EG UK Theory and Practice of Computer Graphics (2007)
Ik Soo Lim, David Duce (Editors)

Seamless Mesh Stitching Using Curve Approximation

Like Gobeawan[1], Shuhong Xu[2] and Stephen John Turner[1]

[1]Nanyang Technological University, Singapore
[2]Institute of High Performance Computing, Singapore

Abstract

Stitching two mesh partitions at their boundaries poses some challenges, especially when a visually seamless result is desired. In general, direct stitching of mismatching mesh boundaries does not readily produce a seamless result, unless some geometric measures are considered in the process. In this paper, a novel stitching technique for mesh partitions with mismatching boundaries is proposed. Using the boundary information of the mesh partitions, smooth curves are constructed across the partition boundaries and new vertices along the stitch path are generated to replace the mismatching boundary vertices. Several seamlessness criteria are considered in the construction of the algorithm to ensure a seamless stitch result.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling I.3.m [Computer Graphics]: Miscellaneous

1. Introduction

One fundamental task in the editing of polygonal meshes is the mesh stitching operation. Stitching two mesh partitions together is useful in geometry processing applications. For example, it is used in joining two model components to form one new model in rapid modeling applications. In other instances, it is employed in the stitching of two partitions which have been previously extracted from a model and processed separately.

A typical situation involves the real-time visualization of a large scale model on a computer system with limited resources. In order to achieve this, the large model needs to be simplified. The simplification process cannot readily take place in a system with limited memory and some out-of-core algorithms can be used to partition the large model into smaller portions, as suggested by Ho *et al.* [HLK01]. These partitions can then be processed sequentially or in parallel. After simplification, the boundaries of the partitions contain mismatching vertices, and mesh stitching is required to merge the individual portions. When visual quality is desired, seamless stitching becomes important.

In general, current mesh stitching approaches emphasize either the geometric accuracy or speed in mesh stitching. This paper attempts to formulate a seamless mesh stitching method featuring a balance between geometric accuracy and speed.

The objective of this paper is to develop an automatic algorithm to stitch two mesh partitions seamlessly at their corresponding boundaries. Here, the notion of seamlessness refers to the relatively smooth and visually undetectable stitched boundaries. We propose a seamless mesh stitching algorithm using a curve approximation technique, based on some seamlessness criteria.

Some of the terms used in this paper are defined as follows. Stitching is defined as the process of joining two mesh partitions together at their corresponding boundaries, while stitching path is the loop at which the boundaries of two mesh partitions merge. Curve approximation refers to the procedure of approximating some smooth curves across the partition boundaries. The terms *mesh* and *polygonal model* will be used interchangeably in this paper.

2. Literature Review

The concept of mesh stitching is closely related to surface reconstruction and mesh fusion. Surface reconstruction generates 3D surface patches from a group of unorganized points, and mesh stitching is employed to merge these patches together. On the other hand, mesh fusion aims to combine two

meshes and their intersecting boundaries must be smoothly connected via a stitching and blending process.

The surface reconstruction work by Meyer *et al.* [MSS92] involves using a surface fitting technique to fit smooth surface patches to a triangulated mesh. Another surface reconstruction technique, proposed by Turk and Levoy [TL94], introduces a mesh zippering concept to stitch two mesh partitions by clipping their boundaries against each other, forming small triangles at the overlapping part, and removing the small triangles. While the latter work is computationally less expensive, it does not guarantee smoothness as does the former.

Existing work in mesh fusion such as [KSMK99], [BMBZ02], [YZX*04], [JLW*06] and [FKS*04] produce smooth blending of two mesh models by modifying the mesh in the locality of the intersecting boundaries. Such approaches modify more than the two immediate boundary layers of the mesh partitions to achieve visual blending at the stitched region at the cost of geometric accuracy.

In our work, instead of using a computationally costly surface fitting approach, a curve approximation technique is developed for the mesh stitching. It is mathematically less complex as it involves a one-dimensional calculation compared to the surface fitting approach, which requires a two-dimensional calculation. Smoothness across the stitched path is not compromised by using curve approximation because seamlessness criteria are considered in the selection and modification of the boundary vertices, as elaborated in Section 4. Moreover, to preserve the polygon distribution of the mesh partitions, we minimize the modification to only the first layer of polygons along the each partition boundary.

3. Seamlessness Criteria

In principle, seamlessness can be defined as having a smooth transition property across the stitch path (from one boundary to the boundary of the other partition) so that the stitch path is not visible, without omitting the features (such as a sharp corner or steep curve) across and along the stitch path. Hence, a smooth curve can be freely constructed across the stitch path while each stitch vertex is formed independently of other stitch vertices along the stitch path (see Figure 1).

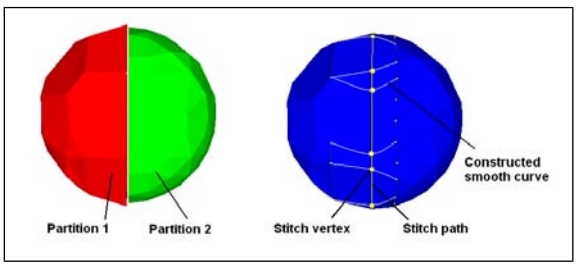

Figure 1: *Stitch vertices, stitch path, and smooth curves*

To achieve seamlessness, some criteria are proposed and considered. These criteria are related to the various geometric properties of each surface element and its sub-entities (i.e., vertex, edge and face), which are the important factors in achieving the seamless appearance. They are subsequently used in the construction of our proposed algorithm accordingly.

3.1. Stitch vertex normals

The normal of a vertex can be approximated as the average of the normal vectors of its corresponding faces. The normals of the vertices along the stitch path are not necessarily continuous relative to each other, but they should be continous with respect to their neighbor vertices beside the stitch path.

This criterion is considered in our algorithm at the curve approximation stage. The curves crossing the stitch path are constructed using the Bézier curve approximation. Each curve is constructed independently of the other curves.

3.2. Dihedral angles and edge lengths

For smoothness across the stitch path, dihedral angles of the directly connected edges across the stitch path are to be as small as possible. Similar to the vertex normal rule above, dihedral angles of the edges which are directly located on the stitch path are not necessarily related to each other.

Sharp features across the stitch path should be retained after the stitching. This can be achieved by prioritizing the feature edges in the selection of the vertex correspondence set which is used in approximating stitch vertices. The vertex correspondence stage of our algorithm is designed to retain sharp features across the stitch path, as explained in Section 4.2.

Edge lengths across the stitch path should be proportionally retained as well. The curve approximation stage of our algorithm, as explained in Section 4.3, determines the position of the new stitch vertices on the constructed curves with respect to the length ratio of the edge pairs by taking the mid-interval points of the curves.

3.3. Face distribution

Ideally, the area of the faces surrounding the stitch path should not deviate from the area of those in the original model without stitching. To minimize modification to the faces of the partitions, our algorithm limits the modification only to the first layer of boundary polygons from each partition.

To achieve a balanced number of faces surrounding the stitch path, the merged boundary vertices should be moderate in number. A reasonable number of vertices along the stitch path ranges between the numbers of boundary vertices of both partitions.

4. Proposed Algorithm

To comply with the seamlessness criteria above, we compose our algorithm as follows. Two single outermost boundary layers of two model partitions are extracted, and the vertices of these layers are grouped into sets by a correspondence rule. Each correspondence will be used to construct an approximating curve where a stitch vertex is estimated to lie on the curve. The stitch vertices replace the existing boundary vertices from both partitions, and the two boundary layers are retriangulated accordingly.

We assume that the input mesh partitions are two-manifold triangle meshes, each having one boundary loop to be stitched together. The proposed algorithm will stitch two meshes together by merging the boundary edges of one partition with the boundary edges of the other, forming a stitch path. The vertices on the stitch path are referred to as the stitch vertices, as illustrated in Figure 1. The location and orientation of the mesh partitions are assumed to be fixed before the stitching. The mismatching boundaries of both partitions should be relatively close to each other with relatively small overlaps or gaps. This is consistent with the fact that both partition boundaries are merged as the same loop in the joined model, so they should not differ extremely in terms of distance. The algorithm aims to stitch the surface smoothly across the boundaries, while the sharp features are naturally retained along the direction of the boundaries.

There are four stages in the proposed algorithm:

- boundary band identification,
- vertex correspondence,
- curve construction and stitch vertex approximation, and
- retriangulation.

4.1. Boundary band identification

A boundary band is a loop of boundary polygons of a mesh partition. Each boundary polygon contains at least one edge lying on the boundary of the partition. A boundary band (Figure 2) consists of a chain of boundary vertices (highlighted in red) and a chain of near-boundary vertices (highlighted in blue). To limit the change over the model after stitching, only the two boundary bands from the partitions will be modified.

4.2. Vertex correspondence

To construct one approximating curve across the stitch path, a set of vertex correspondence, consisting of four vertices, needs to be extracted. This process consists of two parts: boundary vertex correspondence and selection of the accompanying near-boundary vertices.

In the boundary vertex correspondence part, each boundary vertex of one boundary band is paired up with one or more boundary vertices of the other boundary band. A pair

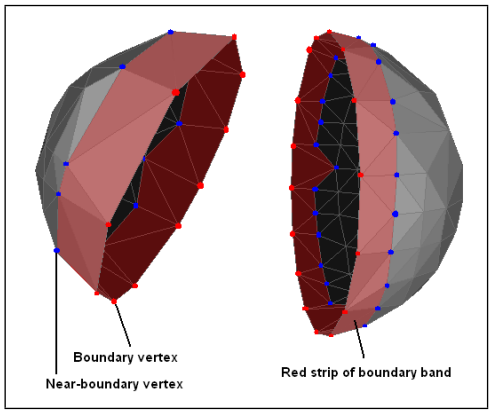

Figure 2: *Boundary band*

of boundary vertices will participate in one curve construction at the later stage.

It is favorable for the stitch path to contain many boundary vertices from both boundary bands to retain the existing information. Nevertheless, a stitch path having as many vertices as the total number of existing boundary vertices will produce overly crowded polygons around the stitch path, which we want to avoid.

A compromise is to have the stitch path to contain n_s vertices, such that $n_a \leq n_s \leq n_b$, where n_a is the number of boundary vertices of boundary band a with fewer boundary vertices, and n_b is the number of boundary vertices of boundary band b with more boundary vertices. We choose the most number of boundary vertices to be retained, that is $n_s = n_b$. To get a stitch path of n_s stitch vertices, n_s pairs of boundary vertex correspondence must be established.

Consequently, the boundary vertex correspondence procedure is as such that each boundary vertex in boundary band b corresponds to exactly one closest boundary vertex in boundary band a, and each boundary vertex in boundary band a corresponds to one or more consecutive boundary vertices in boundary band b. The pseudo-code of the procedure is shown in Figure 3.

Following the boundary vertex correspondence, each boundary vertex from both boundary bands is assigned one representative near-boundary vertex. This forms a set of vertex correspondence comprising four vertices: a pair of boundary vertices together with their assigned representative near-boundary vertices. In order to approximate one new stitch vertex, this set of vertex correspondence acts as the input for one curve construction in the following stage.

A boundary vertex is connected to one or more near-boundary vertices. The representative near-boundary vertex is chosen based on the dihedral angle of the edge connecting the near-boundary vertex to the corresponding boundary vertex (see Figure 4). The dihedral angle reflects the local fea-

B_a: boundary band which has n_a boundary vertices
B_b: boundary band which has n_b boundary vertices
$n_a \leq n_b$
d: $n_b - n_a$, look forward step
$L_a[1,2,...,n_a]$: ordered list of boundary vertices of B_a
$L_b[1,2,...,n_b]$: ordered list of boundary vertices of B_b (same direction order as L_a)
(The indices of L_a and L_b are cyclic in the mathematical operation below, i.e., $n_a+1=1$, $1-1=n_a$ and so on.)
V_C: ordered list of pairs of boundary vertex correspondence
$V_C = \{\}$
init: the index of element in L_b nearest to the $L_a[1]$

while $|L_b[init-1]-L_a[1]| \leq |L_b[init-1]-L_a[n_a]|$
    $init = init - 1$
repeat

for every element $L_a[t]$ where $t = 1,2,...,n_a$
    find one or more first consecutive elements $\{L_b[i] | i = init, init+1, ..., init+m\}$, $0 \leq m \leq d$ where $|L_b[i]-L_a[t]| \leq |L_b[i]-L_a[t+1]|$

    put the pairs $(L_a[t],L_b[init])$, $(L_a[t],L_b[init+1])$, ..., $(L_a[t],L_b[init+m])$ consecutively into V_C

    $init = init+m+1$
    $d = d-m$

Figure 3: *Procedure of establishing boundary vertex correspondence*

ture of the surface. The smaller the dihedral angle of an edge, the flatter is the surface, or the less distinct feature the surface has. As sharp features across the surface should be retained, the near-boundary vertex corresponding to the largest dihedral angle is chosen as the representative near-boundary vertex.

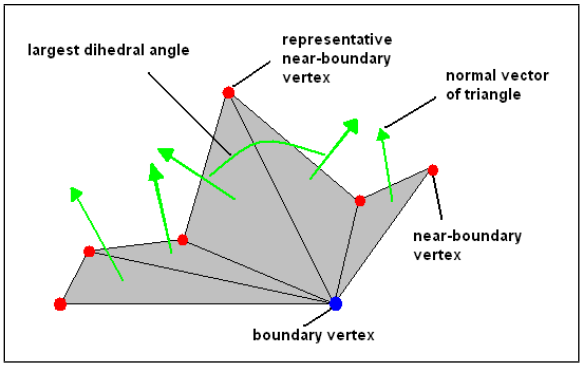

Figure 4: *Selection of representative near-boundary vertex for one boundary vertex*

Another method to select the representative near-boundary vertex of a boundary vertex is by calculating the weighted average of all near-boundary vertices corresponding to the boundary vertex. A larger weight is assigned to the near-boundary vertex which is connected to the boundary vertex by an edge with a larger dihedral angle.

For the special case of a lone boundary vertex of a polygon which has three boundary vertices (that is, the boundary vertex which has no near-boundary vertices), an imaginary near-boundary vertex is created. The imaginary near-boundary vertex can be assigned as the average of the other two boundary vertices of the polygon (see Figure 5).

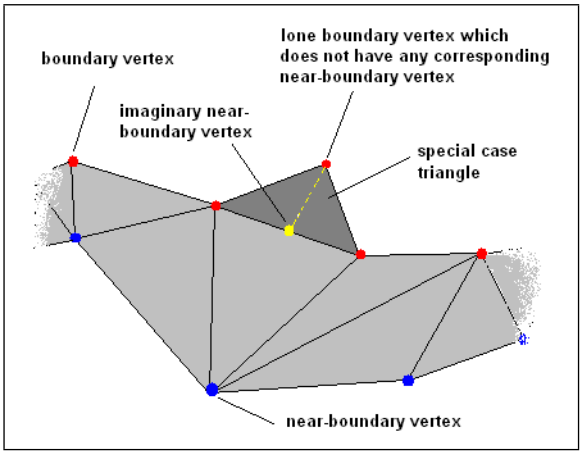

Figure 5: *Special case of lone boundary vertex and its imaginary near-boundary vertex*

4.3. Curve construction and stitch vertex approximation

Each set of vertex correspondence obtained in the previous stage is used to construct a fitting curve or an approximating curve, on which the stitch vertex is derived. In our implementation, we tentatively choose a simple Bézier curve approximation technique. The four vertices in each set of vertex correspondence are used as the control points of the Bézier curve (Figure 6).

The equation of the Bézier curve $s(t)$ is

$$s(t) = (1-t)^3 p_1 + 3t(1-t)^2 p_2 + 3t^2(1-t)p_3 + t^3 p_4$$

where t is the parameter of the curve such that $0 \leq t \leq 1$, p_1 is the representative near-boundary vertex corresponding to the boundary vertex p_2, and p_4 is the representative near-boundary vertex corresponding to the boundary vertex p_3.

According to the seamlessness criteria, the stitch vertex should be located at a point where its corresponding edges have a length ratio equivalent to the length ratio of the edge pairs p_1p_2 and p_3p_4. This is to preserve the edge length ratio

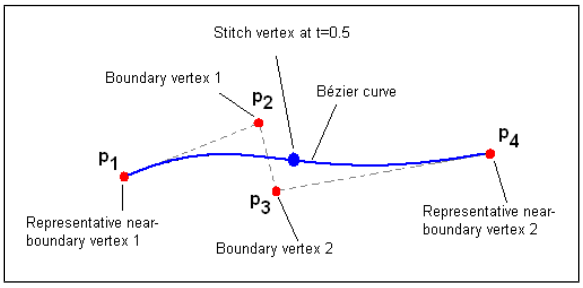

Figure 6: *Bézier curve construction and stitch vertex approximation*

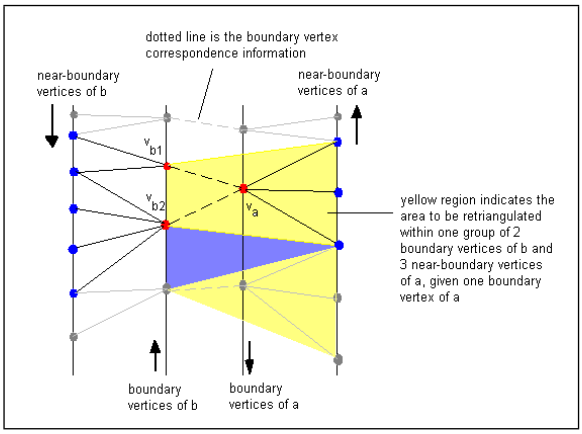

Figure 7: *Retriangulation of one boundary band*

after stitching. To achieve this naturally, the new stitch vertex is the mid-interval point of $s(t)$, where $t = 0.5$.

The stitch path (loop) is therefore formed by all the new approximated stitch vertices.

4.4. Retriangulation

In this stage, the retriangulation is to be done on both boundary bands along the stitch path. All boundary vertices are removed and replaced by the set of new stitch vertices. As the number of stitch vertices is equal to the number of boundary vertices of one boundary band, the retriangulation for this particular boundary band is equivalent to modifying the position of all its existing boundary vertices to the position of all new stitch vertices while retaining the topology of all the associated polygons.

For the other boundary band, the retriangulation involves removing all the existing polygons and filling up the gap. This gap can be viewed as a collection of sub-domains to be retriangulated separately, the boundaries of which can be identified using the information obtained in the vertex correspondence stage.

To illustrate this, we define v_a as a boundary vertex which belongs to the boundary with fewer boundary vertices, and v_{b_i} as the repositioned boundary vertices which correspond to v_a according to the vertex correspondence information. As shown in the top yellow area in Figure 7, we form a sub-domain from the vertices v_{b_i} and the near-boundary vertices corresponding to v_a. The sub-domain is then retriangulated internally. There are as many sub-domains as the number of boundary vertices on the boundary band with fewer boundary vertices. The remaining holes are single triangles by nature (see blue area in Figure 7).

5. Results and discussion

Analytically, the algorithm involves the processing of the vertices on the two single layers of boundaries of two partitions. If we approximate one unit of complexity of the algorithm as the construction time of one curve, the complexity of the algorithm is equivalent to the larger number of boundary vertices of one partition, i.e., O(n). In addition, the construction calculation of one curve is faster than that of one surface.

For the experiment discussed in this section, all models are represented in a normalized bounding box ranging in space from [-1,-1,-1] to [1, 1, 1]. The original models obtained from [Aim04] are each cut into two partitions. These partitions are then separately simplified using MeshLab [Cig06] to similar and or different resolution such that their boundaries are mismatching. The simplified partitions are then stitched back together to form a whole model.

A visually seamless stitching result can be observed in all the models. For models with a smooth surface across the stitch path, such as the horse model (Figure 9) and the camel model (Figure 10), we see that the vertex normals around the stitch path appear continuous, hence the surface is smooth. For models containing sharp edges along the stitch path such as the Homer model (Figure 13), the polygons forming the mouth are stitched seamlessly: sharp features along the stitch path are retained, yet the surface is smooth across the stitch path. Features across and along the stitch path on the bumpy sphere model (Figure 12) and the Armadillo model (Figure 11) are also well retained after stitching.

Figure 14 shows two Homer model partitions of relatively different resolutions. The stitching result shows the seamless stitch across the boundaries of the partitions.

Another version of the model, simplified directly from the original model so that it has the same number of faces as the stitched output is also provided for similarity comparison. Figure 8 shows the Hausdorff distance between the original model, the directly simplified model, and the stitched model produced by our algorithm. Note that the figures in brackets are the normalized Hausdorff distances with respect to the bounding box diagonal. We use the Metro tool [CRS96] for this evaluation.

3D Model (Original # of Faces, Simplified Version's # of Faces)	Original vs. Directly Simplified	Original vs. Stitched	Directly Simplified vs. Stitched
Horse (39698, 4010)	0.014748 (0.005009)	0.014748 (0.005009)	0.006955 (0.002367)
Camel (69092, 3490)	0.018510 (0.006060)	0.018510 (0.006060)	0.010932 (0.003580)
Bumpy sphere (68480, 3520)	0.011089 (0.002991)	0.019274 (0.005199)	0.021412 (0.005774)
Homer (10202, 5356)	0.006907 (0.002718)	0.013739 (0.005407)	0.014229 (0.005601)
Armadillo (345944, 17352)	0.005030 (0.001556	0.007947 (0.002458)	0.007348 (0.002273)

Figure 8: *Comparison of original models, directly simplified models, and stitched models*

The results in Figure 8 show that our stitched models are, in general, relatively close to both the original model and the directly simplified model. In the cases of smooth surfaces, such as the camel and the horse models, the stitched models are as good as the directly simplified models when compared with the original models (Hausdorff distance of 0.018510 for the camel model, 0.014748 for the horse model).

6. Conclusions

This paper presents an efficient seamless mesh stitching algorithm using a curve approximation technique. The presented algorithm produces a visually seamless stitch result with a balance of relatively good geometric accuracy and speed. The use of the curve approximation method for our mesh stitching algorithm, instead of the more computationally heavy surface reconstruction method, speeds up the mesh stitching process, while at the same time preserving as much as possible the geometric accuracy of the stitched model by abiding to the seamlessness criteria.

Potential applications of the proposed algorithm are mainly in the field of geometry modeling (for rapid modeling to join existing components to form new models) and mesh editing (joining preprocessed partitions, filling holes, mesh fusion).

Presently, our algorithm works by stitching the partitions smoothly across the stitch path while maintaining some sharp features across and along the stitch path. This can be extended to preserve more complicated features of the model across the stitch path such as folding. It can also be expanded to address the issue of self-overlapping possibility during the curve approximation and retriangulation stages in our algorithm for surfaces with local extreme variation of curvatures. Furthermore, the proposed seamlessness criteria can be refined to a more accurate and quantifiable measurement.

Acknowledgements

We would like to extend our appreciation to Su Yi for giving advice in the revision of this paper, and also to Chiet Sing and Zhang Yu who helped in the provision of various pointers for the project.

References

[Aim04] AIM: Aim@shape shape repository version 4.0. http://shapes.aim-at-shape.net/index.php, 2004. accessed on 28 February 2007.

[BMBZ02] BIERMANN H., MARTIN I., BERNARDINI F., ZORIN D.: Cut-and-paste editing of multiresolution surfaces. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM Press, pp. 312–321.

[Cig06] CIGNONI P.: Meshlab version 0.9. http://meshlab.sourceforge.net/, Epoch NOE, 2006.

[CRS96] CIGNONI P., ROCCHINI C., SCOPIGNO R.: *Metro: measuring error on simplified surfaces*. Tech. rep., Paris, France, France, 1996.

[FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. *ACM Trans. Graph. 23*, 3 (2004), 652–663.

[HLK01] HO J., LEE K. C., KRIEGMAN D.: Compressing large polygonal models. In *VIS '01: Proceedings of the conference on Visualization '01* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 357–362.

[JLW*06] JIN X., LIN J., WANG C. C. L., FENG J., SUN H.: Mesh fusion using functional blending on topologically incompatible sections. *Vis. Comput. 22*, 4 (2006), 266–275.

[KSMK99] KANAI T., SUZUKI H., MITANI J., KIMURA F.: Interactive mesh fusion based on local 3d metamorphosis. In *Proceedings of the 1999 conference on Graphics interface '99* (San Francisco, CA, USA, 1999), Morgan Kaufmann Publishers Inc., pp. 148–156.

[MSS92] MEYERS D., SKINNER S., SLOAN K.: Surfaces from contours. *ACM Trans. Graph. 11*, 3 (1992), 228–258.

[TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), ACM Press, pp. 311–318.

[YZX*04] YU Y., ZHOU K., XU D., SHI X., BAO H., GUO B., SHUM H.-Y.: Mesh editing with poisson-based gradient field manipulation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM Press, pp. 644–651.

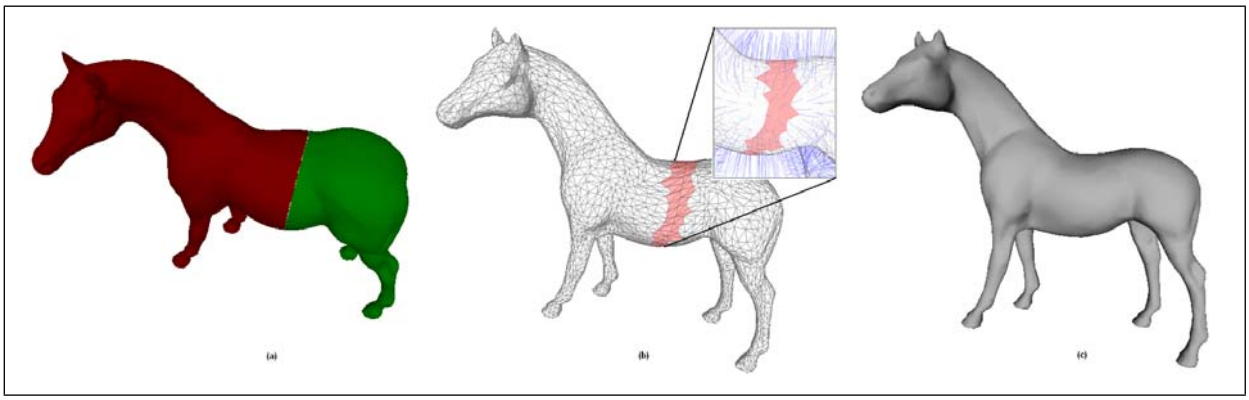

Figure 9: *(a) Two simplified partitions of horse model, (b) Normals of vertices around boundary bands appear approximately continuous after stitching, (c) Seamlessly stitched horse.*

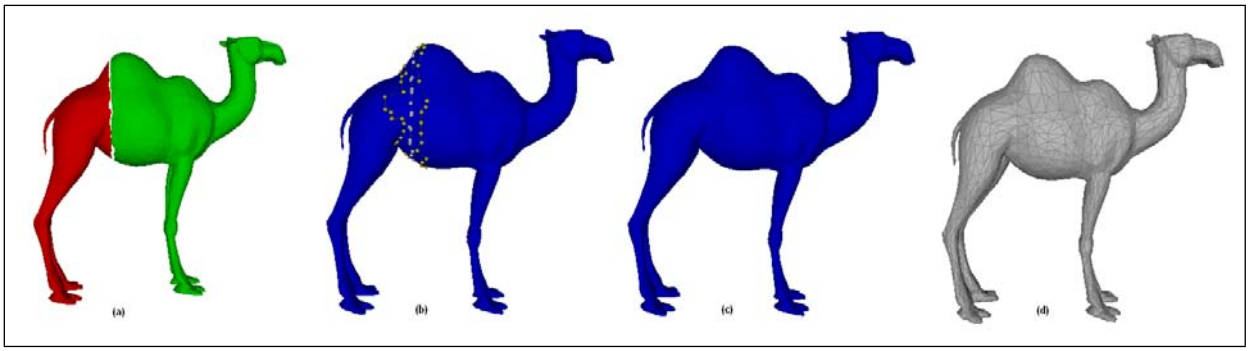

Figure 10: *(a) Two simplified partitions of camel model, (b) Stitched camel shown with its stitch vertices and near-boundary vertices, (c) Smooth looking stitched camel, (d) Camel wireframe showing seamless stitch result with balanced face distribution around stitch path.*

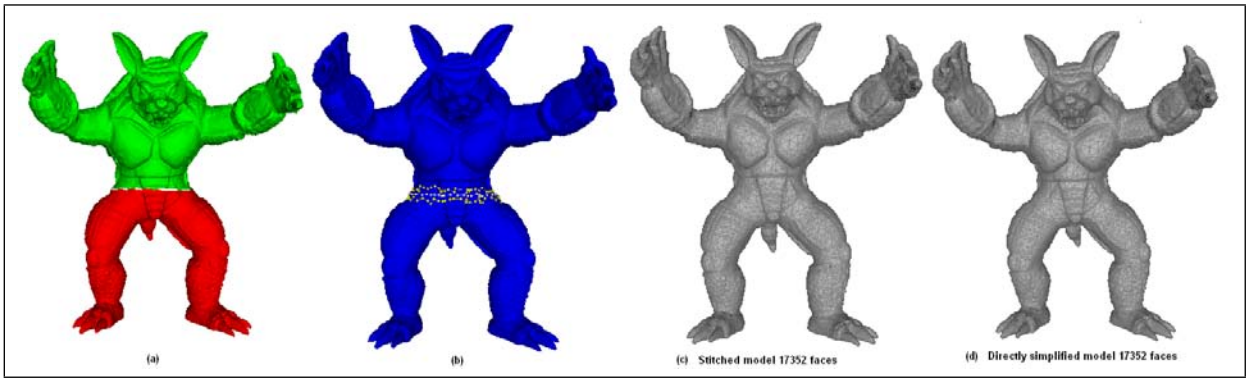

Figure 11: *(a) Two simplified partitions of Armadillo model, (b) Stitched Armadillo shown with its stitch vertices and near-boundary vertices, (c) Wireframe of stitched Armadillo (17532 faces) which has seamless stitch path and natural face distribution around stitch path, similar to wireframe of directly simplified Armadillo of same complexity (17532 faces) in (d).*

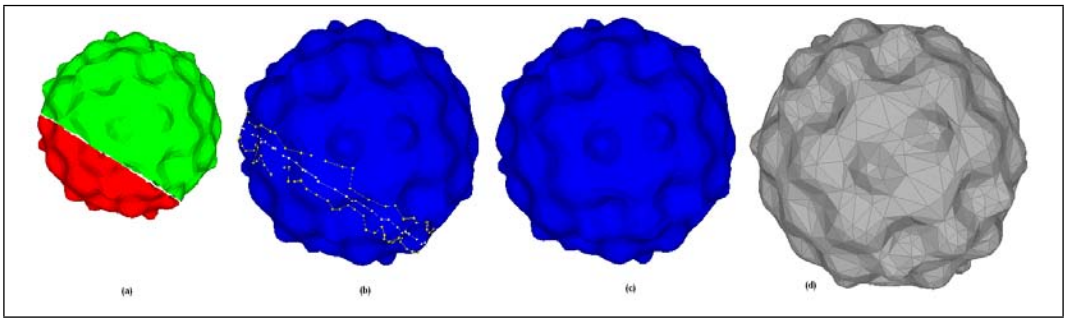

Figure 12: *(a) Two simplified partitions of bumpy sphere model, (b) Stitched bumpy sphere shown with its stitch vertices and near-boundary vertices, (c) Stitched bumpy sphere, with seamless result along and across the stitch path, (d) Wireframe of stitched bumpy sphere, which has seamless stitch path with natural face distribution around stitch path.*

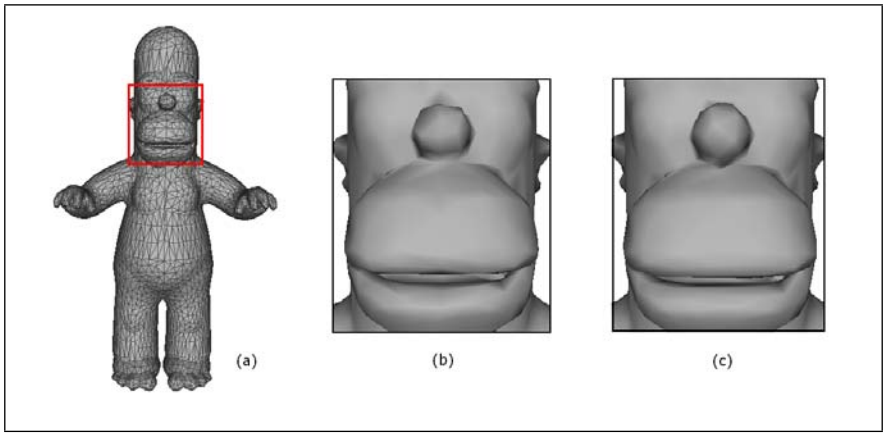

Figure 13: *(a) Wireframe of stitched Homer showing natural face distribution around stitch path, (b) Close-up image of stitched Homer's nose and mouth area in comparison to directly simplified Homer's in (c) of same resolution.*

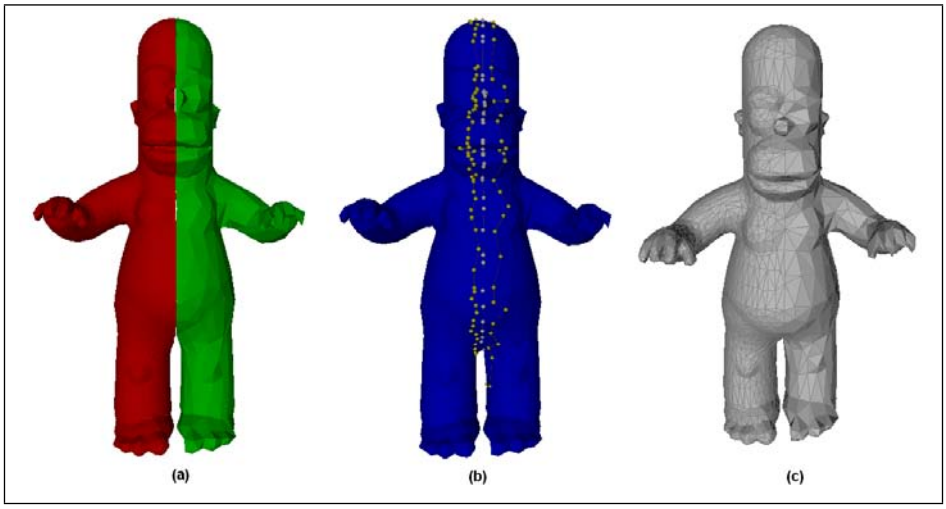

Figure 14: *(a) Two partitions of Homer model with different resolutions: 2658 faces for left partition and 674 faces for right partition, (b) Stitched Homer shown with its stitch vertices and near-boundary vertices, (c) Wireframe of stitched Homer which shows seamless stitch result across different resolution partitions.*