Synthesizing Transition Textures on Succession Patterns

Yueh-Yi Lai, Wen-Kai Tai*
Department of Computer Science and Information Engineering
National Dong Hwa University

Chin-Chen Chang†
Department of Computer Science and Information Engineering
National United University

Chen-Duo Liu‡
Department of Information Management
Shih Chien University

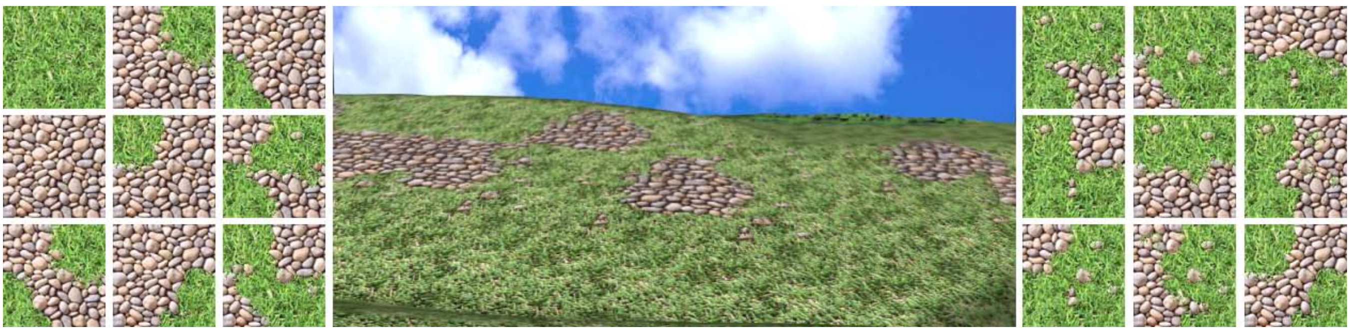


Figure 1: Sample transition textures and a terrain with transition textures.

Abstract

Synthesizing transition textures on succession patterns for displaying visually acceptable terrain is essential to applications such as computer games. In this paper, we exploited the Game of Life model to simulate successions on terrain to show that a good-looking profile can appear quickly and easily. Based on the proposed modified patch-based sampling texture synthesis approach, the basic types of transition textures on a succession pattern can be well synthesized. This method first generates a well-shaped transition cut along the source texture. Next, the feature maps of the source texture and the target texture are used to determine complementary patches from the target texture that will fit well along the source's transition cut. As the experimental results show, only a few input textures are required by our approach to synthesize plenty of tileable transition textures which are useful for obtaining vivid terrain while consuming only a small amount of texture memory.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

Keywords: texture synthesis, texture transition

1 Introduction

*e-mail: wktai@mail.ndhu.edu.tw
†e-mail: ccchang@nuu.edu.tw
‡e-mail: aduo@mail.usc.edu.tw

The terrain editing system is an essential tool in many games. Non-repeating tileable texture patterns are required for these systems to display visually acceptable pictures. Basically, the appearance of a terrain is complex and composed of numerous compound textures. In this paper, we exploit the Game of Life [Conway 1970] model to simulate successions on a terrain. The appearance of a terrain consists of many types of succession patterns. Users are allowed to assign some cells of a terrain as a new born species and apply the cell state update mechanism of the Game of Life model to get preferred succession patterns. While initially editing a terrain's appearance, its profile outlined by this way helps user have a good-look profile on a terrain quickly and easily. We have remarked that there are sixteen transition types in the succession pattern. A modified patch-based sampling texture synthesis approach is proposed to synthesize visually acceptable transition textures. Given two input textures for creating a transition between them, we decide the transition cut for the output transition texture first. Second, our approach uses both color intensity from input textures and feature attribute from respective feature map to evaluate the patch matching. As the experimental results show, variant transition cuts create a number of varied transition textures in the same transition pattern and synthesized transition textures present good visual effects. In short, our approach uses few input textures to synthesize a variety of transition textures and makes the appearance of a terrain more vivid (Figure 1).

Since the two input textures are too dissimilar, the blurred outcome and unconstrained transition path are two major problems to synthesizing a transition texture by traditional texture synthesis methods. We will address these problems and present the following contributions:

- We introduce a novel constrainable path searching method , and hence the transition texture we synthesized is variant and tileable while keeping features of input textures.
- We introduce the idea of the Game of Life to generate a good-look profile for an ecosystem, which offers users a friendly interface for representing the profile of a terrain.

2 Related Works

In recent years, a great deal of work on texture synthesis has been proposed in the field of computer graphics and vision. Here we simply review some up-to-date and typical approaches.

The texture model, Markov Random Field, is well-known as a good strategy to generate a wide variety of textures using probability sampling [Ashikhmin 2001; Efros and Leung 1999; Liang et al. 2001; Wei and Levoy 2000]. Hertzmann et al. [2001] combined the advantages of [Wei and Levoy 2000] and Ashikhmin's [2001] searching algorithm for texture synthesis. The synthesized textures have a high quality as those produced by Ashikhmin's but still produces some undesirable artifacts. Ziv et al. [2001] proposed an approach that considered texture images as examples of approximately stationary 2D signals. And it could merge several different signals statistically to form a new mixing texture. Zhang et al. [2003] proposed feature-based warping and blending mechanisms to control the texture elements, and provided a smooth transition. Image quilting [Efros and Freeman 2001] similar to patch-based sampling scheme [Liang et al. 2001] generates synthesized images by stitching together small patches of existing images. Graphcut [Kwatra et al. 2003] synthesized a new texture by copying irregularly shaped patches from the sample image into the output image. The portion of the patch to copy is determined by using a graph cut algorithm. The graph cut is a powerful method to combine textures that are similar to each other, but it could not be applied to a pair of dissimilar textures. In this paper, we propose a method that synthesizes a transition texture from two input textures and conform to the natural phenomena of ecology.

3 Succession Pattern

A mathematical model, Game of Life [Conway 1970], can be exploited to simulate the succession pattern which provides a quick profile of a portion of terrain's appearance. Given a new species, the mechanism of Game of Life simulates a succession. Without loss of generality, a planar grid consisting of cells can be used to represent the surface of a terrain. Initially, the user selects a number of cells where new species are to be located. At each iteration of succession, a species in a cell may be died (death state), still alive (survival state), or a new born (birth state). The number of occupied cells in eight neighbors, adjacent cells, of a cell are used to determine its state. The cell, an occupied cell, is in a death state if it is of loneliness, zero or only one of it neighbors is occupied by the same species, or over crowded, over half of its neighbors are occupied. This occupied cell may still in a survival state provided that just two or three of its neighbors are occupied. If an unoccupied cell has exactly three occupied neighbors, it turns to being occupied. That means a new born species arrived.

While simulation a succession on a terrain, the user seeded some cells as a new species that invade to the original ecological system, and one can stop and select the succession pattern at any kth iteration. When we select a succession pattern to represents a profile of a portion of terrain, the textures of species should be mapped to the cells of the succession pattern to obtain a good-look terrain appearance. So, many tileable transition textures are required for adjacent cells with different textures. We identify all transition patterns by shifting all cells such that the cell is centered at the cross section of grid lines as shown in Figure 2. From these succession patterns, there are 16 fundamental transition patterns could be summarized as shown in Figure 3.

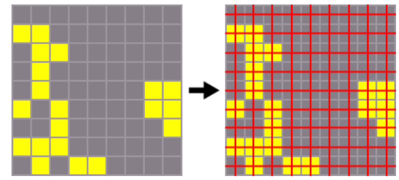


Figure 2: Left is a selected succession pattern. Right is the same with left one but all cells are shifted for identifying all transition patterns.

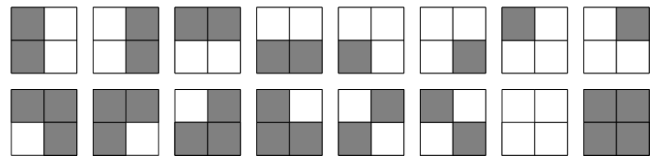


Figure 3: 16 fundamental types of transition patterns.

4 Transition Texture Synthesis

Our approach for synthesizing transition texture is similar to the texture synthesis approach in that the differences are: (1) The synthesis process takes two input textures, namely source and target textures. (2) In addition to the input texture, its feature map is used to enhance the output result while synthesizing. (3) The output texture is initialized to one of 16 fundamental transition patterns, and the resultant output texture contains two inputs, presenting a transition visual effect.

4.1 Feature Map

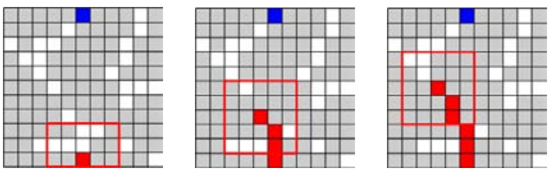
The feature map of an input texture provides high frequency information, i.e., edges or easy-to-notice line segments. We use the feature map to describe the edges of texture elements in a texture. Synthesizing texture based upon the color intensity and this feature leads to better results for transition texture. Inspired by [Wu and Yu 2004], to generate the feature map for a texture, we first apply bilateral filtering [Smith and Brady 1995] [Tomasi and Manduchi 1998] to sharpen the edges. Second, we use finite differences as a simple gradient estimator to obtain an edge response of every pixel. Since the detected features may have a multi-pixel width and some noises, we further apply a modified thinning algorithm to removed pixels redundant. Figure 5(a) and (b) show an example texture and its corresponding feature map.

4.2 Output Texture Initialization

Ahead of proceeding transition texture synthesis, the output texture is initialized as one of 16 fundamental patterns. This can be done by cutting a portion of the source texture and putting it at the output as a fundamental pattern. To cut a portion, a transition cut must be determined in the feature map of the source texture. Let the feature map be a graph $G = < V, E >$ where each feature pixel is a vertex of V and an edge of E is associated for all pairs of vertices which are adjacent. The transition cut is a connected path so that it creates a corresponding closed portion of the source texture. We propose a constraint-based approach to deal with the path finding problem in question. When the starting vertex is given in the feature map, the next vertex is to be found within the active neighborhood $N_{A,s}$, with size $r \times r$ pixels, of starting vertex v_s. If vertex v_i in $N_{A,s}$ is with the shortest Euclidean distance from v_i to v_e (the end vertex), the edge $v_s v_i$ is appended to the path, transition cut. Figure 4 is an example of the path searching. The path can be recursively stated as

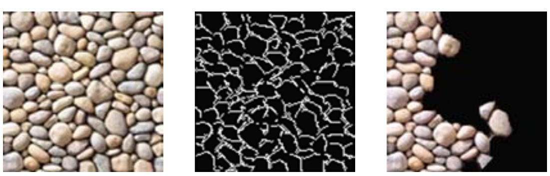
$$P_k\{v_s, v_1, \ldots, v_k\} = P_{k-1}\{v_s, v_1, \ldots, v_{k-1}\} \\ \cup \{v_k \mid k = \arg\min_l |v_l - v_e|, \forall v_l \in N_{A,k-1} \& v_l \notin P_{k-1}\}$$

where j is the current vertex which is the center of the active neighborhood. If the next vertex (v_i) is not connected with the current vertex (v_j), such as Figure 4(b), then we add some vertices from v_j to v_i by line drawing algorithm (Figure 4(c)). Whenever there are no any vertices in $N_{A,j}$, its size is adaptively enlarged to find the most likely vertex. An example of transition cut and its corresponding portion of source texture are shown in Figure 5(c).



(a) (b) (c)

Figure 4: An example of the path finding. (a) is the initial state where the start and end vertices for the search are marked, and the square is an active neighborhood, $N_{A,c}$. (b) and (c) are two states of the path finding process.



(a) (b) (c)

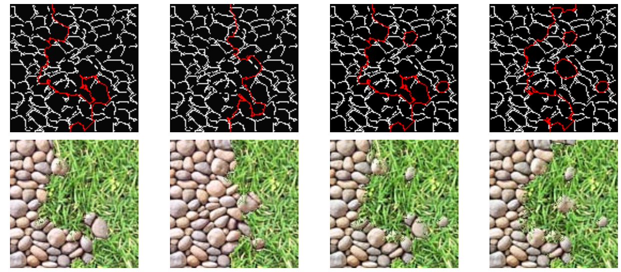
Figure 5: (a) The source texture and its feature map (b). (c) A transition cut found in (b) and its corresponding portion of source texture.

4.3 Transition Texture Synthesis

The patch based sampling approach [Liang et al. 2001] is used to synthesize transition texture after the output texture has been initialized. While searching the best matching patch in the target texture, the similarity between patches of target P^{target} and output P^{out} textures is measured as follows:

$$d(P_k^{target}, P_k^{out}) \\ = w_f \sum_{i=1}^{A} \left| E^i_{P_k^{target},f} - E^i_{P_k^{out},f} \right| + w_c \sum_{i=1}^{A} \left| E^i_{P_k^{target},c} - E^i_{P_k^{out},c} \right|$$

where $E_{P,f}$ indicates the corresponding boundary zone of patch P in the feature map and $E_{P,c}$ in the target texture, and i is the ith pixel of the boundary zone. Parameters w_f and w_c are used to weight the feature term and color intensity term respectively. Note that the overlapped boundary of the best matched patch P_B is not going to blend with the corresponding boundary zone already existed in the output texture. Instead, we place the portion of the best match, $\{P_B^{target} - E_{P_k^{out},c}\}$, in the output. However, when the boundary zone $E_{P_k^{out},c}$ doesn't contain textures from source texture,



(a) (b) (c) (d)

Figure 6: (a) and (b) are synthesized transition textures with the same transition type but different transition cuts. (c) and (d) are transition textures using multiple transition cuts.

we perform feathering method [Nealen and Alexa 2004] on the overlapped zone.

The transition texture must be tileable, namely each pair of connectable fundamental transition textures must be seamlessly tiled together. First of all, we exploit the hybrid-based texture synthesis approach [Wu et al. 2004] to make the input tileable. Let T_A and T_B be two transition textures and T_B be connectable to the right side of T_A. Before we proceed the patch-based sampling approach for synthesizing T_A, we introduce an extra initialization by appending a stripe of texture, s, with the width same as the boundary zone of a patch from the target input texture to the right most side of T_A. As the synthesizing process reaches the right most end of T_A, the best matching candidates are selected by evaluating not only the overlapped boundary but also a section of s with the same height as the patch. Going through this process for all candidate patches selection near the right most end of T_A promises that no seams will be noticed between the right side of T_A and the stripe s. Trivially, while initializing the left side of T_B using this stripe s, T_B would be seamlessly connectable to T_A. Similarly, applying the above mechanism to all transition textures makes them all tileable.

5 Experimental Results

We have carefully tested our approach on synthesizing transition textures. We choose two species texture, grass and Caryophyllaceae textures, and a landform texture, stone textures, to represent the terrain's appearance. There are three transitions and for each transition we synthesize sixteen transition textures. From Figures 7 through 9 display all tileable transition textures synthesized by our approach. For a transition type, variant transition cuts create varied texture patterns. Moreover, when allows more than one transition cuts the synthesized texture provides more convincing transition texture. Figures 6(a) and (b) shows the synthesized transition textures with the same transition type but different transition cuts, and (c) and (d) demonstrates more convincing transition textures while using multiple transition cuts. Finally, we demonstrate a synthesized terrain appearance in Figure 10 using the succession pattern and transition textures from Figures 7 through 9.

All experiments were performed on a 2.4GHz Intel processor with 1GB main memory. On average, without any speed-up efforts, it takes about 2 seconds to obtain a feature map from an input texture with resolutions 120×120, 0.02 seconds to find a transition cut, and 5 seconds to synthesize a transition textures with resolution 120×120. The tiles generation process is time consumed. But, once this is run, generating large amounts of terrain textures at runtime is relatively free both in texture memory and time.

6 Conclusion and Future Work

We have presented an approach for synthesizing transition textures on succession patterns. We exploite the bilateral filter to smooth out the input texture and sharpen the boundary of texture elements. Then we detecte the features using finite difference method to generate a feature map of input source texture. A novel path finding method is used to search a transition cut, and a portion of source texture enclosed in the cut is copied to the output texture as an initialization. Combined with feature map and color value, our synthesis method could synthesize a transition texture that has high quality and good-look visual effect. As experimental results show, when users select a succession pattern simulated by the Game of Life model for a terrain, its appearance with synthesized transition textures mapped on demonstrates visually acceptable results. In the future, we are interested in speed-up our method in order to work interactively in terrain editing system. Other concerned topics include transition methods for more than two textures and texture synthesis in motion.

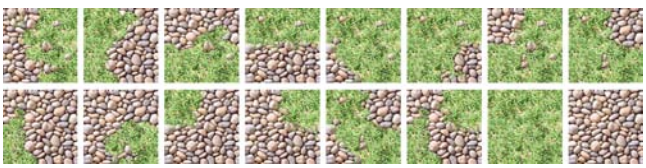


Figure 7: All synthesized transition textures from the Grass and Stone textures, last texture at the top and middle row respectively. The texture at the last row shows an example of succession pattern.

Figure 8: All synthesized transition textures from the Grass and Caryophyllaceae textures, last texture at the top and middle row respectively. The texture at the last row shows an example of succession pattern.

Figure 9: All synthesized transition textures from the Stone and Caryophyllaceae textures, last texture at the top and middle row respectively. The texture at the last row shows an example of succession pattern.

Figure 10: Succession patterns using 16 transition tiles mapped on a terrain.

References

Ashikhmin, M. 2001. Synthesizing natural textures. *ACM Symposium on Interactive 3D Graphics*, 217–226.

Conway, J. H. 1970. *Mathematical Games*. Scientific American.

Efros, A. A., and Freeman, W. 2001. Image quilting for texture synthesis and transfer. In *Proc. SIGGRAPH 2001*, 341–346.

Efros, A. A., and Leung, T. K. 1999. Texture synthesis by non-parametric sampling. In *Proc. of International Comference on Computer Vision*.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. 2001. Image analogies. In *Proc. SIGGRAPH 2001*, 327–340.

Kwatra, V., Schodl, A., Essa, I., Turk, G., and Bobick, A. 2003. Graphcut textures: image and video texture synthesis using graph cuts. In *Proc. SIGGRAPH 2003*, 277–286.

Liang, L., Liu, C., Xu, Y.-Q., Guo, B., and Shum, H.-Y. 2001. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (TOG) 20*, 3, 127–150.

Nealen, A., and Alexa, M. 2004. Fast and high quality overlap repair for patch-based texture synthesis. In *Proc. of the Computer Graphics International*, 582–585.

Rao, A. R. 1990. *A Taxonomy for Texture Description and Identification*. Springer-Verlag.

Smith, S. M., and Brady, J. M. 1995. Susan – a new approach to low level image processing. Tech. Rep. TR95SMS1c.

Tomasi, C., and Manduchi, R. 1998. Bilateral filtering for gray and color images. In *Proc. of International Comference on Computer Vision*, 836–846.

Wei, L. Y., and Levoy, M. 2000. Fast texture synthesis using treestructured vector quantization. In *Proc. SIGGRAPH 2000*, 479–488.

Wu, Q., and Yu, Y. 2004. Feature matching and deformation for texture synthesis. In *Proc. SIGGRAPH 2004*, 364–367.

Wu, C.-H., Lai, Y.-Y., and Tai, W.-K. 2004. A hybrid-based texture synthesis approach. *The Visual Computer 20*, 106–129.

Zhang, J., Zhou, K., Velho, L., Guo, B., and Shum, H. Y. 2003. Synthesis of progressively-variant textures on arbitrary surfaces. In *Proc. SIGGRAPH 2003*, 295–302.

Ziv, B. J., Ran, E. Y., Lischinski, D., and Werman, M. 2001. Texture mixing and texture movie synthesis using statistical learning. *IEEE Trans. on Visualization and Computer Graphics 7*, 2, 120–135.

Synthesizing Transition Textures on Succession Patterns
Lai, Tai, Chang and Liu

Adaptive Streaming and Rendering of Large Terrains using Strip Masks
Pouderoux and Marvie

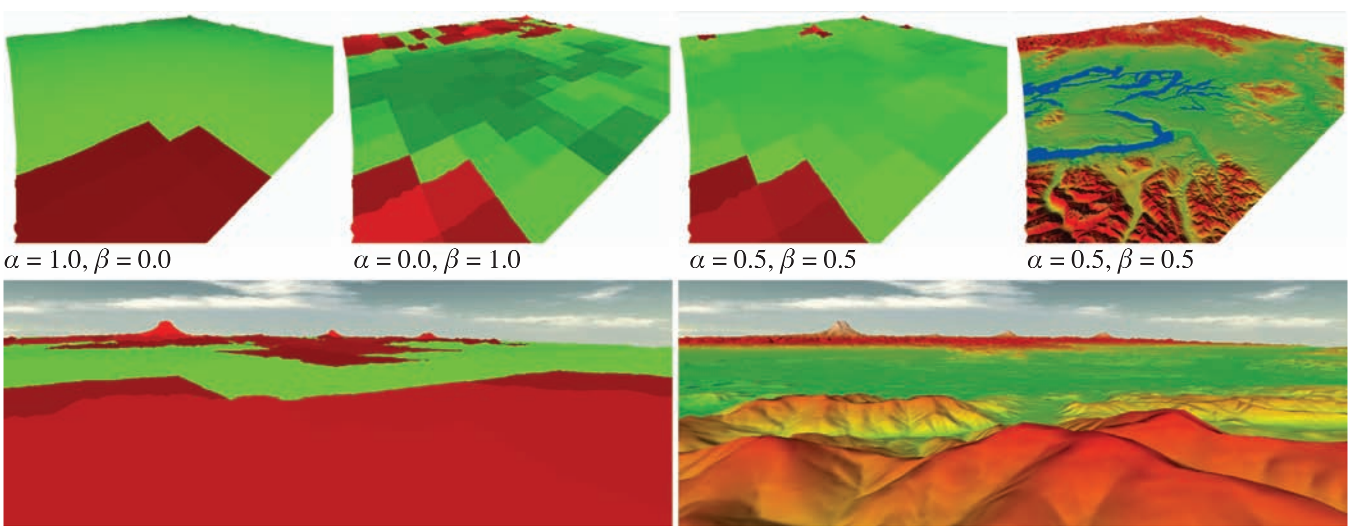


Figure 3: Visual importances on the Puget Sound model. Model elevations are exaggerated in order to see the relief better. Each tile is colored using the following color scale: red is more important than green, and light than dark. Top: from left to right, pictures show the importances using distance only, height only, and both ($\alpha = \beta = 0.5$). The last image shows the textured terrain. Bottom: The left picture depicts the visual importances using $\alpha = \beta = 0.5$ which is a good tradeoff. The right picture shows the texture-mapped result. Note how the far mountains are preserved so the horizon is meaningful.