Title:
Editable Texture Map Generation and Optimization Technique for 3D Visualization Presentation

Authors:
Tsung-Chien Wu, rabbit94mesh577@gmail.com, National Central University, Taoyuan, Taiwan
Jiing-Yih Lai, jylai@ncu.edu.tw, National Central University, Taoyuan, Taiwan
Watchama Phothong, p_watchama@hotmail.com, National Central University, Taoyuan, Taiwan
Douglas W. Wang, dwmwang@gmail.com, Ortery Technologies, Inc., Taiwan
Chao-Yaug Liao, cyliao@ncu.edu.tw, National Central University, Taoyuan, Taiwan
Ju-Yi Lee, juyilee@ncu.edu.tw, National Central University, Taoyuan, Taiwan

Keywords:
Conformal Mapping, Direct Texture Mapping, Mesh Parameterization, Pixel Extraction, Triangular Model

DOI: 10.14733/cadconfP.2017.252-257

Introduction:
Two-dimensional (2D) images of an object are commonly used for product presentation in e-commerce, mainly because these images can reveal the object's color in high quality and are easy to process. However, 2D images provide only limited viewpoints of an object. Three-dimensional (3D) visualization is an alternative technique for product presentation, in which multiple 2D images showing different viewing angles are integrated. The user can orient a 2D image at a given viewing angle via a viewing interface. However, in such a presentation, the user can only view images that were captured beforehand. Further, owing to limited angles recorded, the orientation process is not fluent. In addition, the actual 3D shape and dimensions of an object cannot be obtained using this approach. A 3D model combined with the object's texture, called 3D color model hereafter, is an alternative approach for product presentation in e-commerce applications. Further, the texture might have the photo inconsistency issue at the junctions. Various techniques can be employed to create a texture map for a 3D color model.

The mesh parameterization technique in computational geometry provides several practical applications. Sheffer et al. [10] and Hormann et al. [3] introduced and summarized several typical methods of mesh parameterization and its applications, e.g. texture mapping, normal mapping, detail transfer, morphing, mesh completion, editing, database, remeshing, and surface fitting. The available techniques for mesh parameterization can be divided into types relating to distortion minimization, fixed or free boundary, or numerical complexity. For distortion minimization, an objective function can be formulated in terms of angles, areas or distances, and it is minimized to yield the optimized mapping of the model from the 3D domain to the parametric (UV) domain. The fixed boundary can be obtained by a simple formulation, allowing for an easy solution, but, the distortion in parameterization is usually quite large. In contrast, the free boundary has less distortion in parameterization, but obtaining the solution is time-consuming because the boundary is considered as part of the solution. Numerical complexity is divided into linear and nonlinear methods. The nonlinear method is complex and requires more computational time, but it yields less distortion in the result.

The angle preserving approach, also called conformal mesh parameterization, aims to minimize the angle distortion when unwrapping a 3D model onto the UV domain. Sheffer et al. [8] proposed a mesh parameterization algorithm, called angle-base flattening, to optimize the angles on the UV domain. The topology of triangular meshes is set as constraints to maintain the correctness of the mesh topology on the UV domain. Sheffer et al. [9] proposed another method to improve the efficacy of the optimization process. In addition, the hierarchical algorithm was employed to deal with the situation of huge amount

Proceedings of CAD'17, Okayama, Japan, August 10-12, 2017, 252-257

of triangular meshes. Zayer et al. [12] proposed a method to apply linear equations for solving the optimization problem. Linear equations were derived from the angle-base flattening approach, in which topological constraints were specified. Lévy et al. [4] solved for the meshes on the UV domain based on a least-squares approximation of the Cauchy-Riemann equations. This method can minimize both angle distortion and area distortion, and also establishes the topology of the planar meshes, which can prevent the topology error, such as face flip, on 2D meshes.

The technique of texture map generation not only deals with mesh parameterization, but also solves the texture transferring problem. Niem et al. [6] proposed a procedure of texturing the meshes that includes grouping the meshes using the camera information to find the most appropriate image source, filtering the boundary between two different groups to minimize the color inconsistency, and synthesizing the invisible meshes using the neighboring color. Genç et al. [2] proposed a method to extract the pixels and render the texture dynamically. The extraction is performed by scanning the pixels horizontally and rendering every color onto the meshes. Baumberg [1] proposed an algorithm to process the color difference from two different images using a blending method. The images were filtered into high and low bands. The low band images were averaged to minimize the color difference, whereas the high band images were kept to maintain the outline of the object boundary.

Main Idea:

In this study, a method in accordance with conformal mesh parameterization is developed for unwrapping 3D triangular meshes onto the UV domain and an integrated process is proposed for direct texture mapping. The proposed method can generate an editable texture map for manual editing. Additionally, the most critical problem in texture mapping is that the junctions of different images on the texture usually cannot be connected properly. Considerable seams owing to the inconsistency in

Fig. 1: Visualization of the steps of the proposed method, (a) Input object images, camera information, and object meshes, (b) divide 3D meshes into segments for mesh parameterization, (c) unwrap 3D meshes onto the UV domain to yield a UV map, (d) grouping of 3D meshes, and (e) final texture map and 3D color model.

photo appear on the 3D color model. An algorithm is proposed to detect and improve the photo inconsistent problem at the transitions of different images on the texture.

Proceedings of CAD'17, Okayama, Japan, August 10-12, 2017, 252-257

To generate a 3D color model, the object images must be captured sequentially in a controlled environment. In addition, the camera information should be obtained by capturing the calibration mat and performing the calibration process [5]. The object's 3D model is constructed by the shape-from-silhouette (SFS) algorithm [7], and the shape and surface of the 3D model are optimized by silhouette and smoothing factors [7], [11]. The aforementioned process can yield a 3D model for texture mapping, where the 3D model is composed of triangular meshes. Fig. 1 indicates the steps of the proposed method for generating an editable texture map and 3D color model. The first step is to input the optimized 3D triangular meshes, the object images and camera information. Next, the 3D meshes are separated into several segments. Subsequently, each mesh segment is unwrapped onto the UV domain and all 2D meshes on the UV domain are packed together to form a UV map. Furthermore, the triangles on 3D meshes are grouped, with each group of meshes projected onto an image with the most appropriate viewing angle. Finally, the color extraction and pixel placement are implemented simultaneously. The pixels covered by each triangle on the UV map are filled in using the pixels extracted from the corresponding image. The 3D color model is then output and saved as an OBJ file. The techniques used to achieve these tasks are described below.

1. **Editable conformal mesh parameterization**:
The basic idea of the proposed conformal mesh parameterization is to unwrap 3D meshes onto the UV domain while preserving the angles of each triangle on the UV domain. A critical issue of the preservation is that all angles of 2D meshes on the UV domain cannot be exactly the same as those of 3D meshes. Therefore, an optimization problem is formulated to minimize the deviation of angles on 2D meshes and determine the optimized positions of all vertices [8-9], [12].

1.1 **Input initial angles**:
The angles of all triangles on 3D meshes are regarded as the initial angles of those on 2D meshes. However, the summation of all angles surrounding a vertex on the 3D domain may exceed 360°, whereas it must be 360° on the UV domain. Therefore, a linear adjustment is implemented to obtain the initial angles on 2D meshes.

1.2 **Determine boundary conditions**:
When unwrapping 3D meshes onto the UV domain, the deviation between 2D and 3D meshes may occur due to the following problems: flip in triangles, degenerate triangles, and non-manifold edges. To maintain the correctness of the topology between 2D and 3D meshes and prevent the occurrence of irrational meshes, additional constraints on angles must be specified on unwrapping. In this way, the topology of the meshes on the UV domain after angles adjustment must be kept the same as that on the 3D domain. Three constraints on angles are as follows: (1) triangle consistency: the summation of all angles on a 2D triangle must be 180° (Fig. 2(a)); (2) vertex consistency: the summation of all angles surrounding an inner vertex should be 360° (Fig. 2(b)). It is noted that the vertices on the boundary should be excluded; (3) wheel consistency: the lengths of all edges neighboring a vertex should be equal (Fig. 2(c)). The aforementioned three constraints can be employed to guarantee the topological consistency of the 2D meshes with 3D meshes. The problems, e.g., degenerate triangle, triangle flip, and non-manifold edge, on 2D meshes can therefore be eliminated.

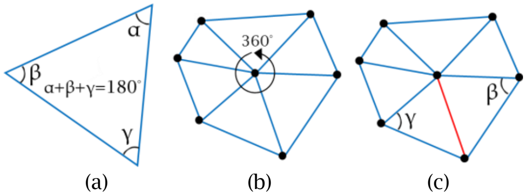

Fig. 2: Constraints for conformal mesh parameterization, (a) summation of angles on one triangle should be 180°, (b) summation of angles surrounding a vertex should be 360°, (c) the lengths of an edge evaluated from two neighboring meshes, respectively, should be equal.

1.3 **Calculate optimized angles**:
The conformal mesh parameterization is obtained by minimizing the errors between the meshes on the 2D and 3D domains. From the constraint conditions, a set of initial errors on the UV domain has already

been determined. By minimizing this set of errors, the optimized angles can be obtained. The proposed algorithm employs a linear system (***A**x*=**b**) to solve for the optimized angles. The optimized angles can be obtained by applying a set of initial angles and upgrading the angles gradually. As some of the angles obtained may be larger than 180° or less than 0°, the optimization process is implemented iteratively. The iteration stops when all optimized angles are within the range 0°-180°.

1.4 **Calculate new vertices in accordance with optimized angles**:
The last step of mesh parameterization is to calculate the new positions of all vertices on 2D meshes in accordance with the optimized angles obtained. The calculation is based on the triangle similarity. The calculation of vertices on the UV domain can be applied using the least-squares approximation. For all 2D meshes, if the first two vertices on a mesh can be determined first, remaining vertices can be calculated using the least-squares approximation [10], which is formulated as a linear system ***A**x*=**b**. After solving this linear equation, all vertices can be obtained. The topology of all vertices on 2D meshes can be maintained correctly.

2. **Direct texture mapping algorithm**:
The 3D color model is generated from the 3D model covered with texture that stores the color information on a texture map. The 3D model with a texture map can provide the physical dimensions of an object as well as its color, enabling a vivid modeling of the object. However, the resolution and quality of the texture plays an important role regarding the feasibility of this technique in practical applications. High-quality object images can be obtained by an image taking process. Thus, the main idea of direct texture mapping is to texture the 3D model using the object images directly.

2.1 **Grouping of 3D triangles**:
A series of camera information and 3D meshes are input. Each triangle on 3D meshes is checked to find an image on which the texture of this triangle is to extract. All triangles using the same image are considered as a group. One component of the camera information is the looking direction, which represents the camera viewing vector. The camera viewing vector is perpendicular to the image plane. In addition, all 3D triangles have their own surface normal vectors. With these two vectors, the angle between a triangle and an image can be calculated. The primary image of a triangle is defined as the one with the minimum angle among all images. All triangles with the same primary image are regarded as a group.

2.2 **Extraction of pixels on the object images**:
The texture of the 3D color model comes directly from the object images. Each of the 3D triangles should be projected back to the image plane of its primary image. The projection is based on the prospective projection in accordance with the camera position and looking direction to project the 3D triangle to the image plane. As Fig. 3 depicts, the triangle on the image represents the projection of one 3D triangle on the image plane. All pixels on and inside this triangle represent the texture corresponding to it. To generate a texture map for all 2D triangles, the pixels of a 2D triangle on the texture map is obtained using the pixels obtained above. The extraction of any pixel inside a triangle is explained below. The sequence of the image pixel is similar to a grid plane, composed of horizontal and vertical lines. Thus, a scanline method can be implemented to evaluate all pixels inside a triangle. This scanline can intersect two edges on a triangle, which represent two end points of the scanline, respectively. As long as two end points of the scanline are known, all pixels within this scanline can be evaluated in sequence.

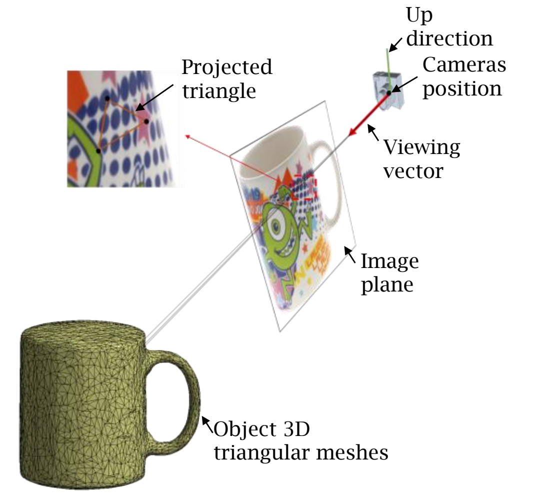

Fig. 3: The method for projecting a 3D triangle onto an image plane.

2.3 **Placement of pixels on the texture map**:

The pixels corresponding to each 2D triangle come from the previous step. However, the areas of the pixels from the previous step are different to those that should be filled on the UV map. Therefore, the main issue is how to extract the correct pixels from the previous step and place them appropriately on the UV map. As each 2D triangle on the UV domain is different from the projected mesh on an image domain, the pixels cannot directly be placed one by one. A transformation algorithm is developed to map the pixels between two pixel domains.

2.4 **Optimization of the texture map**:

The photo inconsistency problem is handled in this step. The basic idea is to extend the layer of the image to make the boundary lie on a place where the color difference is small. However, the original number of images used was four. The difference in the viewing angle for two adjacent images is 90°. This might also cause the photo inconsistency problem after extending the layer. Thus, the method to handle this kind of problem is to add more images to group the triangles. In this way, the difference in the viewing angle for adjacent images is reduced and the photo consistency can be improved.

3. **Examples and discussion**:

Several examples were employed to evaluate the feasibility of the proposed method. The editable texture map was used to verify the extension of the application. Furthermore, the integration of the color model and the 3D visualization viewer were presented and demonstrated.

The texture map in this approach is easy to edit and replace. As Fig. 4 shows, various patterns can be added to the texture map (Fig. 4 (a)) or an existing pattern can be replaced by another one (Fig. 4 (b)). Fig 4 (c) and (d) depict the results of 3D color models after editing. Fig. 5 shows the integrated new presentation viewer, where the 3D color model and 2D images are displayed alternatively. When the object is in a rotating mode, the 3D color model is displayed, enabling the rotating operation smoother. By contrast, when the object is in other modes, a 2D image is displayed, enabling a high-quality display of the object image. Fig. 5 (a) depicts the screenshot in the rotating mode, where the 3D color model is displayed. Fig. 5(b) depicts the screenshot when rotation has stopped, where a 2D image is displayed. Using the aforementioned switching mode to display various kinds of texture or image, the proposed new presentation method can be more realistic for e-commerce applications.

Fig. 4: The texture editing and replacing for the “cat doll” and “mug” example, (a) texture-edited texture map for the cat doll, (b) texture-replaced texture map for the mug, (c) texture-edited 3D color model, and (d) texture-replaced 3D color model.

<u>Conclusion:</u>

In this study, we proposed a method for performing conformal mesh parameterization to unwrap the mesh to a two-dimensional (UV) domain and for direct texture mapping from the extraction of the object image. The main contribution of this study was that we analyzed the photo inconsistency problem at the junctions of different image sources and the seam lines problem between different mesh segments, and proposed an optimization method to improve the quality of 3D texture mapping. To eliminate the photo inconsistency problem, additional images were used to minimize the discontinuity issue and the boundary color was checked for

Fig. 5: The integrated new presentation viewer example for the “blue shoe” example, (a) object 3d color model display when rotating, (b) object image display when rotation ha stopped.

determining the extending criterion. To eliminate the seam lines between different mesh segments, the texture border of each mesh segment was expanded in terms of the resolution of the texture map. The feasibility of the proposed texture optimization method has been demonstrated. Additionally, an integrated presentation viewer was proposed to extend the application of the 3D visualization presentation and to solve the disadvantage of the original 3D visualization presentation application in e-commerce.

References:

[1] Baumberg, A.: Blending Images for Texturing 3D Models, BMVC, 3, 2002, 5. http://dx.doi.org/10.5244/C.16.38

[2] Genç, S.; Atalay, V.: Texture extraction from photographs and rendering with dynamic texture mapping, Image Analysis and Processing, 1999, 1055-1058. http://dx.doi.org/10.1109/ICIAP.1999.797737

[3] Hormann, K.; Lévy, B.; Sheffer, A.: Mesh parameterization: Theory and practice, ACM SIGGRAPH 2007 courses on - SIGGRAPH 07, 2007, 1. http://dx.doi.org/10.1145/1281500.1281510

[4] Lévy, B.; Petitjean, S.; Ray, N.; Maillot, J.: Least squares conformal maps for automatic texture atlas generation, ACM Transactions on Graphics (TOG), 21(3), 2002, 362-371. http://dx.doi.org/10.1145/566654.566590

[5] Liao, C. Y.; Xiong, Y. S.; Wang D.W.; Lai J. Y.; Lee J. Y.: A camera calibration process for 3D digital model reconstruction of huge objects, 2016 Machining, Materials and Mechanical Technologies, Matsue Terrsa, Matsue, Japan: 7-11 October 2016. Japan: IC3MT, 2016.

[6] Niem, W.; Buschmann, R.: Automatic Modelling of 3D Natural Objects from Multiple Views, Image Processing for Broadcast and Video Production, 1995, 181-193. http://dx.doi.org/10.1007/978-1-4471-3035-2_15

[7] Phothong, W; Wu T.C.; Lai, J. Y.; Wang D.W.; Liao C. Y.; Lee J. Y.: 3D Model Reconstruction and Re-meshing, 2016 Machining, Materials and Mechanical Technologies, Matsue Terrsa, Matsue, Japan: 7-11 October 2016. Japan: IC3MT, 2016.

[8] Sheffer, A.; Sturler, E, de.: Parameterization of Faceted Surfaces for Meshing using Angle-Based Flattening, Engineering With Computers, 17(3), 2001, 326-337. http://dx.doi.org/10.1007/PL00013391

[9] Sheffer, A.; Lévy, B.; Mogilnitsky, M.; Bogomyakov, A.: ABF++: fast and robust angle based flattening, ACM Transactions on Graphics, 24(2), 2005, 311-330. http://dx.doi.org/10.1145/1061347.1061354

[10] Sheffer, A.; Praun, E.; Rose, K.: Mesh parameterization methods and their applications, Foundations and Trends in Computer Graphics and Vision, 2(2), 2006, 105-171. http://dx.doi.org/10.1561/0600000011

[11] Yemez, Y.; Sahilioglu, Y.: Shape from silhouettte using topology-adaptive mesh deformation, Pattern Recognition Letters, 30(13), 2009, 1198-1207. http://dx.doi.org/10.1016/j.patrec.2009.05.012

[12] Zayer, R.; Lévy, B.; Seidel, H. P.: Linear angle based parameterization, Fifth Eurographics Symposium on Geometry Processing-SGP, 2007, 135-141. http://dx.doi.org/10.2312/SGP/ SGP07/135-141