

Review on Recent Patents in Texture Synthesis

Weiming Dong

LIAMA-NLPR, CAS Institute of Automation, China

Jean-Claude Paul

INRIA, France / Tsinghua University, China

Abstract

Computer graphics applications often use textures to render synthetic images. These textures can be obtained from a variety of sources such as hand-drawn pictures or scanned photographs. Texture synthesis is an alternative way to create textures. Because synthetic textures can be made any size, visual repetition is avoided. The goal of texture synthesis can be stated as follows: given a texture example, synthesize a new texture that, when perceived by a human observer, appears to be generated by the same underlying process. This paper reviews the recent patents on texture synthesis schemes. The key components in a texture synthesis algorithm, such as neighborhood matching, block sampling, anisometric synthesis, etc., are discussed. Then we discuss the applications of texture synthesis in texture magnification and image repairing. This paper also points out future works on this issue.

Keywords: Texture synthesis, neighborhood matching, block sampling, anisometric synthesis, texture magnification, image repairing

1 INTRODUCTION

Reproducing detailed surface appearance is important to achieve visual realism in computer rendered images. One way to model surface details is to use polygons or other geometric primitives. However, as details becomes finer and more complicated, explicit modeling with geometric primitives becomes less practical. An alternative is 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.

Generating novel photo-realistic imagery from smaller

examples has been widely recognized as a significant problem in computer graphics. A wide number of applications require realistic textures to be synthesized for object decoration in virtual scenes. Texture refers to the class of imagery which is usually defined as an infinite pattern consisting of stochastically stationary repeating elements. The global repeatability within texture images is essential to texture synthesis techniques. This inherent property also makes it possible to express adequate texture information with limited portions.

Texture synthesis is an efficient way to create textures because synthetic textures can be made any size, visual repetition is avoided. Texture synthesis can also produce tileable images by properly handling the boundary conditions. The objective of texture synthesis can be stated as follows. Given an example texture (Figure 1), synthesize a new texture that, when perceived by a human observer, appears to be generated by the same underlying process (Figure 1(b)).

In this paper, we will first discuss the various example-based texture synthesis algorithms. Then we review the recent authorized patents in this area and the texture synthesis-based applications such as style transfer, texture magnification and image repairing.

2 SYNTHESIS TEXTURES FROM EXAMPLES

A number of work has been presented toward synthesizing textures from input examples. Local region-growing techniques generate textures one pixel or one patch at a time. Pixel-based synthesis algorithms [2, 10, 29][1] grow an output texture pixel by pixel, normally using spatial neighborhood compare to match across different frequency bands. These approaches are fit for stochastic textures, but usually fail on textures with more coherent structures. Patch-based methods [9, 16, 30, 24] copy selected source regions into the output instead of single pixels. They are generally more successful on synthesizing structural textures. Some intermediate techniques [25, 15]



Figure 1: Texture synthesis. Given an example texture (a), the goal is to synthesize a new texture that looks like the input (b). The synthesized texture can be arbitrary size specified by the user.

between pixel and patch based methods have also been presented, which somewhat combine the advantages of both. None of the techniques above can avoid laborious neighborhood matching in the input example and this time-consuming process limits their use to only off-line synthesis applications. Lefebvre and Hoppe [19] defines an infinite, deterministic, aperiodic texture, from which windows can be computed in real-time on a GPU. This method attains high-quality synthesis using a novel analysis structure called the Gaussian stack, together with a coordinate unsampling step and a subpass correction approach. They show that quality is greatly improved if pointwise colors are replaced by appearance vectors that incorporate nonlocal information such as feature and radiance-transfer data [20]. They perform dimensionality reduction on these vectors prior to synthesis, to create an appearance space exemplar. The drawback of the GPU-based methods is that they always demand a high performance graphics hardware, and their methods suffer from the pixel-based synthesis issue of performing poorly on textures with semantic structures not captured by small neighborhoods. On the other hand, some near real-time texture synthesis methods have also been presented. Unfortunately, they usually achieve low quality results for the lack of optimization in the pre-processing [31, 32] or need very complex pre-computation and storing data structures [23]. They are not available in many real-time environments.

An alternative approach is to use texture synthesis to pre-compute a set of small tiles (textures) and use these tiles to generate arbitrary size of non-periodic images at run time. The tile-based method usually employs a set of sample patches which are extracted from the input example as texturing primitive. Tiles are then constructed by

stitching sample patches together following some given rules. The technique requires only a small amount of memory and is very useful in many real-time applications, although sometimes achieving low-quality results or dull patterns for the lack of optimization on the tile set. Cohen et al. [4] develop a stochastic algorithm to non-periodically tile the plane with a small set of Wang tiles at run time. Wei [28] extends this work with GPU to improve tile-based texture mapping. Dong et al. [5] improve the synthesis quality of Wang tiles by using Genetic Algorithm for sample patches selection. Zhang and Kim [35] present s-Wang tiles, which are a stricter interpretation of the original Wang tile design. Alternative for Wang tiles are also developed. Ng et al. [26] present an approach to generate a set of small texture tiles which are called ω -tiles from an input example. These tiles can also be tiled together to synthesize large textures. This method is extended on both patterns and qualities in [6]. Lagae and Dutré [18] propose corner tiles which are square tiles with colored corners. Corner tiles are similar to Wang tiles, but their colored corners ensure continuity of the signal over both tile edges and corners, thus avoiding the corner problem. For some applications, Lagae and Dutré [17] present a procedural object distribution function, a new texture basis function that distributes procedurally generated objects over a procedurally generated texture. Dong et al. [8] present an interactive texture design scheme based on the tile optimization and image composition.

For some special textures, novel techniques have also been presented. Zhang et al. [34] present an approach for decorating surfaces with progressively-variant textures. Liu et al. [24] present techniques to analyze and synthesize near-regular textures through multi-model deformation fields. Eisenacher et al. [11] combine a simple user

interface with a generic per-pixel synthesis algorithm to achieve high-quality synthesis from a photography. Dong et al. [7] present a novel texture synthesis scheme for anisotropic 2D textures based on perspective feature analysis and energy optimization.

3 PATENTS

As the field of example-based texture synthesis is relatively new, there are not many patents only describing inventions of texture synthesis. Therefore, this review also includes inventions or methodologies that involve texture synthesis-based applications that can be applied to the topic of review. The subsections that follow discuss the most relevant patents.

3.1 Texture synthesis and transfer for pixel images

Patent US6919903 [12] claims an image-based method for generating novel visual appearance in a new image. Synthetic texture is stitching together from small patches in existing images. First, the algorithm uses a least cost path determination to determine the local boundaries between the patches. Second, texture transfer is performed by rendering an arbitrary object with a synthetic texture taken from a different object. More generally, they provide methods for rendering entire images in styles of different images. The method works directly on pixel images, and does not require 3D information. More specifically, the invention provides a method for generating texture for a pixel image. The pixel image is partitioned into a plurality of blocks in a predetermined order, and for each block further processed by the following steps. A set of texture blocks is searched for a next block that best matches an adjacent previous block in a region where the previous and next block overlap. A least cost path is determined in the overlap region, and the next block is inserted adjacent to the previous block according to the least cost path to synthesize texture.

3.2 Block sampling based method and apparatus for texture synthesis

Patent US7023447 [3] claims a multi-resolution block sampling based texture analysis/synthesis algorithm. A reference texture is assumed to be sample from a probability function. The synthesis of a similar, but distinctive, synthetic texture is handled in a process and by an apparatus that first estimates and then resamples the probability function. In order to achieve good and fast estimation of the probability function for a reference tex-

ture and in order to retain the texel structural information during the synthesis, a concept of block sampling and a corresponding texture synthesis scheme based on multi-resolution block sampling is employed. As a result of this approach, the computational complexity of the present invention is much lower than that of other approaches to the problem. In addition, for textures that exhibit a high degree of directionality, a process, which integrates estimation of dominant texture direction and the synthesis algorithm is employed to handle directional textures. The dominant direction is used to orient and then control the synthesis process so as to preserve the dominant reference image direction.

3.3 System and method for real-time texture synthesis using patch-based sampling

Patent US6762769 [13] claims a system and method for synthesizing textures from an input example. A system and method according to the present invention uses a unique accelerated patch-based sampling system to synthesize high-quality textures in real-time using a small input texture example. The patch-based sampling system of the present invention works well for a wide variety of textures ranging from regular to stochastic. Potential feature mismatches across patch boundaries are avoided by sampling patches according to a non-parametric estimation of the local conditional Markov Random Field (MRF) density function.

3.4 System and method for performing texture synthesis

Patent US7136072 [27] claims a method for synthesizing a texture of a desired size from an example texture. The method comprises the steps of generating a matrix of the desired size, and providing values to the matrix. The values include random values and at least a portion of the values represents a desired structure according to which graphical features of a synthesized texture are to substantially conform. The method further comprises executing a texture synthesis process that utilizes the matrix to generate a synthesized texture of the desired size having graphical features arranged therein substantially in conformance with the desired structure.

3.5 Synthesis of progressively-variant textures and application to arbitrary surfaces

Patent US7161601 [33] claims methods for synthesizing progressively-variant textures based on texon masks are

provided. A first method creates a synthesized texture image guided by a sample texture, first texon mask corresponding to the sample texture and a second texon mask modified based on the first texon mask. A second method also creates a synthesized texture image guided by a first and second sample textures and corresponding first and second texon masks. A method for rendering a synthesized texture on an image of a three-dimensional object includes creating a synthesized texture on the object guided by a two-dimensional progressively-variant sample texture, a texon mask for the example texture and a mesh of a plurality of vertices representing the object.

3.6 Parallel texture synthesis having controllable jitter

Patent US20070002069 [14] claims a method and system for synthesizing texture using upsampled pixel coordinates and a multi-resolution approach. The parallel texture synthesis technique, while based on a neighborhood matching technique having order-independent texture synthesis, extends that approach in at least two areas, including efficient parallel synthesis and intuitive user control. Pixel coordinates are upsampled instead of pixel colors, thereby reducing computational complexity and expense. These upsampled pixel coordinates then are jittered to provide texture variation. The jitter is controllable, such that a user has control over several aspects of the jitter. In addition, each neighborhood matching pass is split into several sub-passes to improve correction. Using sub-passes improves correction speed and quality. The parallel texture synthesis system and method disclosed herein is designed for implementation on a parallel processor, such as a graphics processing unit.

3.7 Anisometric texture synthesis

Patent US2007011379 [22] claims an anisometric texture synthesis system and method for generating anisometric textures having a similar visual appearance as a given exemplar, but with varying orientation and scale. This variation is achieved by modifying the upsampling and correction processes of the texture synthesis technique using a Jacobian field. The modified correction process includes accessing only immediate neighbors of a pixel instead of non-local pixels. This constraint that only immediate neighbors be used also allows the generation of seamless anisometric surface textures. This is achieved by using indirection maps containing indirection pointers that are used to jump from a set of pixels outside the boundary of a texture atlas chart to another chart. The system and method also includes an anisometric synthesis magnification technique that uses a Jacobian field to

modify the magnification step of a synthesis magnification scheme and account for anisometry.

3.8 Magnification of indirection textures

Patent US20070002067 [21] claims an indirection texture magnification system and method for producing high-resolution indirection texture results. The system and method uses an indirection texture, designed for use with a low-resolution texture image, and a high-resolution texture image, which is a higher-resolution version of the low-resolution texture image. The indirection texture magnification system and method re-interprets an indirection texture computed for a low-resolution image so that a higher-resolution image can be used with the same indirection texture. This generates additional samples and allows the generation of a magnified, high-resolution indirection texture result. The indirection texture magnification system and method takes three or more neighboring pixel coordinates stored in an indirection texture and offsets those pixel coordinates in order to access the higher-resolution image with an increased precision. Colors obtained from adding this offset are combined to obtain a color for a corresponding pixel at a certain location in the magnified, high-resolution indirection texture result.

3.9 Texture synthesis for repairing damaged images

Patent US7012624 [36] claims a method for generating texture includes (1) selecting a target patch to be filled in an image, (2) selecting a sample patch as a candidate for filling the target patch, (3) determining a first difference between a first area surrounding the target patch and a corresponding first area surrounding the sample patch, and a second difference between a second area surrounding the target patch and a corresponding second area surrounding the sample patch, (4) multiplying a larger of the first difference and the second difference with a first weight factor, and a smaller of the first difference and the second difference with a second weight factor, and (5) summing the weighted first difference and the weighted second difference as a distance between the target patch and the sample patch.

4 CURRENT & FUTURE DEVELOPMENTS

It is clear from the articles and patents reviewed above that currently the area of example-based texture synthesis has already been well studied by many researchers. The neighborhood matching method is widely used for either

local or global search in the input example. While there are inventions that attempt to acquire some inner information of the texture example to improve the synthesis quality. It is important for researchers and developers to realize that visual features of the textures can play a very significant role in enhancing the quality of the synthesis results.

In the meantime, we look at the patents which utilize texture synthesis for other applications. The strength of such techniques lies in the exact selection of the examples and the adaptive synthesis processes.

In the near future, we think that techniques which are designed with the main purpose of increasing the synthesis quality of texture examples with special visual effects will be developed. For example, synthesizing highly structural texture examples is still a challenging problem. Also, methodologies developed for this field can be easily extended for many applications, such as image inpainting, artistic style transfer, image tiling, and so on. It is envisioned that in the near future, we can use textures in various areas.

Acknowledgements

This work is supported by China Postdoctoral Science Foundation funded project.

References

- [1] M. Ashikhmin, "Synthesizing natural textures," in *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*. New York, NY, USA: ACM Press, 2001, pp. 217–226.
- [2] J. S. D. Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 361–368.
- [3] C. W. Chen, J. Luo, and Y. Yu, *US7023447*, 2006.
- [4] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen, "Wang tiles for image and texture generation," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 287–294, 2003.
- [5] W. Dong, S. Sun, and J.-C. Paul, "Optimal sample patches selection for tile-based texture synthesis," in *CAD-CG '05: Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 503–508.
- [6] W. Dong, N. Zhou, and J.-C. Paul, "Optimized tile-based texture synthesis," in *GI '07: Proceedings of Graphics Interface 2007*. New York, NY, USA: ACM, 2007, pp. 249–256.
- [7] —, "Perspective-aware texture analysis and synthesis," *Vis. Comput.*, vol. 24, no. 7, pp. 515–523, 2008.
- [8] —, "Tile-based interactive texture design," *Lecture Notes in Computer Science*, vol. 5093, pp. 675–686, 2008.
- [9] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 2001, pp. 341–346.
- [10] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*. Washington, DC, USA: IEEE Computer Society, 1999, p. 1033.
- [11] C. Eisenacher, S. Lefebvre, and M. Stamminger, "Texture synthesis from photographs," *Computer Graphics Forum*, vol. 27, no. 2, pp. 419–428, 2008.
- [12] W. T. Freeman and A. Efros, *US6919903*, 2001.
- [13] B. Guo, L. Liang, H.-Y. Shum, and Y.-Q. Xu, *US6762769*, 2002.
- [14] H. H. Hoppe and S. Lefebvre, *US20070002069*, 2007.
- [15] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 795–802, 2005.
- [16] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, 2003.
- [17] A. Lagae and P. Dutré, "A procedural object distribution function," *ACM Trans. Graph.*, vol. 24, no. 4, pp. 1442–1461, 2005.
- [18] A. Lagae and P. Dutré, "An alternative for wang tiles: colored edges versus colored corners," *ACM Trans. Graph.*, vol. 25, no. 4, pp. 1442–1459, 2006.
- [19] S. Lefebvre and H. Hoppe, "Parallel controllable texture synthesis," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 777–786, 2005.

- [20] —, “Appearance-space texture synthesis,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 541–548, 2006.
- [21] S. Lefebvre and H. H. Hoppe, *US20070002067*, 2007.
- [22] —, *US2007011379*, 2007.
- [23] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, “Real-time texture synthesis by patch-based sampling,” *ACM Trans. Graph.*, vol. 20, no. 3, pp. 127–150, 2001.
- [24] Y. Liu, W.-C. Lin, and J. Hays, “Near-regular texture analysis and manipulation,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 368–376, 2004.
- [25] A. Nealen and M. Alexa, “Hybrid texture synthesis,” in *EGRW ’03: Proceedings of the 14th Eurographics workshop on Rendering*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, pp. 97–105.
- [26] T.-Y. Ng, C. Wen, T.-S. Tan, X. Zhang, and Y. J. Kim, “Generating an ω -tile set for texture synthesis,” in *Proceedings of Computer Graphics International 2005 (CGI’05)*, Stone Brook, NY, USA, 2005, pp. 177–184.
- [27] B. A. Ritter, *US7136072*, 2006.
- [28] L.-Y. Wei, “Tile-based texture mapping on graphics hardware,” in *HWWS ’04: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*. New York, NY, USA: ACM Press, 2004, pp. 55–63.
- [29] L.-Y. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” in *SIGGRAPH ’00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 479–488.
- [30] Q. Wu and Y. Yu, “Feature matching and deformation for texture synthesis,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 364–367, 2004.
- [31] S. Zelinka and M. Garland, “Towards real-time texture synthesis with the jump map,” in *EGRW ’02: Proceedings of the 13th Eurographics workshop on Rendering*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2002, pp. 99–104.
- [32] —, “Jump map-based interactive texture synthesis,” *ACM Trans. Graph.*, vol. 23, no. 4, pp. 930–962, 2004.
- [33] J. Zhang, K. Zhou, B. Guo, and H.-Y. Shum, *US7161601*, 2007.
- [34] J. Zhang, K. Zhou, L. Velho, B. Guo, and H.-Y. Shum, “Synthesis of progressively-variant textures on arbitrary surfaces,” in *SIGGRAPH ’03: ACM SIGGRAPH 2003 Papers*. New York, NY, USA: ACM, 2003, pp. 295–302.
- [35] X. Zhang and Y. J. Kim, “Efficient texture synthesis using strict wang tiles,” *Graph. Models*, vol. 70, no. 3, pp. 43–56, 2008.
- [36] Y. Zhu, H. Wang, Y. Jin, D. Wu, and L. Zhou, *US7012624*, 2006.