Geometric Based Structure Propagation and Texture Matching for 3D Texture Completion

Kan Guo, Xiaowu Chen, Yili Liu, Bin Zhou*, Yu Guo
State Key Laboratory of Virtual Reality Technology and Systems
School of Computer Science and Engineering, Beihang University, Beijing, China
{*guokan, chen, liuyl, zhoubin, guoyu*}@*vrlab.buaa.edu.cn*

***Abstract*—This paper proposes a novel method to complete the missing appearance of mesh surfaces through structure propagation and texture matching. In advance, the appearance on surfaces is divided into structure and texture components. First, the missing structures are predicted through structure propagation. The structure propagation is guided by surfaces geometric features by using 3D tensor voting and crest line detection. Then, with the guidance of predicted structures, the optimal local sample patches are matched to inpaint the missing texture. Finally, the completed result is obtained by mapping the optimal patches to the missing texture regions on surface. We acquire convincing completion results as shown in the experiments and comparisons.**

***Keywords*-structure propagation, texture completion, tensor voting**

I. Introduction

A. Objective

Realistic texture models remain one of the most popular and ubiquitous forms for expressing the real object in our world, which has a wide variety of applications. Sometimes the texture on 3D model surface is not complete for many reasons, and even though many of the textures have significant structure, filling the so-called texture hole on 3D model surface is still a challenging problem in computer graphics and computer vision.

Compared with the problem of inpainting and texture completion of 2D images [1] [2], texture completion on 3D surface is more challenging for several reasons. Points on 3D model surface are float so that the point sampling is irregular and does not make up a regular parameter domain as the image. In addition, similarity comparison between the point sets is difficult to measure. Most of the previous work on surface texture completion concentrates on homogeneous textures. Homogeneous textures, however, could hardly behave the structure of textures on model surface. In this paper, we present an approach for texture completion on 3D model surface, which uses geometric based structure propagation. And as some special examples, we involve the prior knowledge to assist the propagation procedure.

*Corresponding author

B. Related work

Many researches apply texture synthesis methods to fill the 3D model surface, and most of which are done at vertex level, such as Turk [3] and Lefebvre et al. [4]. While others concentrate on patch level like Magda et al. [5]. Markov random field (MRF) models [6] are frequently used as in [7]. Generally, the methods algorithmically construct a large digital image from a small digital sample image by taking advantage of its structural content, such as orientation field or vector field, and map them onto the 3D model surface. Meanwhile, Ying et al. [8] and Zelinka et al. [9] synthesize the texture directly on the model surface. Local parameterization illustrated in [10] is also a common technique for generating textures on surfaces. Xu et al. [11] explore the use of salient curves in synthesizing intuitive, shape-revealing textures on surfaces by matching the direction of the texture patterns to those of the salient curves and aligning the prominent feature lines in the texture to the salient curves exactly. Even though the methods mentioned above could get homogeneous and seamless texture, they seem not put attention on the real world texture of the model. Walter et al. [12] integrate a biologically plausible pattern generation model, which can effectively deliver a variety of patterns characteristic of mammalian coats. Zhang et al. [13] present an approach for decorating surfaces with progressively variant textures. Intuitively, they could also synthesis a natural texture with patterns. However, in addition, they must define the texton masks manually in advance.

To our knowledge, little work has been done on defective texture completion while simultaneously consider the existing texture on 3D model surface. By using the radial basis functions, Zhou et al. [14] propose a Poisson-based method to make an efficient surface texture inpainting. Aim at three different situations, they provide three solutions, simple interpolation, user-guided inpainting, and seamless cloning, and could achieve convincing results. However, their methods require lots of human interaction, and without considering the structure information the inpainting result is excessively blurry. With a constrained texture synthesis algorithm and the surrounding existing texture, Xiao et al.

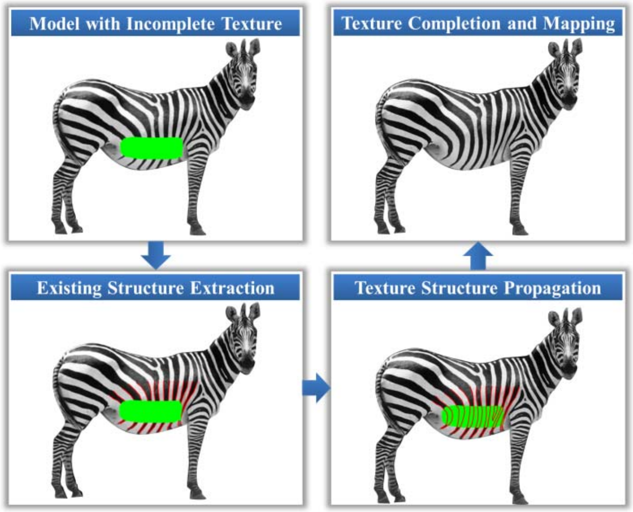

Figure 1. The brief workflow of our method. The green region on the 3D model surface is the missing texture region, shown at the top left corner. Firstly, the existing texture structure is extracted, which is represented by the red lines shown at the lower left corner. Secondly, guided by the existing structure and geometry features, the texture structure is propagated and shown at the lower right corner. Finally, the texture is completed based on the completed structure and also the geometric features, shown at the top right corner.

[15] raise a point-based texture completion method, which could get nice appearance on 3D surface. Yet the method needs the user to select a region on the existing surface as an input sample texture.

C. Our approach

To deal with the large texture hole region on 3D model surface, we divide the completion process into two main steps. The one is to complete the structure, and the other is to complete the texture in the hole region. In order to maintain the consistency of texture structure on the surface of 3D model, we firstly extract the existing structure on the model surface according to the texture gradient, and next we use tensor voting algorithm [16] to propagate the structure line in the hole while considering the surface geometry features. While completing the texture, we make the structure line texture completion come first so that the breaking of salient structures which human eyes are sensitive to could be reduced. Through the local projection we normally sample the texture patch on model surface. And similar to Sun et al. [17], we propagate the structure texture in three-dimensional space. Last we introduce the Markov chain Monte Carlo (MCMC) algorithm [18] for the remaining texture completion. And the central workflow of our method is shown in Fig.1. As we mentioned before, for special examples, we involve the prior knowledge, which considers the real natural texture, into the structure and texture completion.

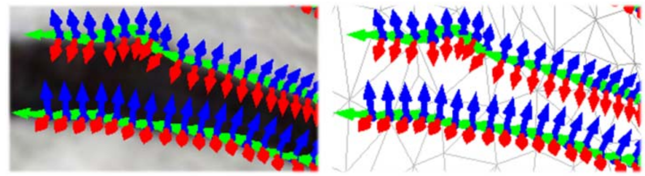

Figure 2. Initial results of sparse tensor voting. (a) face style; (b) line style. Each sample point inside the texture structure lines obtains the salient direction information, the green arrowhead denotes the tangent direction (salient direction), the blue one denotes the normal direction and the red one is the cross product direction of the tangent direction and normal direction.

Our main contributions include: (1) the process of texture completion is divided into two steps which are structure propagation and texture completion, and (2) the surface geometric features are involved into the process of the texture completion.

II. Structure propagation

A. Existing structure extraction

For structure propagation, we need firstly to find and extract the existing structure line. In our approach, we achieve that by detecting the point texture gradient on 3D surface. Primarily, with the scan line algorithm, we rasterize the 3D triangle faces into discrete points and so as to conveniently calculate the texture gradient. Due to the floating feature of 3D points, similar to Yuksel [19], we define a coarse resolution parameter which is used for the rasterization. Next, by setting down a gradient threshold, we can grab the points which have a bigger gradient value. Ultimately, after removing the noise points and selecting the points around the defective 3D surface, we successfully obtain the existing structure, as shown in Fig.3. As this process is somehow similar to an image segment problem, the result is hard to control and predict. Nevertheless, model texture is less complex than scene texture on the one hand, and on the other hand, as we can correct the deviation in the process of structure texture propagation, the existing structure extraction could have a tolerance and could be not completely precise.

B. Structure propagation without prior knowledge

Inspired by Jia et al. [20], we first propagate the structure line using the tensor voting algorithm proposed by Tang et al. [16], considering the geometry feature as a significant factor. Tensor voting is not only a very effective way to analyze the geometry information of points but also a robust method to address the problem of salient structure inference on 3D model surface.

There are three types of the 3D tensor, the stick tensor, plate tensor and ball tensor. Case by the unknown directions of the vertices inside the existing structure line, ball tensor is used for initializing. The initial ball tensors vote each other and turn out the generic second order symmetric tensors,

Figure 3. Structure propagation without prior knowledge. (a) existing structure extraction result shown as the red points and (b) structure propagation result exhibited by red points inside the texture hole.

Figure 4. Photos of real zebras and crestlines on the model surface. Clearly as we can see from (a) the real zebra photos, there is a black stripe in the middle of the zebras' stomach. And shown in (b) with the red lines, several crestlines detected from the model surface are exactly with the corresponding to it.

which encodes confidence of this knowledge (given by the tensor size), curve and surface orientation information (given by the tensor orientations), as shown in Fig.2. In a second stage, these generic tensors propagate their information in their neighborhood, leading to a dense tensor map which encodes feature saliency at every point in the domain. The second order symmetric tensor T is represented as

$$T = \lambda_1 \vec{e_1}\vec{e_1}^T + \lambda_2 \vec{e_2}\vec{e_2}^T + \lambda_3 \vec{e_3}\vec{e_3}^T \tag{1}$$

The vertices in the texture hole region are involved into the voting process. Then the vertices with top saliency are selected and carried out to the existing line. After the tensor re-initialization, the voting procedure iteratively repeats. To achieve more actual results and avoid the wrong connection, we particularly set some parameters defined below to terminate the voting process at appropriate time and finally get the whole texture structure as needed.

Considering the character of tensor voting and real world texture, we add two factors to restrict the tensor voting. The one is the distance between two structure lines. While the vote is under process, we calculate the distance between each two lines at each vote step, and decide whether two of those could combine or not by defined threshold α. Let $D(v,p)$ be the geodesic distance [21] between vertex v and vertex p, $D(v,L) = \min_{p \epsilon L} D(v,p)$ be the distance between vertex v and line L, and define $D(L_i, L_j) = \min_{v \epsilon L_i} D(v, L_j)$ as the distance between line L_i and line L_j. If $D(L_i, L_j) < \alpha$, we call that the two lines are distance satisfied. Totally, if two structure lines are distance satisfied, we consider that they are attachable and connect them into one completed line. Otherwise we continue the vote step and update the vote result. Simultaneously we also set a count threshold β of each line to limit the vote iterative steps. And the final completed result is shown in Fig. 3.

C. Structure propagation with prior knowledge

For most 3D texture completion or synthesis methods only concentrate on the smoothness, though their results can achieve satisfied appearance, they do not consider that whether the texture accord with the real world texture or not. We propose a new method which take the true natural texture as prior knowledge into account, and use the geometry feature on 3D model surface, which corresponds to the prior knowledge, to guide the texture propagation procedure.

For example, as shown in Fig.4, in nature there is a black stripe in the zebra's stomach. By analysis of the zebra model and experiments, we find that one of the crest lines extracted by Yoshizawa's method [22] is just identical to the stripe in zebra's stomach. And after some necessary processes, we obtain a crest line and two assistant structure lines around, which accords with the true natural texture, to guide the structure propagation.

Similar to the structure propagation without prior knowledge illustrated above, we add the assistant structure lines which correspond to the prior knowledge into the voting process, as shown in Fig.5, and the only difference compared to the previous is that we regard the prior structure lines as greatest impact lines when choosing the high saliency vertices and make all the other structure lines connect to the prior lines. And the final structure line completion result is shown in Fig.5 as well.

III. Texture completion

A. Structure texture propagation

After the structure lines fixed, based on the sparsely sample points, we specify the sample rectangles on 3D surface. As shown in Figs. 6, for one sample point p_i, the vector f_i and b_i denote the directions of p_i front and back, vector n_i stand for the normal of p_i, and we denote $d_i = 1/2 * (f_i + b_i)$ as the direction of p_i, $v_i = d_i x n_i$ as another direction of p_i. By this way, a local frame v_i, d_i, n_i is then established. With a specified radius r next a local rectangle patch is built on p_i. Afterwards we project the vertices nearby and detect whether they belong to this patch. Corresponding to the texture atlas [23], which store the

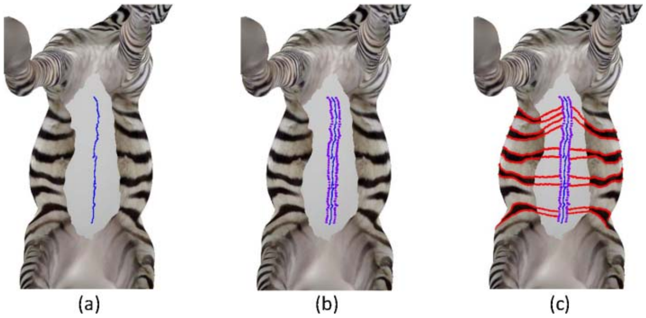

Figure 5. Structure propagation with prior knowledge. (a) the processed crest line displayed with the blue points; (b) two assistant lines around shown by the purple points; (c) the final structure propagation result exhibited by the red points.

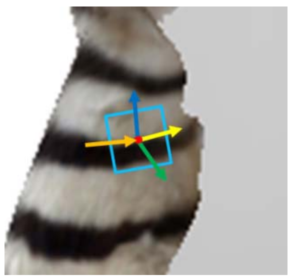

Figure 6. Local parameterization at the sample vertex. The yellow and orange arrows respectively denote the front and back directions of the sample vertex, which is shown as the red point. While the green arrow denotes the normal and the blue one represents the cross product of the normal and the average direction. And the light blue rectangle denotes the final sample patch.

texture of the model, we finally fill in the sample patches by using a triangle rasterizing algorithm. Thus, we build regular 2D texture samples for irregularly 3D points and triangle faces and make the 3D texture completion problem similar to 2D cases. In most cases, as the point set is densely sampled, the sample patches are quite flat and so the distortion of the local projection is quite small. According to the requirement, by controlling the sample radius r and sparsely sampling, we force the neighboring sample patches overlap to make it less computationally expensive for computing the energy.

The propagated structure texture should keep consistent with the surrounding texture and the boundary between the completed and existing regions should be continuous. Here we use the belief propagation algorithm similar to Sun et al. [17]. As a local message passing algorithm, belief propagation can minimize the Gibbs energy. Each patch on the model surface should receive messages from its neighbors and send updated messages back to each other. And the core iterative message update procedure is restricted by the energy we define below.

$$E(P) = \sum_{P_k \epsilon \Omega_B} E(P_k) + \sum_{P_i \epsilon \Omega} \sum_{P_j \epsilon N(P_i)} E(P_i, P_j) \quad (2)$$

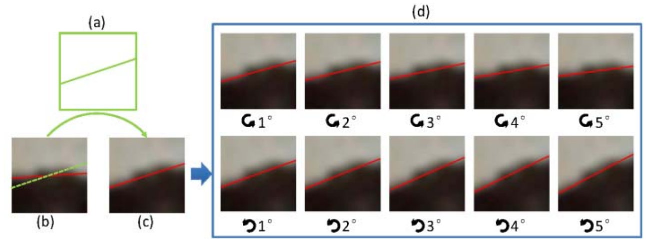

Figure 7. Structure-based rotated patch candidate set. (a) target patch in the hole(the line inside the patch denotes the approximate structure); (b) one source patch(the red line denotes its own structure and the green corresponds to the target); (c) rotated source patch consistent with the target; (d) rotated patch candidate set within positive five and negative five degrees by a step size of one degree.

where Ω denotes the missing texture region, Ω_B denotes the boundary region of Ω where the patches has partial but incomplete texture, and P represents the sample patch, $N(P_i)$ denotes the neighbor patches of P_i. $E(P_k)$ constrains the boundary smoothness between the known and unknown region, while $E(P_i, P_j)$ measure the coherence constraint between two adjacent patches in Ω .

Furthermore, we define $E(P_k)$ and $E(P_i, P_j)$ as

$$E(P_k) = \sum_{BP_i \epsilon BL_k, BP_i{}' \epsilon BL_k{}'} \left\| BP_i - BP_i{}' \right\|^2 \quad (3)$$

where BL_k denotes the borderline of the texture region and no texture region in the source patch, $BL_k{}'$ denotes the borderline in the filled patch, and $BP_i, BP_i{}'$ denote the points inside the borderline, $\left\| BP_i - BP_i{}' \right\|^2$ simply calculates the color normalized squared differences (SSD) between the points.

$$E(P_i, P_j) = \sum_{OP_k \epsilon P_i \bigcap P_{i \bigcap j}, OP_k{}' \epsilon P_j \bigcap P_{i \bigcap j}} \left\| OP_k - OP_k{}' \right\|^2 \quad (4)$$

where OP_k denotes the points overlapped in patch P_i, while $OP_k{}'$ denotes the ones in patch P_j, and $\left\| OP_k - OP_k{}' \right\|^2$ calculates the color SSD value. It is noteworthy that here the overlapped points in different sample patches mean that they correspond to the same three-dimensional point on model surface.

Improve Sun's algorithm [17], when measure the structure similarity between the source patch and the target one, we do not use the energy but establish a rotated patch candidate set for each target patch instead. According to the completed structure line, we approximately treat the line inside one patch as a straight line. Between one target patch line and each source one, there is an angle difference $\Delta\theta$. And we define an angle range as $\Delta\theta$ within plus or minus a threshold angle, which always defined as ten degrees, for each source one. Followed by a step size of one degree, we rotate each source patch in the range illustrated above and add the rotated patches into the candidate set, as shown in Fig.7. Though we increase the computational complexity, we could achieve more precise result. And in this way, more or less, the lack of source texture diversity could be solved.

B. Remaining texture completion

As the structure texture is completed, there still exists large unknown regions that need to be filled. In this paper, we use a variant of texture synthesis method [24] to finish the completion work. First, with defined patch sizes, the remaining hole region is divided into lattice patches. A multi-scale graph is then built for the hole region. Then, using a Simulated Annealing based Markov Chain Monte Carlo method, we apply an inference algorithm to find a global optimization solution. Finally, with the global optimization solution W^* shown below, the completion work of remaining texture is accomplished.

$$W^* \propto \arg\max_{W} \{P(\Psi|W)P(W)\} \tag{5}$$

where

$$P(\Psi|W) = \frac{1}{Z}\exp\left\{-\sum_{x_i\in\Omega}\frac{D(p(x_i),\Psi(x_i))}{2\sigma(x_i)^2}\right\} \tag{6}$$

$$P(W) = \frac{1}{Z}\exp\left\{-\sum_{x_i,x_j\in N}\frac{D(p(x_i),p(x_j))}{2\sigma(x_i,x_j)^2}\right\} \tag{7}$$

where $P(\Psi|W)$ is the likelihood probability for a multi-scale graph patch with known texture, and $P(W)$ is the prior probability for one without texture. Just similar to the energy illustrated in Eq.(3) and Eq.(4), according to the mapping between the model vertices and the local sample patches, $D(p(x_i),\Psi(x_i))$ constrains the boundary patches of unknown region to match well with the known part, and $D(p(x_i),p(x_j))$ measures the coherence constraint between two adjacent completed patches. And basing the standard deviation of the normal distribution function $N(0,\sigma^2)$, a context-aware parameter $\sigma(x_i),\sigma(x_i,x_j)$ is defined to ensure that the completed patch and known patch are visually plausible. For some sharp seams generated by the overlapped patches, similar to [25], we use the possion method to achieve smooth results.

IV. Results

We test our method on the conditions of both prior knowledge awareness and not. The input of our method is a 3D model with incomplete texture. The models and textures are all from [26]. In order to obtain the incomplete texture, we cut off some areas on the texture atlas, usually one hole for one experiment. All experiments were run on a 3.1GHz PC.

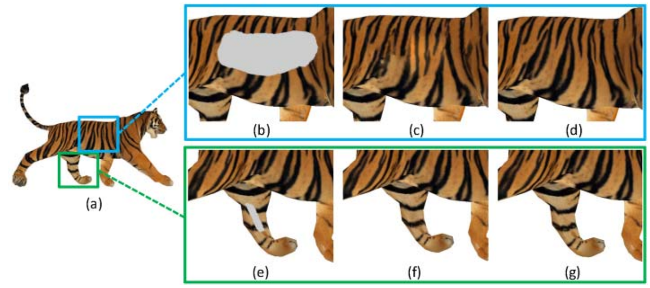

Figure 8. Texture completion results of the tiger model. (a) the ground truth texture model; (b) and (e) show two zoomed models with different missing texture region on their surfaces, while the texture hole region is displayed by gray; (c) and (f) respectively show the final completion results of (b) and (e); (d) and (g) show the ground truth correspond to (c) and (f).

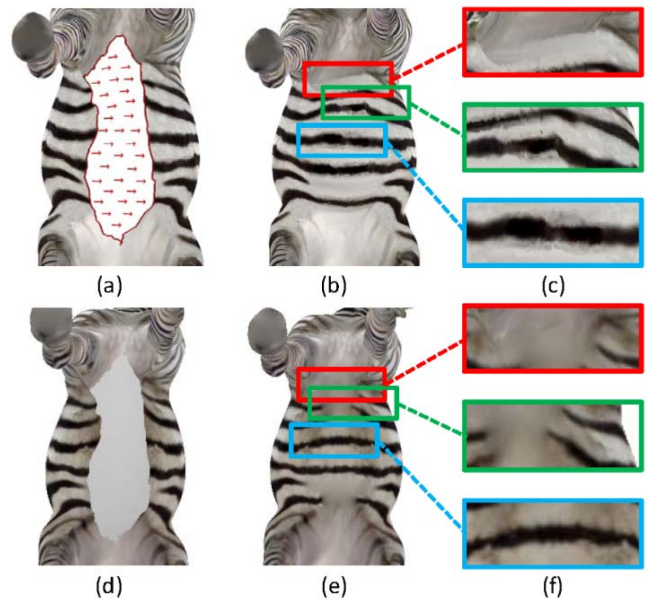

Figure 9. Comparison with Zhou's approach. (a), (b) and (c) respectively represent the input(the red arrows denote the vectors interactively specified by user), the output texture model and the zoom blocks from Zhou et al. [14] and (d), (e) and (f) behave ours.

As shown in Fig.8, we cut off different regions (leg and body) on the texture atlas of tiger model. With the guidance of the structure and no prior knowledge, the completion result is both realistic and smooth compared to the ground truth.

Fig.9 shows comparison results of the zebra model with Zhou et al. [14]. Although our input is slightly different with [14], with less interaction, our result is visually more natural and less artificial. And exhibited in Fig.10, with the prior knowledge, the completion result according to the real zebra in nature is achieved.

Depending on the model complexity and the hole size, time cost varies from about 120s to 600s. For the tiger model has 5396 vertices and 10788 faces while the zebra model

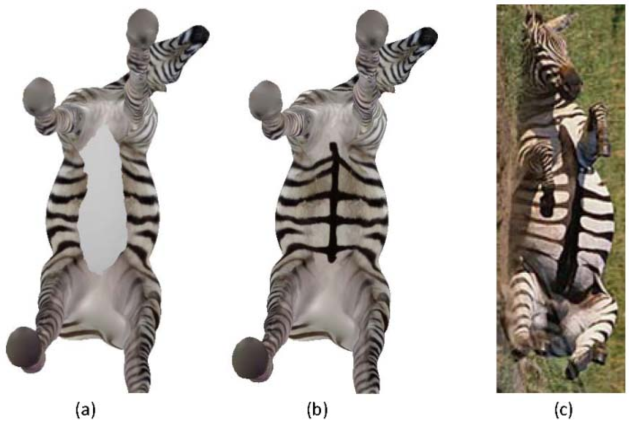

Figure 10. Texture completion result of the zebra model with prior knowledge. (a) the input zebra model with a texture hole denoted by gray on the surface; (b) the completed model with prior knowledge; (c) real zebra photo in nature.

has 20157 vertices and 40310 faces, the completion time cost of the tiger's leg is about 120s while the tiger's body completion costs about 300s, and the stomach completion of the zebra costs about 600s.

V. Conclusion and limitation

This paper proposes a novel method to complete the texture hole on 3D model surface by integrating surface geometric features. It divides the completion process into structure and texture components. Considering the surface geometry feature and maintaining the consistency of the existing structure, the missing structures are firstly predicted by the tensor voting algorithm. And guided by the completed structures, the remaining texture is completed by the MCMC algorithm. Finally we obtain optimal and convincing results.

As an additional special case illustrated in this paper, prior knowledge which accords to the real object and the surface geometry feature is used to assist the propagation of the structure. And the result we show is realistic.

Even though we could gain satisfied results, there still exists many defects and limitations in our method. As structure detection and extraction is a hard segment problem, our method only deals with the missing texture model which has obvious texture structures at present. And although complete the texture on the model surface by many iterations, there also exists sharp seams cased by the overlapped patches somewhere. For seamless texture completion, we have to use the possion method in those places. In the future, we would like to consider more complex texture and find more natural patterns.

Acknowledgment

This work was partially supported by 863 Program (2012AA011504), ASFC (2011ZC51027), R&D Program (2013BAH35F01), and CSTC (cstc2012ggC40004).

References

[1] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, pp. II:721–II:728.

[2] I. Drori, D. Cohen-or, and H. Yeshurun, "Fragment based image completion," in *SIGGRAPH*, 2003, pp. 303–312.

[3] G. Turk, "Texture synthesis on surfaces," in *SIGGRAPH*, 2001, pp. 347–354.

[4] S. Lefebvre and H. Hoppe, "Appearance-space texture synthesis," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 541–548, 2006.

[5] S. Magda and D. Kriegman, "Fast texture synthesis on arbitrary meshes," in *the 14th Eurographics Workshop on Renderin*, 2003, pp. 82–89.

[6] S.-C. Zhu, Y. Wu, and D. Mumford, "Filters, random fields and maximum entropy (frame)," *International Journal of Computer Vision (IJCV)*, vol. 27, no. 2, pp. 107–126, 1998.

[7] J. Han, K. Zhou, L.-Y. Wei, M. Gong, H. Bao, X. Zhang, and B. Guo, "Fast example-based surface texture synthesis via discrete optimization," *The Visual Computer*, vol. 22, pp. 918–925, 2006.

[8] L. Ying, A. Hertzmann, H. Biermann, and D. Zorin, "Texture and shape synthesis on surfaces," in *the Eurographics Symposium on Rendering*, 2001, pp. 301–312.

[9] S. Zelinka and M. Garland, "Interactive texture synthesis on surfaces using jump maps," in *14th Eurographics workshop on Rendering*, 2003, pp. 90–96.

[10] L.-Y. Wei and M. Levoy, "Texture synthesis over arbitrary manifold surfaces," in *SIGGRAPH*, 2001, pp. 355–360.

[11] K. Xu, D. Cohen-Or, T. Ju, L. Liu, H. Zhang, S. Zhou, and Y. Xiong, "Feature-aligned shape texturing," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, pp. 108–115, 2009.

[12] M. Walter, A. Fournier, and D. Menevaux, "Integrating shape and pattern in mammalian models," in *SIGGRAPH*, 2001, pp. 317–326.

[13] J. Zhang, K. Zhou, L. Velho, B. Guo, and H. Shum, "Synthesis of progressively variant textures on arbitrary surfaces," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 295–302, 2003.

[14] K. Zhou, X. Wang, Y. Tong, M. Desbrun, B. Guo, and H. Shum, "Texture montage," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 1148–1155, 2005.

[15] C. Xiao, W. Zheng, Y. Miao, Y. Zhao, and Q. Peng, "A unified method for appearance and geometry completion of point set surfaces," *The Visual Computer*, vol. 23, no. 6, pp. 433–443, 2007.

[16] C. Tang and G. Medioni, "Inference of integrated surface, curve and junction descriptions from sparse 3d data," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 20, no. 11, pp. 1206–1223, 1998.

[17] J. Sun, L. Yuan, J. Jia, and H. Shum, "Image completion with structure propagation," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 861–868, 2005.

[18] Z. Tu and S. Zhu, "Image segmentation by data-driven markov chain monte carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 24, no. 5, pp. 657–673, 2002.

[19] C. Yuksel, J. Keyser, and D. House, "Mesh colors," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 2, p. 15, 2010.

[20] J. Jia and C. Tang, "Image repairing: Robust image synthesis by adaptive nd tensor voting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, pp. I:643–I:650.

[21] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. Gortler, and H. Hoppe, "Fast exact and approximate geodesics on meshes," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 553–560, 2005.

[22] S. Yoshizawa, A. Belyaev, and H. Seidel, "Fast and robust detection of crest lines on meshes," in *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, 2005, pp. 227–232.

[23] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 362–371, 2002.

[24] X. Chen, B. Zhou, Y. Guo, F. Xu, and Q. Zhao, "Structure guided texture inpainting through multi-scale patches and global optimization for image completion," *SCIENCE CHINA Information Sciences*, 2013.

[25] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 313–318, 2003.

[26] http://www.kunzhou.net/tex-models.htm.