents the method used by Wei & Levoy .The simplicity of their single resolution method for deciding a new pixel on the output image exemplifies a typical pixel based texture synthesis approach and their attempts to improve performance show the penalty on efficiency incurred by their proposed basic single resolution method and the efficiency issues of several similar pixel based approaches. They typically synthesize a texture in scan-line order by finding and copying pixels with the most similar local neighbourhood as the synthetic texture. These methods are very useful for image completion. They can be constrained, as in "Image Analogies", to perform many interesting tasks. They are typically accelerated with some form of Approximate Nearest Neighbour method since the exhaustive search for the best pixel is somewhat slow. Synthesization: Too slow when the synthesized image is large. Time: Finding the neighbour so more time taken and also control over individual pixel value. Execution: Execution is slower. Seams: Here seams are not present. In this approach employs a pixel-based multi-resolution texture synthesis algorithm, which is based on a non-parametric sampling method. In it assumes a Markov random field texture model, which means a pixel value at a certain location only depends on its immediate neighbourhood.

4. Conclusion

There have been outstanding advancements in terms of quality, efficiency for synthesizing textures from image samples. In this paper we have shown the different Texture Synthesis methods. This dissertation addresses two parts. First part addresses brief introduction about texture, texture synthesis, and various types of Texture Synthesis and finally different application of Texture Synthesis. While second part of dissertation addresses methods for cleaning gray scale image, these two methods are Pixel Based Texture Synthesis and Patch Based Texture Synthesis.

5. Acknowledgement

The author is thankful for providing facilities during the work. Author is indebted by the useful suggestion and constant motivation and help received from his colleagues for the preparation and improvement of manuscript. Author also expresses his country for the images and information from respective websites:
1) www.google.com
2) www.generation5.org
3) www.wikipedia.org
4) IEEE Journals

6. References

[1] Yingqing Xu Baining Guo Lin Liang, Ce Liu and Heung-Yeung Shum. Real-time texture synthesis by patch based sampling. Technical Report, (MSR-TR-2001-40), March 2001

[2] Li-Yi Wei Marc Levoy. Fast texture synthesis using tree-structured vector quantization. Stanford University.

[3] Li-Yi Wei. Texture Synthesis By Fixed Neighborhood Searching. PhD thesis, STANFORD UNIVERSITY, November 2001.
[4] LI-YI WEI. Deterministic texture analysis and synthesis using tree structure vector quantization. Gates Computer Science Building, Stanford University, CA 94309, U.S.A., (386).
[5] W. Guo Y. Meng, W.H. Li and Y.L. Liu. Particle swarm optimization method used in pixel-based texture synthesis.
[6] Aaron Bobick Nipun Kwatra Vivek Kwatra, Irfan Essa. Texture optimization for example-based synthesis.
[7] Pizzanu Kanongchaiyosy Jakrapong Narkdej. An efficient parameters estimation method for automatic patch based texture synthesis.
[8] Ankit Shah, Parth Bhatt, Prof. Kirit J. Modi. Image Enhancement Techniques by Texture Synthesis, Volume 2, Issue 1, January 2012
.