Mesh Geometric Editing Approach Based on Gpu Texture

Guiping Qian[1], YUE Wang[2]

[1]Assoc Prof., College of New Media, Zhejiang University of Media and Communications, Hangzhou, China

[2]Assoc Prof., Department of Mathematics, China Jiliang University, Hangzhou, China

Abstract

This paper presents a novel interactive mesh editing approach based on GPU texture mapping. The main feature is that it copies 2D surface geometry information to GPU frame buffer. The planar mesh information is transformed into GPU texture and placed on apposite position of target mesh. 3D information is retrieved after stitching two mesh components from the primitive vertex coordinates. When running real-time mesh cloning operator, our mesh editing approach can copy arbitrary irregular geometric features from source mesh to target mesh. Experimental results indicate that our method can outperform previous related mesh editing techniques.

Keywords: *Mesh Editing, GPU Buffer, Mesh Copy-paste*

1. Introduction

Mesh editing for three dimensional models has long been an active research area in geometric modeling and computer graphics. In this paper, we introduce a new mesh editing method, which can produce real-time continuous copying of arbitrary irregular mesh between two 3D models.

Most previous mesh editing methods reuse existing mesh parts to synthesize new models, which mainly concentrate on detail editing, deformation or Boolean operations. Laplacian-based editing method [1], [2], [3] use signal processing filters for enhancing or deforming some specified frequencies and regions. Yu et al.[4] introduced a mesh editing method by bilateral filter and shifting vertex positions with their Poisson-based mesh gradient field manipulation. However, these detail editing methods for filtering face normals proposed so far are limited to smoothing and enhancing, and are difficult to carry the information about the size of local details.

Mesh cloning tools commonly allow copy-paste of height field geometric details [4] or transplanting of entire large-scale features [5], [6] which is realized as a one-time operation rather than a continuous cloning. The approaches based on variational formulation [4] and [5], which require solving a large sparse linear system problem, are costly and prohibits the use of continuous cloning in real time. Geobrush[7] can clone interactively 3D mesh parts using a paintbrush interface, which is similar to the continuous cloning brush popular in image editing. However, automatic generation of cage meshes leads to frequent distortions and self-intersections at the boundary of pasted region.

Our key contribution is a mesh processing with GPU general-purposed computation. By mapping mesh vertices in selected region of interest (ROI) of source mesh onto a 2D image, The 2D topological information is abstracted into image texel and placed on apposite position of target mesh. 3D information is retrieved afterwards. The pasted mesh in selected region is deformed to keep C0 continuity, which avoids to applying cage meshes that frequently lead to severely distorted and even self-intersecting. Details of these applications will be introduced in Section 2 & 3.

Our mesh editing method has been successfully applied to mesh deformation, copy-paste and Boolean operations. These operations have been integrated into a mesh editing system with a few novel real-time interactive processing. Our work adopts the round brush and mesh stamp compared with the GeoBrush system analogously, while present a novel approach to overcome the difficulty in long and protuberant mesh clone. Through the direct projective mapping method, mesh clone can be realized without closed cage which induce to instability and fallibility. In section 4, the experimental results will show our work excellence.

2. Mapping

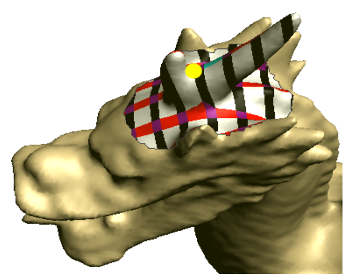
Our editing system draws inspiration from many previous modeling techniques and mesh editing approaches, such as coordinates-based image cloning[8], [9] and Meshmixer[6], and brings their capabilities to our mesh clone system. Compared with the GeoBrush system [7], our work adopts the round cloning brush and mesh stamp analogously, while present a novel approach to overcome the difficulty in long and protuberant mesh clone. By the extended mean value coordinates, mesh clone and deformation can be realized without enclosed cage which induce to instability and fallibility.

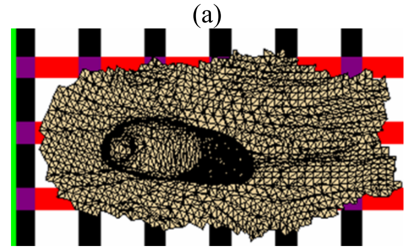
When the user explicitly defines stamps on both source and target meshes, the stamped meshes provide a local map correspondence between two meshes. Firstly the user clicks on the source mesh to define a stamp for copy. Its boundary should be an enclosed loop. Secondly, the mesh with stamp will be projected and parameterized to a plane which intersected with the center of the stamp. Finally, the user makes a stamped mesh on the target mesh as the corresponding source stamped mesh. The user can interactively rotate or scale the mapped zone of source mesh on target mesh. Through the round brush the ROI of source, mesh is cloned to the target mesh.

Previous parameterization mapping methods are almost applied to CAD model, texture mapping, remeshing and mesh morphing. Therefore these parameterization methods are generally complicated and unsuitable for our system. It needs only right 2D topological information by projection. It is achieved from projecting directly the vertices of stamped mesh onto the hypothetical plane P (see Figure 1.a), and especially suitable for the 3D information retrieval from 2D to 3D.

Here we introduce a very easy and effective parameter method named direct projective mapping. A round brush is used to make a mapping zone on the source mesh in screen space. The painted mesh zone is named mesh decal (left on Figure 1.a). We need to project the vertices of mesh decal onto a hypothetical plane P with maximum radius R which is used for texture mapping. In order to align the source mesh decal with the target mesh decal, the U, V coordinate lines are signed on the hypothetical plane P. After picking a center vertex vo, the plane P intersects the vertex vo and parallels with the screen. Of course, the normal np of P is vertical to screen.

The angle between projection edge $v_i' v_o'$ and U line denotes θ. The initial U, V coordinates of vi is $U_{io} = \| v_i' - v_o' \| \cos\theta$ and $V_{io} = \| v_i' - v_o' \| \sin\theta$ correspondingly. While considering the parameter space, the coordinates are necessarily rectified as $U_{io} = \left(\| v_i' - v_o' \| \cos\theta \right) / \sqrt{2}R$ and the V coordinate is $V_{io} = \left(\| v_i' - v_o' \| \sin\theta \right) / \sqrt{2}R$. Finally the para-meterization is translated by (0.5, 0.5), so that the "mapping zone" inscribed in the stamped mesh decal lies at [0,1]×[0,1] in parameter space. Direct projective mapping parameter method can accurately capture the center vertex of mapped mesh in 2D parameter space. Compared with the discrete exponential map algorithm[10] depending on the normal attached to every mapped vertex, which leads to center vertex position and projective direction drift in parameter space, our algorithm can be fit for followed editing operation.


(a)


(b)

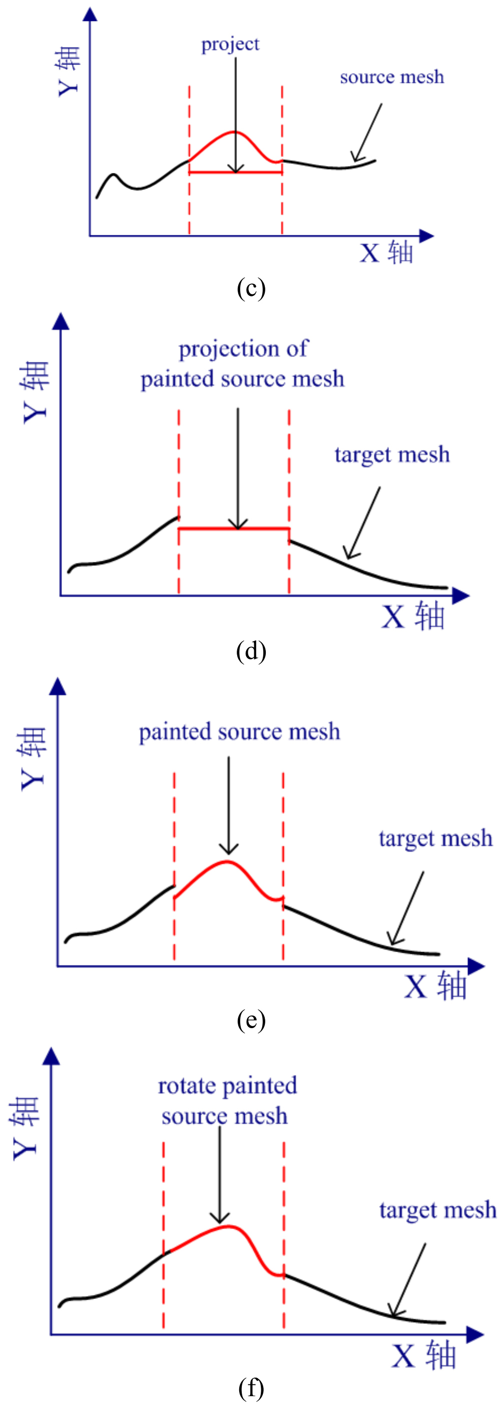


Fig 1. Mesh is projected and information retrieves from coplanar mesh, (b) is parameterization mapping of stamped mesh, (c–f) are 2D presentation of mesh editing process.

3. Image Copy-paste on the Gpu

By parametric mapping, the corresponding relations between source and target mapping zone are constructed by U,V parameter space(Fig. 1.b). Once mapping zone is stamped and direct projective mapping completes both in source and target mesh, ROI need be selected to copy from source mesh to target mesh. Here all 3D information will be ignored. Because of shape difference between two models, 3D vertex coordinates can't be stitched to target mesh directly. Otherwise, there will be a big gap in the stitched zones.

We consider to transmit the 2D mesh topological information and U, V parametric data in some format between two meshes. The triangular face index identities are assumed to color stored into image texels for pixel shader, and their barycentric coordinates are computed for GPU vertex shader. When user renders on the source mesh by round brush, the projective mesh of the ROI are copied to target mesh position (see Fig. 1.d). CPU is used to manage frame buffers and create the list of face indices.

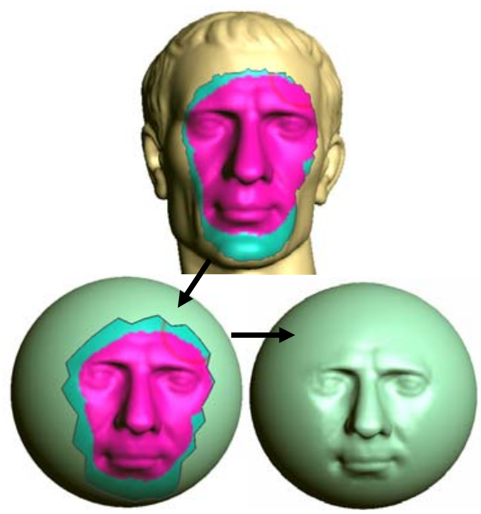


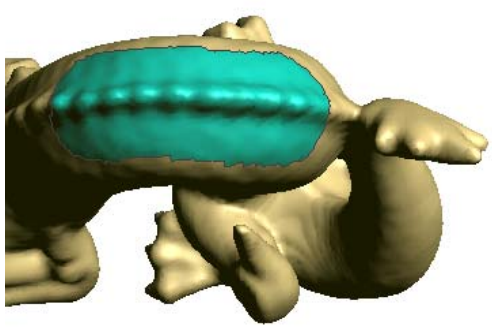
Fig 2. Render ROI of man face onto sphere mesh.

Now the whole source and target mapping zones with U, V parameters lying at [0,1]×[0,1] are loaded into two frame texture buffers. During the process of rendering, the position of round brush is caught and transformed into the GPU frame buffer and the CPU keep the vertex offset from the projection of the painted source mesh. The face index identities and barycentric coordinates of the source ROI as image texels are copied to target buffer texture to replace the corresponding section. However, gap between source ROI data and residual target data is preserved for further stitch (see Fig. 1.e). Here, the copied mesh in target mesh is coplanar without effective 3D shape information. At the same time, we need to get ride of gap between painted mesh and target mesh.

During the process of rendering ROI onto target mesh (see Figure 2), the surrounding of pasted region need keep continuity and smooth. The 3D coordinate information is retrieved from the preserved vertex offset based on the projection of the painted mesh enclosed by target mesh. By the conforming Delaunay triangulation, the gap is repaired. Vertices and segments on the boundary are collected and triangularized to meet constraints on the minimum angle and maximum triangle area.

4. Experiments and Discussion

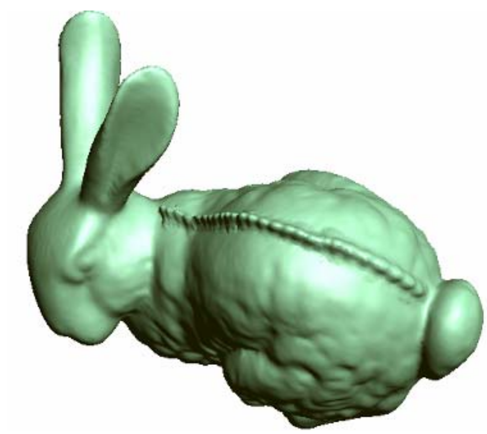
We implemented our system by the proposed approach on a machine including Intel Core I5-2520M, 2.5GHZ CPU with 2.0GB RAM and NVIDIA Geforce 520M GPU. Though Geforce 520M is low performance, the mesh cloning brush operation is completed in real time but for parameterization of stamped mesh. In order to confirm the quality of the our method, we provide comparisons against the mesh cloning system such as GeoBrush [7] which relies on cage meshes to perform mesh copy-paste and Meshmixer06[6]. Fig. 3(b), 4(c) show our mesh cloning algorithm applied in different mesh models. Fig. 4(c,d,e) demonstrate our method comparision against GeoBrush and Meshmixer06. Fig. 4(d) is GeoBrush's clone brush paints on different source mesh. The excessive distortions on the pasted region are produced because of automatic generation of cage meshes. Without constructed correctly cage meshes, the Green coordinates deformation will be ineffective. Fig. 4(e) shows the mesh cloning using Meshmixer06 software.



(a) paintings on the Dragon

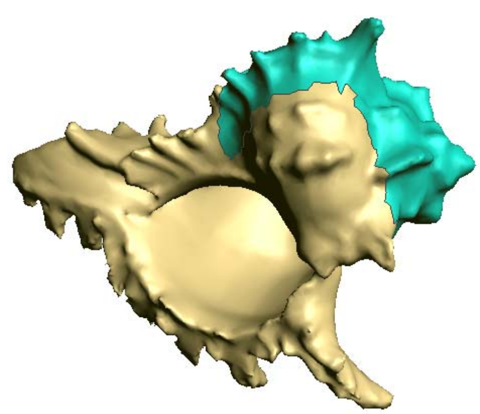
Other than the good performance of direct projective mapping mesh cloning, another key feature of our method is that they allow users to copy arbitrary irregular meshes features between two 3D modes. They also can be applied to mesh repair and hole filling when reconstructed from point cloud.

In our current system, the boundary of stamped mesh must have a single loop. The selected arbitrary irregular meshes for copy mush that exists on a larger and flat region, otherwise the direct projective mapping will be ineffective. In future work, we would like to alleviate this restriction by appending a flat large mesh for mapping. Although our method is powerful and flexible, they do not guarantee C1 continuity between stitched mesh components. In our experiments, C1 continuity commonly leads to ugly cloning with non-manifold mesh.

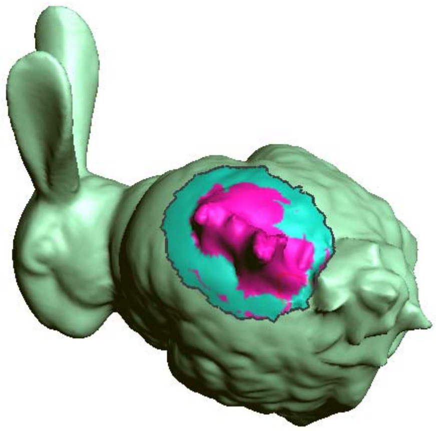


(b) copied to Bunny

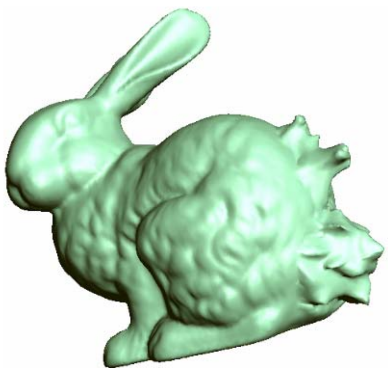
Fig 3. cloning the back mesh of Dragon to Bunny



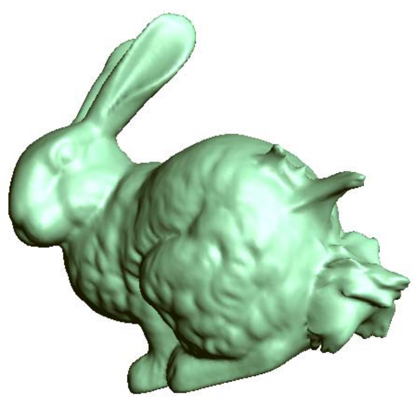
(a) paintings on the trumpet shell



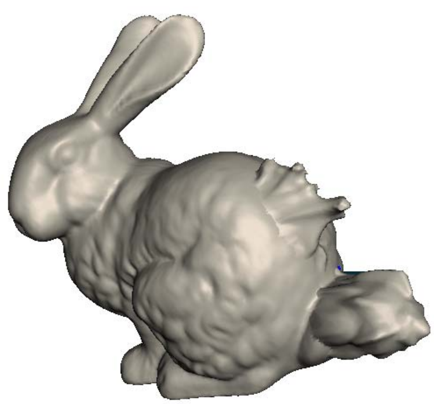
(b) copied to Bunny



(c) our cloning method



(d) Geobrush cloing method[7]



(e) Meshmixer cloing method[6]

Fig 4. comparison with different methods

5. Conclusions

This paper presents a novel real-time mesh cloning method based on GPU texture copy-paste. By mapping mesh vertices in selected ROI of source mesh onto a 2D image using GPU frame buffer, the 2D topological information is abstracted into image texel and placed on apposite position of target mesh. 3D coordinate information of cloned mesh is retrieved by preserved vertex offset. It avoids deforming the cloned mesh using cage mesh. Our approach can copy arbitrary irregular meshes features between two 3D modes.

Future works involve extending the proposed method to robust selection in arbitrary mesh region. In the future, we will focus on local parameter-rization mapping method which can be projected on a hypothetical plane more effectively for both image texture and mesh. Furthermore, the system's interface will be improved to copy image onto mesh forming texture mapping by cloning brush.

Acknowledgements

This research was supported in part by National Science Foundation in China (No. 10901147), Zhejiang Provincial Natural Science Foundation of China (No.Y1100018, LY12A01028) and Science & Technology Planning Projects of Zhejiang Province of China (2011R09015).

References:

[1] O. Sorkine , D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, H.P. Seidel, "Laplacian surface editing", *In Proceeding of the Eurographics Symposium on Geometry Processing 2004*, New York, pp. 179-188.

[2] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossl, H.P. Seidel, "Differential coordinates for interactive mesh editing", *In Proceedings of Shape Modeling International 2004*, Los Alamitos, pp. 181-190.

[3] K.C. O. Au,C.L. Tai, H. Fu,L.G. Liu, "Mesh editing with curvature flow Laplacian", *In Proc. of Eurographics Symposium on Geometry Processing 2005* , Vienna, Austria, pp. 191-199.

[4] Y.Z. Yu, K. Zhou,D. Xu, X. Shi,H.J. Bao, B.N. Guo, H.Y. Shum, "Mesh editing with Poisson-based gradient field manipulation", *In Proceeding of ACM SIGGRAPH 2004,* pp. 644-651.

[5] M. Eyiyurekli, D. Breen, "Interactive free-form

level-set surface-editing operators" , Computers & Graphics, Vol.34, No.5, 2010, pp. 621-638.

[6] R. Schmidt, K. Singh, "Mshmixer: an interface for rapid mesh composition", *In Proceeding of ACM SIGGRAPH 2010*, talk.

[7] K. Takayama, R. Schmidt, K. Singh, T. Igarashi, T. Boubekeur, O. Sorkine, "GeoBrush: Interactive Mesh Geometry Cloning", *Computer Graphics Forum*, Vol.30, No. 2, 2011, pp. 613-622

[8] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-or, D. Lischinski, "Coordinates for instant image cloning", *ACM Transaction on Graphics*, Vol. 28, No. 3, 2009, 67:1-67:9

[9] J. Mccann, N.S. Pollard, "Real-time gradient domain painting", *ACM Transaction on Graphics*, Vol. 27, No.3 ,2008, pp 93:1-93:9

[10] R. Schmidt, C. Grimm, B. Wyvill, "Interactive decal compositing with discrete exponential maps", *ACM Transaction on Graphics*, Vol. 25, 2006, pp. 605-613

Dr. Guiping Qian received the Ph.D. degree in computer science and technology from Zhejiang University, in 2008. Currently, he is an Associate Professor at Zhejiang University of Media and Communica-tions. His interests are in computer graphics, CAD & mesh processing.

Dr. Yue Wang received the Ph.D. degree in Mathematics from the Zhejiang University, in 2006. Currently, she is a Associate Professor at China Jiliang University. Her research interests include Differential Geometry and Digital Geometric Processing.