Simulation Ready Anatomy Model Generation Pipeline for Virtual Surgery

Abstract

For surgery simulation application, a high quality anatomical model is very important not only for rendering but also for physics simulation. Many commercial 3D human anatomical models on the market can only be used for visualization rather than physics simulation because of the non-manifold geometric degeneracies (such as self-intersection, non-watertight, noise, inter-penetrations etc.). In this paper, we proposed a simulation ready model generation pipeline which can convert a non-manifold polygonal surface mesh into a degeneracy free surface mesh (**simulation ready state**) while preserving the original model's surface parameterization attribute. Our pipeline includes two stages. The first stage is a voxelization and remesh based simulation ready model generation pipeline which can keep the shape of original 3D surface model meanwhile eliminate the ill shaped and degenerated polygons. The second stage is the main contribution of this paper. A cutting based surface mesh parameterization transfer algorithm is proposed which can transfer the original surface parameterization (UV mapping especially the UV seam) to the simulation ready model. It solves the parameterization space distortion problem in state-of-art tools when performing surface parameterization transfer and result is compared.

Keywords: simulation ready model, surface parameterization transfer, cutting, voxelization, remesh

1 Introduction

A realistic 3D anatomical model is of great importance to medical applications such as surgery simulation. In the current market, there are a lot of 3D human anatomical models which often includes complete set of human anatomies such as muscles, organs, skeletons and nerve system etc. Those models are mostly created by 3D artists who work under the guidance of medical practitioners or human anatomy structure books. However, when artists create those models, they focus more on the visual appearance. To achieve complex shapes and structures especially for junior artists, they may use irregular primitives to approximate the shape of the real anatomy structure. Due to the complexity of anatomic structure, non-manifold geometric degeneracies (such as self-intersection, non-watertight, noise, inter-penetrations etc.) are always inevitable during the modeling process. Although such artifact will not influence the rendering effect, it makes the anatomical model can not be used for continuum mechanics based physics simulation.

In surgery simulation, continuum mechanics based soft body simulation is widely used which requires the anatomical model having a solid structure (not empty inside). Tetrahedralization is a popular way to solidify surface mesh into solid structure but it requires the surface model to be self-degeneracy free and watertight. If the model have those properties and can be directly used for tetrahedralization to generate the solid structure object, we call it **simulation ready model** or a model in **simulation ready state**.

For the development of surgery simulator, using existing 3D anatomical models on the market rather than modeling everything from scratch is a cost saving choice especially for small companies or research groups which have inadequate support from professional digital artists. To convert existing anatomical models into a simulation ready model, manual fixing using 3D content creation tools is a tedious and time consuming process. An automatic conversion pipeline, which can be treated as a black-box and does not influence traditional artistic pipeline, will liberate artists from tedious works and improve the working efficiency. During the conversion process, there are mainly two challenges. The first is how to generate a simulation ready model which can efficiently get rid of the ill shaped and degenerated polygons in original model without manual fixing. The second is how to transfer the original mesh's surface parameterization attribute to the newly generated simulation ready model without distortion.

In this paper, we proposed an automatic pipeline that converts the poor quality anatomical model into simulation ready model while preserving the original model's surface parameterization property. The pipeline is composed of two stages. In the first stage, the original mesh is sent to a voxelization and remesh based pipeline which can keep the shape of original 3D surface model but eliminate the ill shaped and degenerate polygons without influencing existing artistic pipeline. In the second stage, a cutting based surface mesh parameterization transfer algorithm is proposed which can transfer the original surface parameterization (UV mapping) to the simulation ready model without distortion in the parameterization space, which is the main contribution of this paper.

2 Related Works

For the anatomy modeling, the ideal pipeline is to obtain 3D anatomy model from patient specific CT or MRI data. However, the labeling of the interested anatomy area from CT or MRI data is normally manual based and inaccurate. Digital modeling based on coarse CT and MRI data is popular in surgery simulator development but the polygon quality of the mesh normally are not good, which will influence the simulation accuracy and stability. There are a lot of commercial 3D human anatomical models on the market but most of them can only be used for visualization rather than physics simulation because of poor mesh quality and non-manifold structure.

The process of optimizing a poor quality non-manifold 3D surface model into a simulation ready model is a geometric processing problem. Traditional geometric processing techniques such as split, collapse, fuse operation on vertex, edge, face etc [1] can be helpful in fixing the geometric degeneracies but they often require the human intervention to get desired result. A practical framework for the automatic generation of finite element ready meshes based on subject-specic segmented MRI data is proposed in [2]. They resolves the noise and self-intersections mesh artifact by using surface smoothing and region grow based method. However, their solution only limited to triangular mesh. They didn't proposed any solution on other artifacts such as mesh degeneracy, non-closed mesh etc. In fact, instead of working on the trivial polygon representation level, those non-manifold geometry optimization problems can also be solved from the perspectives of voxel representation, which can facilitate the detection of self-intersections and other degeneracies using signed distance field (SDF). It is non-trivial to get rid of the non-manifold geometry by performing the topological operations (such as union, intersection and difference etc.) based on the SDF. The volumetric data can be represented on Cartesian, unstructured, Octree grid [3]. The Cartesian grid is convenient for fast interpolation, level set schemes such as marching based method [4], which has been widely used in game and VFX industries[5].

After the voxelization step, an isosurface mesh can be extracted based on the SDF. However, the original source mesh's surface parameterization information will be lost in the isosurface mesh as the topology of the new mesh is different from the original one. To preserve the source mesh's surface attribute, the transfer of the surface properties between the source and the target mesh is the key technique during this process.

Transferring surface properties from source mesh to target mesh is an important research topic in computer graphics, which includes detail synthesis [6], shape analysis[7], texture synthesis[8] and surface editing [9] etc. In our case, the input mesh is a non-manifold geometry with self-intersections and other degeneracies, the output target mesh is a high quality simulation ready polygon model. To match the original mesh to the newly generated target mesh, finding correspondences between two surfaces [10] is needed before transferring surface properties. The classical shape correspondence applications involves two steps: shape alignment [11] and feature matching[12]. In our case, the target mesh is generated from the source mesh. The target mesh has already preserved the shape of the original mesh but with different polygon representation so that shape alignment step is not necessary. We only need to match the feature of source and target mesh, and transfer the texture color and surface parametrization.

For the texture color transfer problem, there are mainly two general approaches. The first one builds a common parameterization which is used as a mapping from the source surface to the target surface [13]. The smoothness of the mapping will affect the scale of local distortion. This method will preserve large scale pattern in textures but suffer from local scale distortion especially when the shape variation is large between source and target mesh. Texture synthesis, as the second approach, is a process of constructing large images from exemplar by preserving its structural content and detail feature. The traditional methods in texture synthesis is a process of pixel-based neighborhood-matching which finds the best matching pixel between partially synthesized neighborhoods and exemplar neighbours [8]. This method is able to reproduce the small scale details of the texture but can not well preserve the large scale patterns. The above methods will inevitably produce distortion which are not suitable for our case.

Beside the distortion problem, texture discontinuity across the UV seam (chart boundary) is another issue when transferring texture color. UV seam of the texture atlases will produce discontinuities artifact because the neighbouring points on the mesh surface may be far away from each other in the parameterization space (UV space). There are many works [14][8][15] on solving the problem of UV seam discontinuity. However, those methods focus on healing the UV discontinuity problem based on the original mesh's polygon representation but they are not suitable for our case because we focus on transfer surface parameterization property between two surfaces of different polygon representation. Instead of solving the UV discontinuity on original mesh, we need to preserve the source mesh's UV parameterization for the target mesh and ensure the original UV parameterization causes no artefact (UV space element stretch around UV seam) on the target mesh.

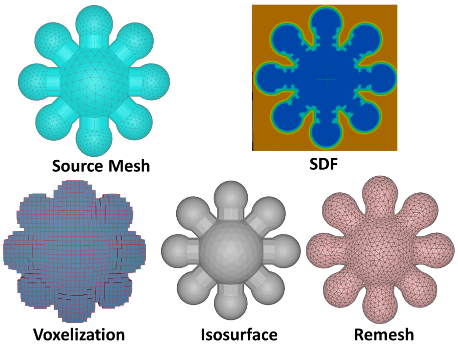


Figure 1: First stage of voxelization and remesh based simulation ready model generation pipeline.

Compare to the existing methods, most of current texture transfer techniques can only maintain the texture color and inevitably cause distortion to some extent. In our case, we need the target mesh not only inherit the source mesh's texture color but also the parameterization because the parameterization information may be needed for other purposes, such as bleeding, burning effect etc in surgery simulation. In modern digital content creation tools such as Maya, Houdini, Blender etc., all support proximity based attribute transfer however this method will cause artifact in surface parameterization space (see more details in section 5). In this paper, we solve the texture transfer problem using a cutting and remesh based technique which retains the original mesh's surface property and eliminates the distortion problem during the transfer process.

3 Voxelization Based Mesh Optimization

The common way of constructing the volume representation is to convert the explicit geometry representation into a signed distance function $\phi(\mathbf{x})$. The signed distance function computes the minimum distance from a given point $\mathbf{x}$ to the boundary ($\phi(\mathbf{x}) = 0$) on the mesh, with a positive value sign outside the domain and negative value sign inside the domain.

For the input non-manifold geometries with self-intersections and degenerate elements, they are firstly converted from the explicit polygon presentation into a signed distance function $\phi(\mathbf{x})$ (level set) using scan conversion [16][17]. The gradient of the signed distance function can provide geometric information. The resolution to sample the SDF will determine how well the volumetric data represent the shape of the input polygon geometry. In the center of each voxel data, a distance field sampled value is stored.

After building the signed distance function for input geometries (see result in algorithm procedure description figure 1), the SDF can easily get rid of self-intersections and degenerate elements of the input geometries (called *source mesh*) using the simple and fast topology operations between distance fields such as the union, difference and intersection.

After the topology modification, the volumetric representation of the desired object's shape is obtained. Then the volumetric data needs to be converted back into the polygon mesh, which is a process of surfacing the volumetric data along the specified iso value. However, there are two challenges for the output isosurface. Firstly, the quality of the isosurface is dependent on the resolution of the voxel size. Secondly, the new mesh will lost the surface parameterization information (UV coordinate) in the original mesh.

For the first challenge, in order to maintain the source mesh's shapes and details of the input polygon geometries, small voxel size is often used, which normally will create high density polygons (see the isosurface in figure 1). Here we apply remesh technique to reduce the polygon count while preserve isosurface's shape. This remeshed model can be used as a simulation ready model, we call it as the *target mesh*

For the second challenge, unwrapping the target mesh and creating new surface parameterization for it is quite inefficient which is no easier than fixing mesh degeneracies piecewisely. In the following part, a cutting based surface parameterization transfer method is proposed which can transfer the source mesh's surface parameterization information to the target mesh without introducing distortion.

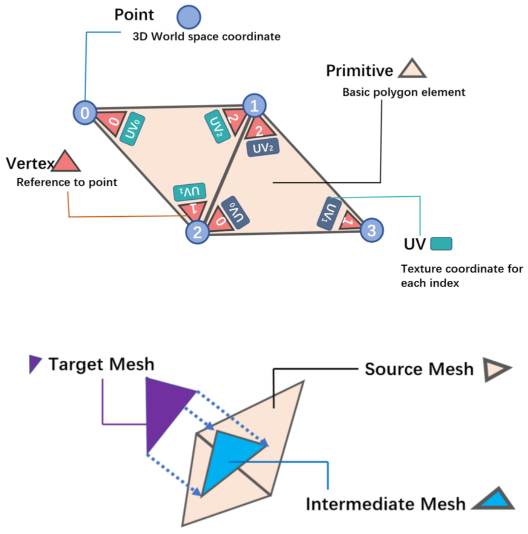


Figure 2: Basic Notion of Geometry

4 Parameterization Transfer

4.1 Basic Notion

Before introducing our method, some basic geometric notions are firstly introduced. As can be seen in figure 2, point is simply a point in world space as defined by a 3D coordinate. Vertex is an integer reference (index) to the point. Primitive consists of a group of vertices that indicate the basic element of the mesh. The point can be shared among primitives but the vertices are unique in each primitive which means a point can be referenced by several different vertices in different primitives. In each primitive, each vertex has its own texture coordinate (UV coordinate). The original input mesh is defined as the *source mesh* and the remeshed isosurface generated from the voxelization step is defined as *target mesh*. The projection of the target mesh onto the source mesh's surface is defined as the *intermediate mesh* which is used to transfer attribute from source mesh to the target mesh. For each vertex on the intermediate mesh, it lays inside a primitive on the source mesh. This primitive is defined as the *host primitive* for intermediate mesh's vertex.

4.2 Method Overview

To transfer the surface parametrization (UV mapping) information from the source mesh to the target mesh, it is natural to transfer the surface attribute from the source model to the closest location on the target mesh. However, this method will also transfer the UV seam from the source mesh to the target mesh, which will result in the UV space stretch for the polygons that cross the UV seam on the target mesh. To solve this problem, our idea is to project the target mesh onto the source mesh and dissecting the target mesh based on how the intermediate mesh is dissected by the UV seam's corresponding edges (defined as *UV seam edges*) on source mesh, which can completely eliminate the UV space stretch artifact. Then use the newly generated dissected edge as the hard edges and feed the dissected target mesh into a remesher to generate the final good quality simulation ready model, which can completely inherit the surface parameterization of the source mesh without distortion.

4.3 UV Island Based Surface Parameterization Transfer

When performing the surface transfer operation, identify the location where the surface attribute is transferred from is important. Due to the shape of the target mesh is conform to that of the source mesh, it is natural to transfer the surface attribute from the source mesh to its closest location on the target mesh. The transfer process is performed by iterating the source mesh's vertices and piece-wisely transferring attribute to each vertex on the target mesh. For the i-th vertex on the target mesh, its corresponding point $\mathbf{x}_i$ is projected to the source mesh's surface (projection position $\mathbf{x}_i^p$). $\mathbf{x}_i^p$ is used as the intermediate for the transfer which means surface parameterization attribute is firstly transferred from source mesh to intermediate $\mathbf{x}_i^p$ and then this attribute is copied to its corresponding vertex on the target mesh. The transfer operation is based on barycentric coordinate. For each $\mathbf{x}_i^p$, a barycentric coordinate $\mathbf{w}_i$ can be computed from the *host primitive* it is projected onto, $\mathbf{w}_i \in \mathbf{R}^k$ where k is the number of vertices in this primitive (mostly $k = 3$ or 4). If k is larger than 4, the source mesh can be triangulated or quadrangulated to meet this standard. Then $\mathbf{x}_i^p$'s surface parameterization attribute ($\mathbf{uv}_i \in \mathbf{R}^2$) is computed by barycentrically interpolating the corresponding vertices' parameterization attribute on source mesh. Then assign $\mathbf{uv}_i$ to the corresponding i-th vertex on the target mesh.

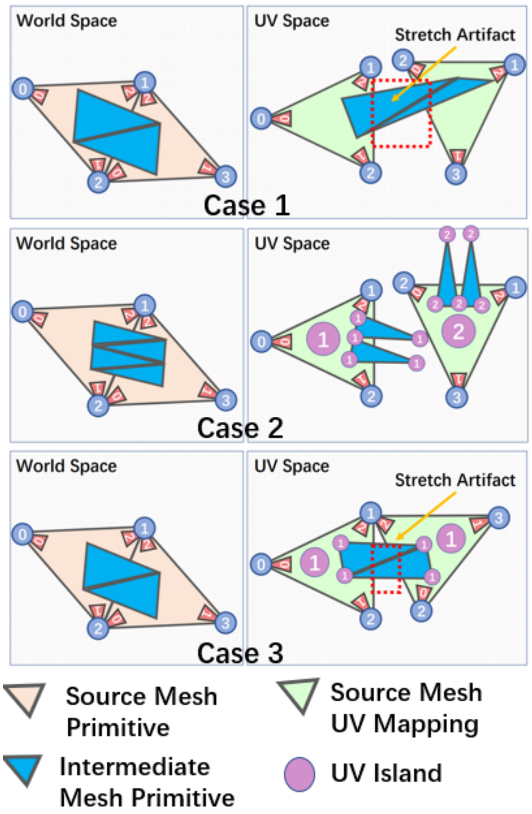


Figure 3: The UV space stretch artifact.

However, this method will generate artifact. As can be seen in figure 3 (case 1, World Space), the *intermediate mesh's* primitive may cross the source mesh's UV seam in the world space. If the UV seam separates the source mesh into several isolated UV islands, then the primitive that cross this UV seam will be stretched in the UV space, as can be seen in figure 3 (case 1, UV space). This problem is caused by the fact that the intermediate mesh's vertices in the same primitive belong to different UV islands, which means the neighbouring points on the intermediate mesh surface are far from each other in the parameterization space (UV space). When performing the barycentric interpolation for those cross UV seam primitives, their vertices will be interpolated to different UV islands which cause the distortion or stretch artifact.

To solve this crossing UV island artifact, ensuring each intermediate mesh's primitive is barycentrically interpolated into the same UV island will effectively alleviate this artifact. Before performing the barycentric based attribute transfer, which UV island each primitive should be interpolated into is needed to be identified. For the intermediate mesh, the UV island (κ_j) is firstly labeled for each vertex by assigning the host primitive's UV island ID to it. For the labeling of the primitives that cross the UV seam, we compute the area portion split by the UV seam and assign the largest portion's vertex UV island ID to all the vertices in this primitive, as can be seen in figure 3(case 2). Although this method may work for some models and artifact may not be obvious, it still generates zigzag UV pattern near the UV seam. However, if the UV seams do not split the mesh completely into two separate UV islands which means the area crossing the UV seam share the same UV island ID. The above method will still generate the UV space stretch artifact near UV seams, as can be seen in figure 3 (case 3).

In general, when transferring UV attribute, artifact is difficult to be avoided if the intermediate mesh's primitives cross the UV seam of the source mesh. To completely eliminate the UV space stretch artifact, there should not be any intermediate primitives cross the source mesh's UV seam. To achieve this, the intermediate mesh's primitives that cross the UV seams must be split or adjusted to align with the UV seam.

4.4 *UV Seam Based Primitive Cutting Transfer*

To split the primitives on the intermediate mesh that cross the source mesh's UV seam, intersection is tested between the primitives and the corresponding edges of the UV seam. If the intermediate primitive intersects with the UV seam edges, corresponding primitive on the *target mesh* will be dissected into refined primitives according to how the intermediate primitive is dissected by the UV seam edges. This process is called as UV seam cutting based parameterization transfer. The idea proposed in [18] is used which combines the vertex snapping with the element refinement to avoid small or ill shaped primitives. For the intermediate mesh's primitive which cross the UV seam, the polygon area is measured on both side of the UV seam. If the polygon area ratio is too large or too small, the vertex will be snapped onto the seam instead of being split.

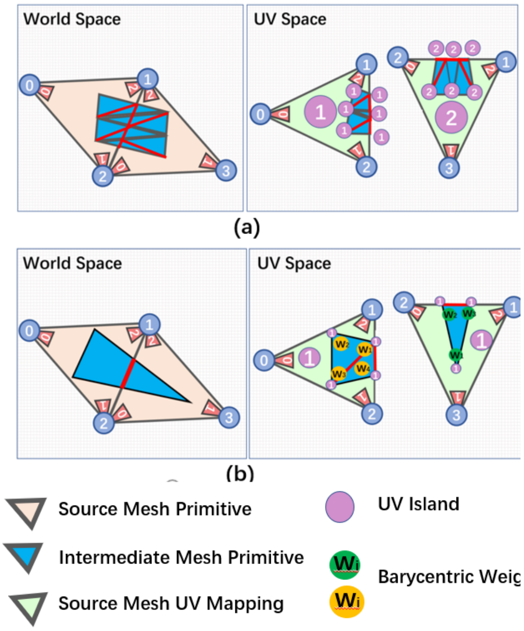


Figure 4: UV seam based primitive cutting.

After this operation, there will be no primitives crossing the UV seam. And then the UV island based surface parameterization transfer operation is performed (see section 4.3), there will be no artifact near the seam between two different UV islands, see figure 4 (a). For the seam that splits the same UV island, when computing the barycentric coordinates for the newly generated vertices that lay on the UV seam, we use the rule that the vertices in the same primitive (on the target mesh) should use the same *host primitive* (on source mesh) to compute the barycentric coordinate and transfer attribute, can be seen in figure 4 (b). If not use this rule, the vertices in the same primitive will receive attribute from two different primitives on the source mesh, resulting in UV space stretch again.

Then the remesh technique [19] is applied to optimize the shape of polygon by using the edge that lay on the UV seam as hard edge. The hard edge may be subdivided by remesher according to the user-specified edge length for remesher, but the hard edge shape will be preserved which can avoid the UV space stretch. Till now the 3D anatomy models with self-intersections and degenerate elements can be converted into a simulation ready model using this pipeline while preserving the original mesh's surface parameterization. The whole procedure of the cutting based surface parameterization transfer has been summarized in algorithm 1.

Algorithm 1 Cutting Based Surface Parametrization Transfer

```
Definition: target mesh ($\mathbf{T}$), source mesh ($\mathbf{S}$), barycentric coordinate ($\mathbf{w}$), uv coordinate ($\mathbf{uv}$).
procedure UVTRANSFER($\mathbf{T}$, $\mathbf{S}$)
    Project $\mathbf{T}$ onto $\mathbf{S}$ to get intermediate mesh $\mathbf{I}$
    for all primitive $\pi_i \in \mathbf{I}$ do
        if $\pi_i$ intersect with UV seam then
            Split $\pi_i$ into $\pi_i^k$ $(k = 1, 2)$
        if $\pi_i^k$ $(k = 1, 2)$ is ill shaped then
            Perform vertex snapping
        for all vertices $\mathbf{x}_j \in \pi_i^k (k = 1, 2)$ do
            Compute $\mathbf{w}_j$ for $\mathbf{x}_j$
            Interpolate $\mathbf{uv}_j$ using $\mathbf{w}_j$
            Transfer $\mathbf{uv}_j$ back to $\mathbf{T}$
        for all edge $\mathbf{e}_m \in \pi_i^k (k = 1, 2)$ do
            if $\mathbf{e}_m$ coincides uv seam then
                Mark $\mathbf{e}_m$ as hard edge
        Feed $\mathbf{T}$ and all $\mathbf{e}_m$ into remesher
        Return simulation ready model $\mathbf{T}$
```

5 Experiment and Results

5.1 Voxelization

The anatomy model used is a human kidney which includes many parts such as renal pelvis, adrenal gland, pyramid, artery and vein. The quality of this kidney anatomy model is fair for rendering. However, it does not meet the requirement of physics simulation because it includes the irregular and ill shaped polygons, self-intersection and not watertight etc., see figure 5.

The voxelization can be performed separately on each part of the source mesh or the whole mesh according to the requirement of simulation. Due to the fact that each part of the model may have different resolutions, perform voxelization on the whole mesh may cause the loss of detailed geometric features and change the source mesh's geometry group information which will influence the artistic pipeline. Perform voxelization on each part of the source mesh will capture the detail feature and make each part of the model simulation ready. For the purpose of better illustration, the experiment is only performed on one part of the mesh (renal pelvis). For other part of the mesh, the same operation can be performed. In figure 7, high resolution voxel can capture the detail of mesh well which can be seen from the SDF and corresponding isosurface. The quality of the isosurface will directly influence the final result of the remesher. When sending isosurface to remesher [19], the edge length of remesher can be fixed or adaptive. For the adaptive remesh, the qual-

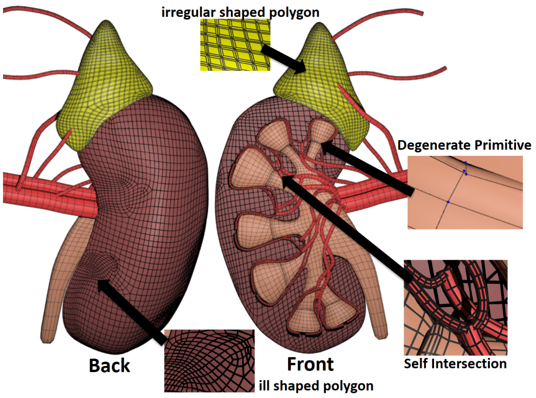
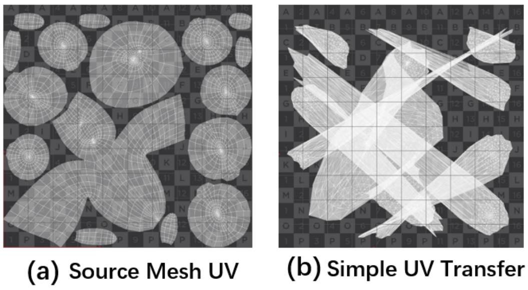


Figure 5: Illustration of the artifacts in the experiment model.



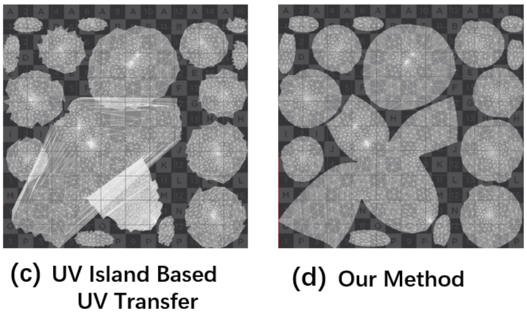


Figure 6: Surface attribute transfer results.

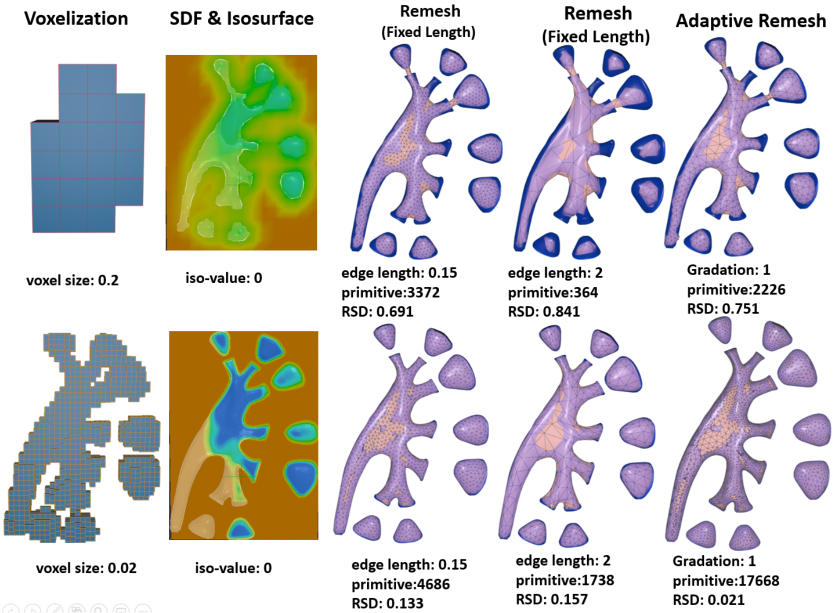


Figure 7: Voxelization based mesh optimization pipeline and remeshed results. In the remeshed result, the shape of source mesh (translucent purple layer) and remeshed model (solid pink layer) is compared.

ity of the result is controlled by the gradation [19] which means the rate that edge lengths are allowed to change from one primitive to another. The more accurate the isosurface is, the better the remeshed model will approximate the source model. How well the remeshed model approximates the source mesh is measured by relative surface distance (RSD). The surface distance (SD) between meshes is measured based on Hausdorff distance [20]. The RSD equals to SD divide the source mesh's average edge length. It can be seen from figure 7 that both fixed and adaptive edge length can well capture the shape of the source mesh even choose large edge length for remesher. Although the shape of the remeshed model is not completely the same as the source mesh, RSD of the high voxel resolution remesh result in figure 7 (lower row) shows that the distance between meshes is no more than 16 percent of source mesh's average length. Such small variance can satisfy the need of most virtual surgery simulator for training purpose. The subtle shape different can be solved using rendering technique like displacement mapping. It will compute the displacement from the target mesh to the source mesh along the local surface normal. When rendering, the target mesh will be displaced to the shape of source mesh.

5.2 *Surface Attribute Transfer*

It can be seen from figure 6 (b) that transferring the UV parameterization without using the UV island information will generate the UV space stretch and obvious object space artifact. This proximity based attribute transfer widely adopted by digital creation software such as Houdini, Maya. By taking into consideration the UV island information (Figure 6 (c)), the UV space stretch artifact can be alleviated but the zigzag artifact near the UV seam and stretch artifact still exist on the same UV island. In Figure 6 (d), it can be seen that our method can well eliminate the UV space stretch. Also, our method can preserve the original mesh's UV density without detail loss and distortion, as can be seen in Figure 8.

In figure 9, when applying our method on each part of the kidney model, a simulation ready model can be generated, which has good quality polygon discretization and completely inherit the source model's surface parameterization. Figure 9 also show the result of simulating the target mesh using finite element method [21].

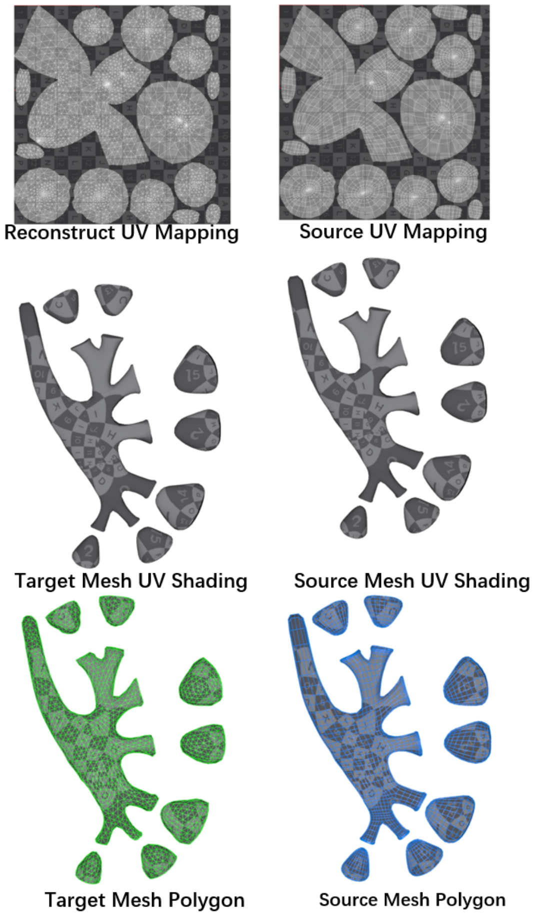


Figure 8: UV shading and polygon quality comparison.

6 Conclusion and Future Works

This paper proposed a simulation ready model generation pipeline which can convert a poor quality polygonal surface with non-manifold geometric degeneracies into simulation ready state while preserving the original model's surface parameterization attribute. The main contribution of this paper is a cutting based surface parameterization transfer algorithm which can be used to transfer the surface parameterization attribute especially the boundary information (UV seam) from the source mesh to the new target mesh. It solves the parameterization space distortion problem in state-of-art tools when performing surface parameterization transfer. As this paper mainly focus on transferring surface parameterization attribute between source and converted mesh. Conversion accuracy is still an issue especially for accuracy-oriented application. In the future, the first stage of the proposed voxelization and remesh based pipeline can be improved. Although the voxelization and remesh operation can maintain the shape of the source mesh, there are still differences between source and target meshs shape. Maximize the conversion accuracy will be of great significant.

References

[1] David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Cláudio T. Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing: A survey. *Comput. Graph. Forum*, 32(6):51–76, 2013.

[2] Pieter Peeters and Nicolas Pronost. A practical framework for generating volumetric meshes of subject-specific soft tissue. *The visual computer*, 30(2):127–137, 2014.

[3] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *ACM transactions on graphics (TOG)*, volume 21, pages 339–346. ACM, 2002.

[4] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[5] Ken Museth. Vdb: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics (TOG)*, 32(3):27, 2013.

[6] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James F O'Brien. Example-based wrinkle synthesis for clothing animation. *ACM Transactions on Graphics (TOG)*, 29(4):107, 2010.

[7] Aleksey Golovinskiy and Thomas Funkhouser. Randomized cuts for 3d mesh analysis. In *ACM transactions on graphics (TOG)*, volume 27, page 145. ACM, 2008.

[8] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis.

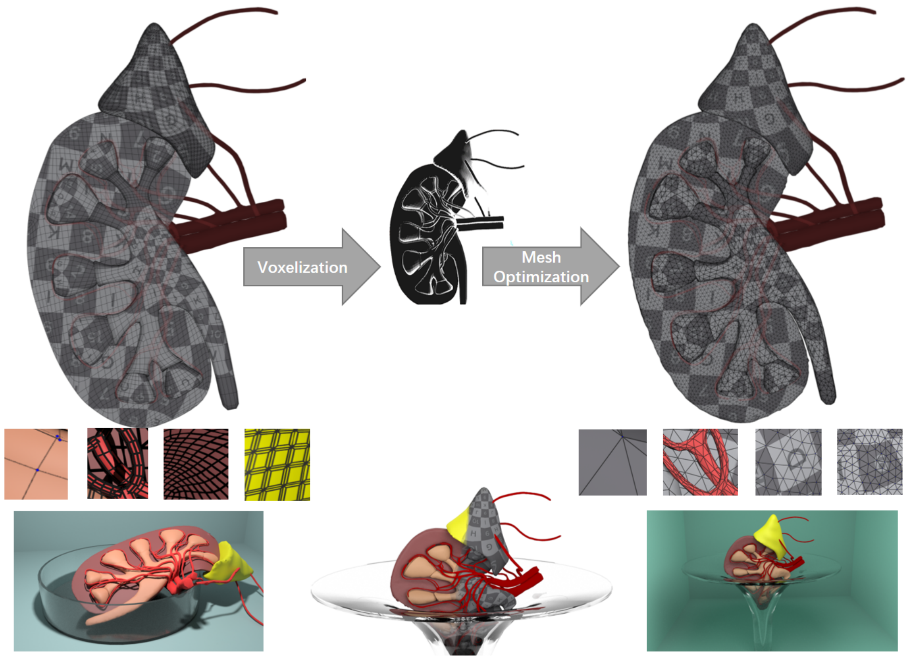


Figure 9: Kidney model conversion and the simulation result.

ACM Transactions on Graphics (TOG), 25(3):541–548, 2006.

[9] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.

[10] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

[11] *3rd International Conference on 3D Digital Imaging and Modeling (3DIM 2001), 28 May - 1 June 2001, Quebec City, Canada*. IEEE Computer Society, 2001.

[12] Yusuke Yoshiyasu, Wan-Chun Ma, Eiichi Yoshida, and Fumio Kanehiro. As-conformal-as-possible surface registration. In *Computer Graphics Forum*, volume 33, pages 257–267. Wiley Online Library, 2014.

[13] Marc Alexa. Recent advances in mesh morphing. In *Computer graphics forum*, volume 21, pages 173–198. Wiley Online Library, 2002.

[14] Nicolas Ray, Vincent Nivoliers, Sylvain Lefebvre, and Bruno Lévy. Invisible seams. In *Computer Graphics Forum*, volume 29, pages 1489–1496. Wiley Online Library, 2010.

[15] Alla Sheffer and Konrad Polthier, editors. *Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June 26-28, 2006*, volume 256 of *ACM International Conference Proceeding Series*. Eurographics Association, 2006.

[16] Byungjoon Chang, Deukhyun Cha, and Insung Ihm. Computing local signed distance fields for large polygonal models. In *Computer Graphics Forum*, volume 27, pages 799–806. Wiley Online Library, 2008.

[17] Ben Houston, Michael B Nielsen, Christopher Batty, Ola Nilsson, and Ken Museth. Hierarchical rle level set: A compact and versatile deformable surface representation. *ACM Transactions on Graphics (TOG)*, 25(1):151–175, 2006.

[18] Denis Steinemann, Matthias Harders, Markus Gross, and Gabor Szekely. Hybrid cutting of deformable solids. In *Virtual Reality Conference, 2006*, pages 35–42. IEEE, 2006.

[19] Per-Olof Persson. Mesh size functions for implicit geometries and pde-based gradient limiting. *Engineering with Computers*, 22(2):95–109, 2006.

[20] Nicolas Aspert, Diego Santa-Cruz, and Touradj Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings. IEEE international conference on multimedia and expo*, volume 1, pages 705–708. IEEE, 2002.

[21] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 20:1–20:50, New York, NY, USA, 2012. ACM.