HAL open science

An Optimized Sub-texture Mapping Technique for an Arbitrary Texture Considering Topology Relations

Sangyong Lee, Cheonshik Kim, Seongah Chin

To cite this version:

Sangyong Lee, Cheonshik Kim, Seongah Chin. An Optimized Sub-texture Mapping Technique for an Arbitrary Texture Considering Topology Relations. 9th International Conference on Network and Parallel Computing (NPC), Sep 2012, Gwangju, South Korea. pp.559-564, 10.1007/978-3-642-35606-3_66 . hal-01551310

HAL Id: hal-01551310

https://inria.hal.science/hal-01551310v1

Submitted on 30 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

An optimized sub-texture mapping technique for an arbitrary texture considering topology relations

Sangyong Lee[1], Cheonshik Kim[2], Seongah Chin[3*]

[1]Department of Computer Science and Engineering, Korea University, Seoul, Korea
[2]Department of Computer Engineering, Sejong University, Seoul, Korea
[3]Division of Multimedia Engineering, Sungkyul University, Anyang-City, Korea
xyleez@gmail.com, mipsan@paran.com, solideochin@gmail.com

Abstract. In recent years, texture mapping techniques have become more advanced as providing real-time rendering tasks. However, few methods have been conducted in optimizing a sub-texture mapping for an arbitrary texture. In this paper, we present a method to optimize sub-texture mapping by reflecting topology relations in which an arbitrary sub-texture source has to be represented as minimizing distortions of the sub-texture source. We have defined a manipulating mechanism that helps a user control a texture source using a track ball interface in order to minimize loss of the texture source. And a sub-texture frame has to be aligned to dominant geometric structures of 3d model by employing some transformations. We have shown experimental results to verify the proposed method as well.

1 Introduction

In general, texture mapping is one of the fundamental techniques which are broadly used to enhance the final appearance of a 3d model by attaching a texture source into the 3d model. In recent years, 3d video games have extensively utilized some advanced texture mappings in real-time rendering such as normal mapping, parallax mapping and displacement mapping etc [1][2][3]. Even in game industry it is becoming more and more important to create cinematic appearance of digital scenes. In certain applications, we need to attach sub-textures like face paintings, scars, wounds, and decorations. Texture parameterization between a set of object mesh and a sub-texture source is critical to minimize texture stretch or texture deviation [4][5][6]. However very few researches concerning sub-texture mapping techniques for an arbitrary sub-texture have been reported, which implies they do not seem to reflect topology relations between the geometric structures of a 3d model and a sub-texture source. This fact causes some loss of the texture source that has to be appeared when the sub-texture source is clapped or scaled. If the sub-texture is larger than the region

* corresponding author

of meshes where a user wants to attach it then the source has no choice but to lose the boundary area. In the case we reduce the source image so that some parts mostly around the boundary of meshes would be filled with background source. In fact, we have to make alignment of the sub-texture source corresponding to the geometric structures by employing interactive interface manipulated by a general user in order to keep the sub-texture source as closely as we really want.

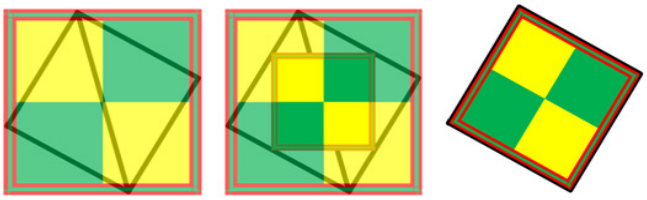
Hence, in this approach, we propose an optimized sub-texture mapping for an arbitrary texture by reflecting topology relations between a 3d models and a sub-texture source. The motivation of the approach is to avoid loss of the sub-texture source. At first, a trackball interface is defined to manipulate a 3d model, which helps a user control a texture source in order to minimize loss of the texture source. Rotation and scaling transformations are required to align the texture source to dominant geometric structures of 3d model. We have shown experimental results to verify the proposed method as well.

2 The proposed approach

The fundamental idea of the approach comes from how to attach a sub-texture image as is without losing any parts. A texture coordinate of a source image has to be accurately matched into a vertex on a 3d mesh model, where a user wishes to attach. However it is not easy to succeed in what we want since geometric structures of a 3d mesh model are not necessarily corresponding to a texture source frame. Hence, we have to seek a proper method to align a sub-texture source to suitable match of the vertices of a 3d model.

2.1 Trackball Interface

We have to keep in mind that a proper match between geometric structures of a 3d mesh and a sub-texture source needs to be solved when texture mapping is carried out. At first, we fix the sub-texture source on the screen then we have employed a trackball interface that makes us keep adjusting transformation of the 3d model until we acquire what we want to have with the sub-texture source. This trackball interface is suitable since it can provide flexible controls to let a user manipulate the 3d model.



(a) clipped (b) scaled (c) rotated

Fig. 1. Variant occurrences of texture mapping

As shown in Fig. 1, there are several cases that we may expect when doing sub-texture mapping. A user picks a specific mesh on a 3d model where he wants to represent a sub-texture. A sub-texture can be clipped as shown in (a). Reduction of a sub-texture source is shown in (b). A wanted mapping appears in (c) after rotations of the sub-texture source. A user can utilize a trackball interface in an interactive way that may provide flexible control to match what they want to represent with a sub-texture by rotating the 3d model. In particular, it is critical to provide an accurate matching when a sub-texture source seems to be complex enough with sophisticated patterns since we do not want to expect lots of losing the sub-texture source. To realize a trackball interface, Eq. (1) is used.

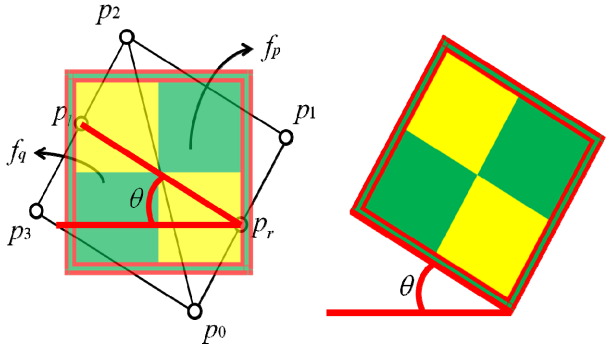
$$R_{zyx}=R_z(\theta_z)R_y(\theta_y)R_x(\theta_x)$$

$$=\begin{bmatrix} \cos\theta_y\cos\theta_z & \sin\theta_x\sin\theta_y\cos\theta_z-\cos\theta_y\sin\theta_y & \cos\theta_x\sin\theta_y\cos\theta_z+\sin\theta_y\sin\theta_y \\ \cos\theta_y\sin\theta_z & \sin\theta_x\sin\theta_y\sin\theta_z+\cos\theta_x\cos\theta_z & \cos\theta_x\sin\theta_y\sin\theta_z-\sin\theta_x\cos\theta_z \\ -\sin\theta_y & \sin\theta_x\cos\theta_y & \cos\theta_x\cos\theta_y \end{bmatrix}, \quad (1)$$

where the rotation angles with x, y, and z axis are θ_x ,θ_y and θ_z. And $R_x(\theta_x)$, $R_y(\theta_y)$ and $R_z(\theta_z)$indicate the transformations of rotations.

2.2 Texture Alignment

Let a center mesh that a user clicks and wants to attach a sub-texture source be f_p source. This center mesh is composed of three line segments. Let the longest line segment be f_q among three line segments. Then we can acquire the rotation angle θ using inner product of two vectors as shown in Fig. 2(a). Once we have carried out the rotation then the sub-texture source is aligned to the mesh model. The final appearance looks as shown in Fig. 2 (b).



(a) Schematic diagram (b) After rotation

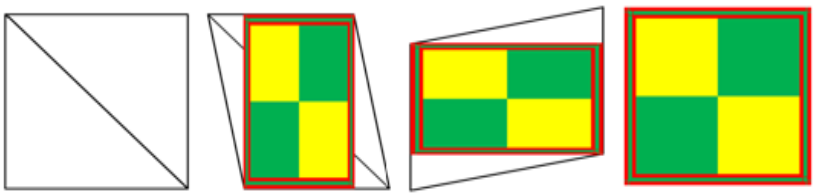
Fig. 2. Conceptual diagram of rotation

2.3 Scaling Interface

If the mesh that a user selects is skewed and stretched along one direction and a sub-texture source is like a square or round shaped, we require a scaling transformation in order to represent the sub-texture sources that we want to attach. If we do not deal with the surroundings, then some parts of the sub-texture source cannot be appeared since the mesh looks skewed. To solve this, we understand that 3d coordinates on the mesh model are projected onto the screen coordinates. In the processing this procedure, the normalization has to be done using the maximum and minimum coordinates which will be used as texture coordinates. In Fig. 3, we display the conceptual diagrams to help us comprehend the approach of scaling. The mesh is shown in Fig. 3(a). Horizontal scaled texture mapping and vertical scaled one appear in Fig. 3(b) and (c) respectively given the texture source in (d). The screen coordinates can be translated into texture coordinates using Eq (2).

$$(x', y') = \left(\frac{x}{\max(x) - \min(x)}, \frac{y}{\max(y) - \min(y)}\right), \qquad (2)$$

where *(x, y)* is the screen coordinates and *(x ′y ′)* is the texture coordinate.



(a) 3d mesh (b) Horizontal scaled (c) Vertical scaled (d) Texture source

Fig. 3. Conceptual diagram of scaling.

3 Results and Discussion

To verify the proposed methods, we have carried out experiments under Intel Core (TM) i5 CPU 750@2.67GHz, 2.99RAM, NVIDIA GeoFore GTS 250 (VGA) with Windows XP Professional. The proposed methods have been realized using Microsoft MFC and OpenGL API. The comparisons between the proposed methods and the general method that does not consider automatic transformations have been conducted.

In the experiments, we have made a number of texture samples that can be used in face paintings and decorations of the face. We have carried out the proposed texture mapping by attaching the sub-texture onto the 3d face model. The sub-textures are composed of face painting pictures shown in Fig. 4 on the first row. Face painting sub-texture mapping results have been created with scaling factors 1, 0.9, and 0.8 shown in the first, second and third respectively from the left in Fig. 4. The proposed

method and mesh models are shown in the fourth and last column in Fig. 4. As we have shown here, the proposed method creates the sub-textures as good as the original ones comparing to the other results mostly showing loss of the sub-textures.

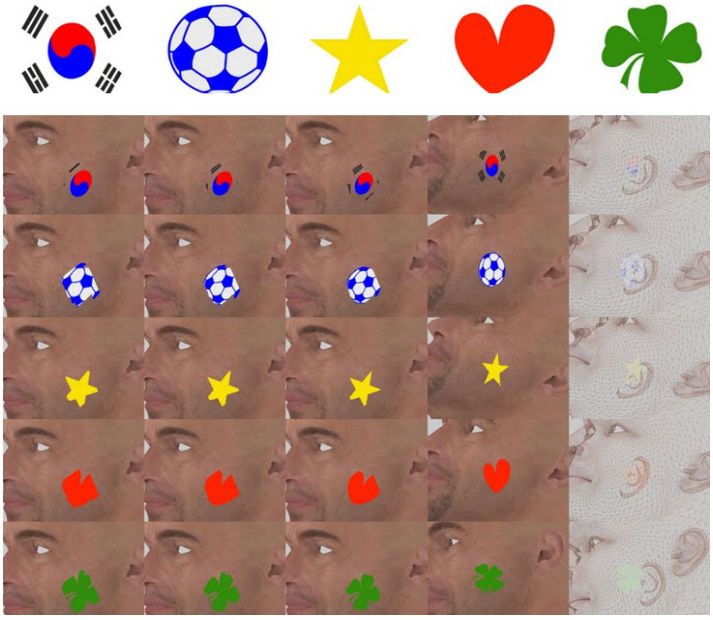


Fig. 4. The first row shows the sub-texture sources. Face painting sub-texture mapping results are shown with scaling factors 1, 0.9, and 0.8 in the first three columns. The proposed method and mesh models are shown in the fourth and last from the left column.

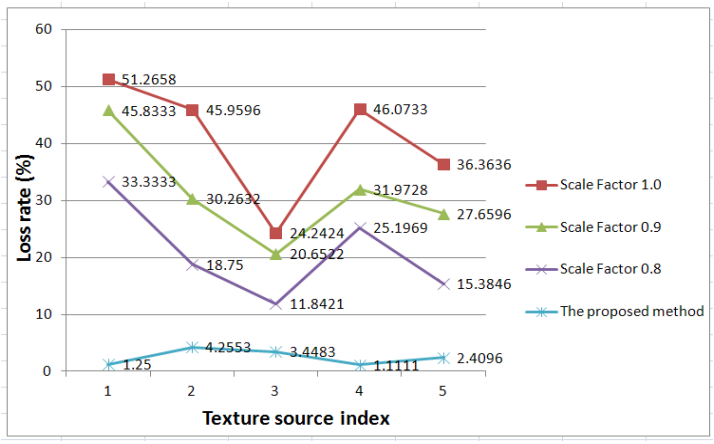


Fig. 5. Loss rate comparisons

In Fig. 5, we have shown loss rate (%) comparisons computed by the number of loss mesh in Fig. 4 indicating that the proposed method in the fourth column in Fig. 4, shows better loss rate than others created by scaling factors observed in the first, second and third column in Fig. 4.

Authors want to mention that mesh tagging tasks are skipped in this article, which will be reported in the other article, since mesh tagging tasks required for sub-texture mapping are beyond the scope of the proposed method.

4 Concluding Remarks

In this paper, we present a technique that makes it possible to automatically create an optimized sub-texture mapping. This basic idea is from matching topology relations between geometric structures of a 3d model and a sub-texture source. The approach starts with providing a trackball interface that utilize a convenient user control when texture mapping. Once a user gains a sub-texture source then a 3d model can be manipulated using the trackball functions until the user is satisfied with the texture appearance. This functionality makes the sub-textures be as good as the appearance the user wants to attach with keeping the sub-texture source as much as we represent.

Acknowledgement

This work was partially supported by Korea Science and Engineering Foundation (NRF) grant funded by the Korean government (No. 2012-0002889).

References

1. Sloan, P., : Normal Mapping for Precomputed Radiance Transfer. Proceedings of the 2006 symposium on Interactive 3D graphics and games, ACM, New York, pp. 23-26(2006)
2. Gao, R., Yin, B.; Kong, D., Zhang, Y., Si, H., : An Improved Method of Parallax Mapping. IEEE Conf. on Computer and Information Technology, Sydney, pp.30-34(2008)
3. Szirmay-Kalos, L., Umenhoffer , T., : Displacement Mapping on the GPU — State of the Art. Computer Graphics Forum, 27(6), pp. 1567-1592(2008)
4. Sander, P., Snyder, J., Gortler, S., Hoppe, H., : Texture Mapping Progressive Meshes. ACM SIGGRAPH 2001, pp. 409-416(2001)
5. Praun, E., Finkelstein, A., Hoppe, H., : Lapped Textures, ACM SIGGRAPH 2000, pp. 465-470(2000)
6. Neyret, F., Cani, M., : Pattern-based texturing revisited. ACM SIGGRAPH 1999, pp. 235-242(1999)