International Journal of CAD/CAM Vol. 9, No. 1, pp. 71~77 (2009)

Accurate Stitching for Polygonal Surfaces

Lifeng Zhu, Shengren Li, and Guoping Wang
The key lab on machine perception and intelligence of MOE Peking University Beijing, China

Abstract – Various applications, such as mesh composition and model repair, ask for a natural stitching for polygonal surfaces. Unlike the existing algorithms, we make full use of the information from the two feature lines to be stitched up, and present an accurate stitching method for polygonal surfaces, which minimizes the error between the feature lines. Given two directional polylines as the feature lines on polygonal surfaces, we modify the general placement method for points matching and arrive at a closed-form solution for optimal rotation and translation between the polylines. Following calculating out the stitching line, a local surface optimization method is designed and employed for postprocess in order to gain a natural blending of the stitching region.

1. Introduction

With the development of hardware and software for digital geometry processing, discrete surfaces are getting more popular in CAD and virtual animator modeling due to its flexibility in topological expression and convenience for rendering. The precision and smoothness is a major disadvantage of discrete surface, which can be compensated by high resolution and meaningful sampling.

However, in modeling realm, it is still difficult to design polygonal surface having complex topology. Traditional modeling methods, such as extruding, lofting and tessellating spline surfaces, only generate simple topology. Thus constructive solid geometry (CSG) operations are raised to tackle this problem, in which users are required to place each primitive carefully for generating complex models by Boolean operations, which, however, suffers a lot of problems such as computational efficiency and numerical stability [1]. If we mark feature lines to guide the placement of each surface, by stitching up them according to the feature lines, a complex model can also be constructed in a bottomup manner.

Besides, a popular metaphor of modeling called modeling by example asks for a composition operation [2]. Some mesh editing techniques, such as mesh fusion [5][7] and snapping [6], are employed to implement the composition operation by either extrapolating the disconnected surface or blending the overlap areas. In order to obtain visually good merging results and retrench computing cost, stitching the boundaries [2], [3], [4] is also a appropriate alternative for mesh composition.

Another way to acquire complex models is to reconstruct them from 3D scans. Due to the noise, different reconstruction algorithms are not always able to produce watertight surface [8]. Artifacts like small gaps and overlaps can be handled by some appropriate stitching operations [9].

In the applications above, current implementations of stitching mainly care about the visual quality. Here we present a stitching strategy which mainly focuses on precision : an optimal stitching line which optimizingly fits each seam of the surface patches. In the following text, we call the seams to be stitched feature lines and name the merged version of feature line stitching line. Regarding polygonal surface, all the lines can be represented as polylines. After marking the feature lines to be

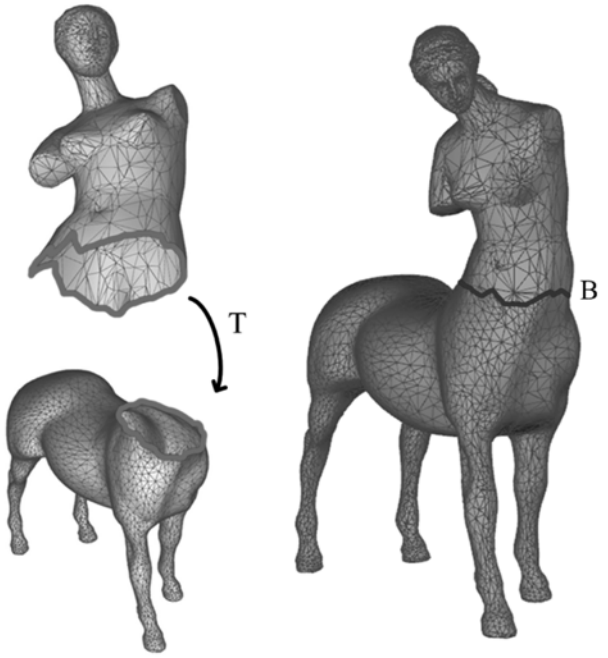

Fig. 1. Given the feature lines, we're searching for an optimal transformation *T* and stitching line *B*.

*Corresponding author:
E-mail: {zhulf,lsr,gwang}@graphics.pku.edu.cn

stitched, motivated by [10], we give a closed-form solution for optimal rigid transformation between polylines using quaternions, which enables free initial placement of the input surfaces. Then the optimal fitted stitching line is calculated, minimizing the L_2 error of the feature lines. Local surface optimization is then taken to the stitched surface and achieve creating a smooth transition from the stitching line to each surface patch.

2. Related work

The key problems in mesh composition are correspondence construction and snapping region merging, similar to the basic problems in shape morphing [11].

Previous work on vertex correspondence problem could be classified into two categories: boundary correspondence and surface correspondence. For boundary correspondence, [3] proposed three strategies for mesh merging, among which is helpful to surface stitching needs some user interaction to mark the sparse key vertex correspondence. In [13], part placement is aided by a user sketch interface, and the correspondence is then constructed by a projection from one boundary to the other surface. In modeling by example phase [2], Funkhouser et al. provide a voxel space implementation of ICP algorithm [12] to solve the boundary placement problem. For surface correspondence, in [14], a combined parameterization for the source and target mesh is built. Further contributions on compatible surface mapping see [15], [16]. One problem with combined parameterization is that it is only possible for two homeomorphic surfaces. This constraint could be solved by performing a joint parameterization just on the base region other than the whole meshes [17]. [6] introduces a soft-ICP algorithm, building the surface correspondence locally and softly, simultaneously finishing the blending stage.

Besides soft-ICP, there are some other snapping region merging methods. In [2], corresponding vertices on each boundary were attached and triangulated, followed by an additional smooth command. Poisson mesh editing [3] distributes the error from merging to the free vertices smoothly by solving a Poisson equation on mesh, while Laplacian mesh editing [18] linearly interpolates differential coordinates to generate a natural blending between two surfaces. Surface extrapolation could also do the merging work for disconnected parts. [19] performs a mesh extrapolating algorithm, while [5], [7] proposed a point-based extrapolation, followed by a mesh reconstruction algorithm.

For gap closing in mesh repair, volumetric methods are popular [20], [21], for identifying matching parts and gaps is convenient by volumetric representation. Surface oriented algorithms usually involve a stitching operation, which is instructive to our purpose. [22] matches the parts, and triangulated them minimizing some measurement such as the total areas of triangles and the dihedral angles of them. [23] determines corresponding edges within an error tolerance and snaps the corresponding vertices to close the gap. [8] builds both the vertex-vertex correspondence and vertexedge correspondence, and snap the correspondence progressively due to the error measurement of each correspondence.

Comparing to all previous work, our boundary-based method avoids distortion-introducing cross parameterization for surfaces and guessing the blank region. Comparing to previous boundary joining approaches, ICP-based technique [2] should take the local optimality into consideration, projection-based approach [3], [13] only make use of one boundary polyline, and another method in [3] is sensitive to the initial placement of the part. Our method enables free initial placement of surfaces, and the non-iterative solution doesn't lead to a local optimality. Full information including point geometry and point order from feature lines are sufficiently utilized for calculating a stitching line. Part placement and intermediate boundary solving are all accomplished in a framework of minimizing the L^2 error between polylines. Section 3 will give the details of our algorithm, and some experiments on mesh composition and mesh repair using our method will be presented and discussed in section 4.

3. A framework for surface patch stitching

Due to the arbitrariness of initial positions of the surfaces, an automatic placement for the feature lines is needed, for it is not convenient for users to put them together manually. Moreover, because of the ubiquity of the designing errors, it is not always possible to match the seams perfectly. Hence, we are going to search for an optimal transformation and an intermediate line to attach each surface patch together. We integrate the placement and polyline blending stage to obtain a good performance for CSG-like applications. At the same time, we decouple them from surface blending stage, and provide a possibility for a postprocess stage to erform local surface optimization, making it also useful for mesh merging.

Here we build a boundary-based framework for polygonal surface stitching. We take the polylines to be stitched as our input, and generate an attached model with the aligned and sewed polylines. The input of our algorithm could be either specified manually or generated by upstream applications.

3.1. Consistent parameterization

We note the two polygonal surface M_1, M_2 with the directional feature lines B_1, B_2 to be stitched up. Although the consistent parameterization for surfaces is avoided, we still have to build a correspondence between feature lines, which is more robust and meaningful for surface stitching without unnecessary distortion and heavy computational cost.

We select chord-length as the parameter and normalize the parametric domain to [0, 1] to get a consistent parameterization for the two feature lines. Suppose the parameterizationsof $B_1 = \{\mathbf{p}_i\}_{i=1}^{n}$, $B_2 = \{\mathbf{q}_i\}_{i=1}^{m}$ are $\{u_i\}_{i=1}^{n}$, and $\{v_i\}_{i=1}^{m}$, with $u_0 = v_0 = 0$ and $u_n = v_m = 1$. Then a combined parameterization is constructed as an overlay of $\{u_i\}_{i=1}^{n}$, and $\{v_i\}_{i=1}^{m}$, that is $\{s_i\} = \{u_i\} \cup \{v_i\}$. A parametric domain resampling approach is then taken to B_1 and B_2, that is, the points with the parameter $\{s_i\}$ are sampled on B_1 and B_2. The points having the same parameter are correspondent in the following stage.

3.2. Error-directed surface placement

After resampling, we still note points on B_1 and B_2 as$\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$, with $i = 1$ to n. An optimal transformation T should be determined to automatically align these parts, making the feature lines best fitted. Most previous approaches use the algorithm in [10], or a standard SVD minimization [24], solving for a rotation R and translation **t** which minimizing the error functional:

$$\sum_{i=1}^{n-1} \|\mathbf{p}_i - (R\mathbf{q}_i + \mathbf{t})\|^2 \tag{1}$$

Observing that the information we get is not only the point geometry but also connectivity of the point sets, here we introduce a different error measurement as objective of our optimization, the L^2 distance for curves:

$$E(\mathbf{x},\mathbf{y}) = \int_0^1 \|\mathbf{x}(u) - (R\mathbf{y}(u) + \mathbf{t})\|^2 du \tag{2}$$

Where $\mathbf{x}(u)$, $\mathbf{y}(u)$ is the parametric presentation of B_1 and B_2 on the parametric domain stated in 3.1. For $\mathbf{x}(u)$ and $\mathbf{y}(u)$ are piecewise linear,

$$E(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n-1} \int_{S_i}^{S_{i+1}} \|\mathbf{x}(u) - (R\mathbf{y}(u) + \mathbf{t})\|^2 du \tag{3}$$

Substitute $\mathbf{x}(u) = \mathbf{p}_i + (\mathbf{p}_{i+1} - \mathbf{p}_i)u$ and $\mathbf{y}(u) = \mathbf{q}_i + (\mathbf{q}_{i+1} - \mathbf{q}_i)u$ in $[s_i, s_{i+1}]$ into (3), we could rewrite (2) as

$$E(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n-1} \left(\left\| \frac{\mathbf{p}_i + \mathbf{p}_{i+1}}{2} - R\frac{\mathbf{q}_i + \mathbf{q}_{i+1}}{2} - \mathbf{t} \right\|^2 + \frac{1}{12} \right. \\ \left. \| \mathbf{p}_{i+1} - \mathbf{p}_i - R(\mathbf{q}_{i+1} - \mathbf{q}_i) \|^2 \right) \tag{4}$$

The first term measures the error between the middle point of each segment, depicting the “positional distance” between two segments after the rigid transformation. The second term contains the stretching and rotation energy after the transformation (see Fig. 2.). Comparing with the measurement (1), we could not simply use algorithm in [10] to minimize the first term of (4). We should also take the second term into consideration. In order to take advantage of quaternion for calculating the rotation, after circumspect analysis of formula (4), we achieve a close-form solution for T using quaternion, see appendix.

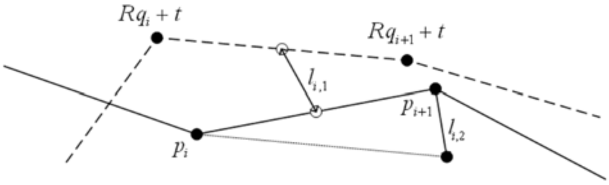

Fig. 2. $l_{i,1}$ is the positional distance and $l_{i,2}$ measuresthe stretching and rotation

3.3. Stitching line computing

Assume boundary B_2 is transformed by the optimal rotation R and translation t, we note the new position of B_2 as $\mathbf{y}(u)$. In order to stitch up $\mathbf{x}(u)$ and $\mathbf{y}(u)$, we would design a common boundary replacing B_1 and B_2, that is just the feature where surface M_1 touches surface M_2. For the purpose of making the common boundary, we call it stitching line $\mathbf{z}(u)$, keep as much information as $\mathbf{x}$(u) and $\mathbf{y}(u)$, we minimize the error term:

$$\int_0^1 \| \mathbf{z}(u) - \mathbf{x}(u) \|^2 + \| \mathbf{z}(u) - \mathbf{y}(u) \|^2 du \tag{5}$$

We’ve already built the correspondence for polyline $\mathbf{x}(u)$ and $\mathbf{y}(u)$. So here comes the calculation of the polyline $\mathbf{z}(u)$ which is closest to both $\mathbf{x}(u)$ and $\mathbf{y}(u)$ on the same parametric domain. Assuming at $[s_i, s_{i+1}]$, $\mathbf{z}(u) = \mathbf{r}_i + (\mathbf{r}_{i+1} - \mathbf{r}_i)u$. We still use the notation for $\mathbf{x}(u)$, $\mathbf{y}(u)$ in 3.2, and derive a polyline version of (5):

$$\begin{aligned} E(\mathbf{z},\mathbf{x},\mathbf{y}) = \sum_{i=0}^{n-1} 4(2\| \mathbf{r}_i \|^2 + 2\| \mathbf{r}_{i+1} \|^2 + 2\mathbf{r}_i^T \mathbf{r}_{i+1} \\ -(2\mathbf{r}_i^T \mathbf{p}_i + 2\mathbf{r}_{i+1}^T \mathbf{p}_{i+1} + \mathbf{r}_i^T \mathbf{p}_{i+1} + \mathbf{r}_{i+1}^T \mathbf{p}_i) \\ -(2\mathbf{r}_i^T \mathbf{q}_i + 2\mathbf{r}_{i+1}^T \mathbf{q}_{i+1} + \mathbf{r}_i^T \mathbf{q}_{i+1} + \mathbf{r}_{i+1}^T \mathbf{q}_i) \\ +\| \mathbf{p}_i \|^2 + \|\mathbf{p}_{i+1}\|^2 + \mathbf{p}_i^T \mathbf{p}_{i+1} \\ +\| \mathbf{q}_i \|^2 + \|\mathbf{q}_{i+1}\|^2 + \mathbf{q}_i^T \mathbf{q}_{i+1}) \end{aligned}$$

We range all $\mathbf{r}_i$ in a vector $\mathbf{R}$. The above formula could be written as $E(\mathbf{z}, \mathbf{x}, \mathbf{y}) = \frac{1}{2}\mathbf{R}^T A\mathbf{R} + \mathbf{b}T\mathbf{R} + c$. Where A is a constant matrix and $\mathbf{b}$ is a column vector with respect to $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$. Noticing E is quadratic in $\mathbf{R}$, minimizing E leads to solve a linear equation

$$A\mathbf{R} = \mathbf{b} \tag{6}$$

Proposition 1 Given two polylines $\mathbf{x}(u)$, $\mathbf{y}(u)$ on the

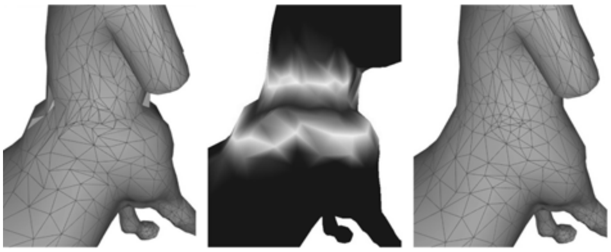

Fig. 3. left:unsatisfied stitched surface. middle:the ROI of stitching operation and its weight distribution. right:result after our local optimization

sameparameter domain, the polyline $\mathbf{z}(u)$ which minimizes the L^2 error (5) to both $\mathbf{x}(u)$ and $\mathbf{y}(u)$ could be written as $\mathbf{z}(u) = 1/2(\mathbf{x}(u) + \mathbf{y}(u))$

Proof. Substitute $\mathbf{r}_i = 1/2(\mathbf{p}_i + \mathbf{q}_i)$ into $\mathbf{R}$, the equation (6) holds.

Moreover, we could get the conclusion that for polyline $\{\mathbf{r}_i\} : \mathbf{r}_i = w\mathbf{p}_i + (1 - w)\mathbf{q}_i$ minimizes

$$\int_0^1 w\| \mathbf{z}(u)-\mathbf{x}(u) \|^2+(1-w)\| \mathbf{z}(u)-\mathbf{y}(u) \|^2 du, \quad w\in[0,1]$$

3.4. Stitching region optimization

We simply update the feature lines $\mathbf{x}(u)$ and $\mathbf{y}(u)$ to $\mathbf{z}(u)$ calculated in 3.3, and get a joint surface of M_1 and M_2. Stitching surfaces in this way keeps each surface's extrinsic shape (surface quality) and intrinsic shape (triangulation quality) mostly, only except the neighborhood of the stitching line. For some ill-designed surface patches (In our application, that means two patches have quite different feature lines), although we place the surface patches to the position where they best fitted each other and get the optimal stitching line, the quality of the stitched surface is still unsatisfied (Fig. 3). Here, we suggest an optional local surface optimization stage as a postprocessing. It's not quite necessary to perform surface optimization to the whole stitched surface, for most of the surfaces may be intentionally designed before our stitching, and the global optimization would possibly affect them. In contrast with selecting the ROI (region of influence) submesh and performing global mesh optimization on it, we assign weights to all the vertices and smoothly optimize the stitched surface near the stitching line.

We adopt geodesic-based and laplacian-based weighting strategies respectively. For geodesic-based weighting techniques, we calculate the geodesic distance from each vertex v to the stitching line B:

$$d(v,B) = \min_{v' \in B} d(v,v')$$

Here $d(v, v')$ is the approximate geodesic distance from v to v'. Fast marching method [15] or Dijkstra algorithm could be used to evaluate them. Other techniques like MMP algorithm [26] could get a more accurate result in the cost of computation. We use region growing technique to calculate these distances gradually from the stitching line, and terminate the growing when the distance reaches a userdefined threshold *maxd*. Then the weight of vertex v could be simply assigned by

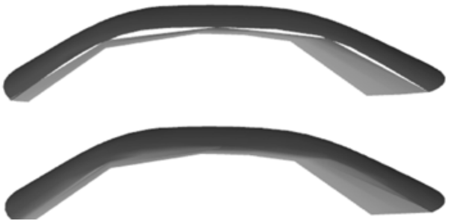

Fig. 4. top:part of car body with cracks. bottom:the repaired model.

$$w(v) = \begin{cases} 1-\dfrac{d(v,B)}{maxd} & d(v,B)\leq maxd \\ 0 & d(v,B)>maxd \end{cases} \tag{7}$$

or using the region-of-influence function designed in [27].

For Laplacian-based weighting techniques, we recommend the harmonic fields introduced in [28]. Setting weights of the vertices on B to 1, and 0 to the boundary of ROI, all the weights of vertices in ROI could be assigned gracefully by solving a linear equation. The two strategies we proposed both produce weights in [0, 1].

Set the ROI and the weights, we could perform mesh optimization smoothly from the stitching line, whose vertices all have weight 1, to vertices far from it, which have quite little weights or 0 as their weights. Almost all the mesh optimization algorithms involve some vertex relocation process, either iterative [29], [30] or non-iterative [31]. Suppose after vertex relocation, vertex v would be moved to v'. We modified the relocation procedure as

$$v'' = v + w(v' - v)$$

where w is the weight of v. In iterative optimizations, this new relocation stage is performed iteratively. Based on this framework of local optimization, we kept the vertices far from the stitching line unchanged and obtained a smoothly transition near the stitching line.

4. Applications and discussion

Implementation Since our algorithm merely needs some calculation on the boundary, the stitching could achieve at an interactive speed. We only need the user to specify the end points and the direction of the boundary. Then the model placement and stitching are both performed automatically. Of course, the input of our algorithm could also come from some upstream applications, such as mesh segmentation, trimming, feature extraction, etc. In the optional local region optimization procedure, we implemented the geodesic-based approach. It simultaneously generates the weights and the ROI. If we don't need the local optimization, we just simply set *maxd* = 0 in Eq.(7). Comparing to laplacian-based pproach, it is more convenient for users to define the ROI.

Comparison The complexity of our algorithm and the previous algorithm are both $O(n)$, where n is the number of the vertices. The computational cost satisfied with almost any applications. Since we focus on the precision, here we provide a quantitative comparison between methods[4][6] which minimize (1) and our method which minimizes (2). Noticing that algorithms in [4][6] are surface-based, in order to make a more meaningful comparison, we use a boundary-based version of [4][6]. We use the hausdoff distance of the feature lines after placement as the criterion of stitching error. We randomly generate two polylines at different scales. In order to reduce the influence of numerical stability, for each scale, we do the experiment 1000 times, and calculate the average errors as our results. Table1 shows the experiment results.

Discussion In table.1, we can find that, excluding the influence of numerical instability and data irregularity, our method is more accurate statistically. However, as the scale of the polylines increases, the vantage of our algorithm reduces. This is an interpretable phenomenon. The geometric meaning of (2) could be regarded approximatively as the square area of the surface bounded by the two polylines, which is more related to hausdoff distance; Contrastly, metric (1) only cares about the sum of the square distance of the corresponding vertices of the two polylines. Although as the vertices gets much denser, (1) approximates (2) gradually, the conceptional "infinitely small" is unreachable, which makes our method still more accurate than previous methods. This higher precision of our method is more beneficial to some manufacture application, which asks for a higher resolution than screen error.

Applications An example of applying our approach to CAD models see Fig. 5. For feature lines of CAD parts are required to kept after the composition, we skip the optimization stage.

Table 1. Precision comparison with previous algorithm

n = 3	error (2)	error (1)	hausdoff error
our algorithm	11.1392	10.1115	2.5544
algorithm in [4][6]	11.3223	9.4075	3.2532
n = 5	error (2)	error (1)	hausdoff error
our algorithm	16.6654	14.0505	3.6366
algorithm in [4][6	16.9027	13.5235	3.7066
n = 50	error (2)	error (1)	hausdoff error
our algorithm	63.4165	49.4309	2.3453
algorithm in [4][6	63.4697	49.2564	2.3626
n = 300	error (2)	error (1)	hausdoff error
our algorithm	157.8152	122.3988	1.2528
algorithm in [4][6]	157.8179	122.3816	1.2541

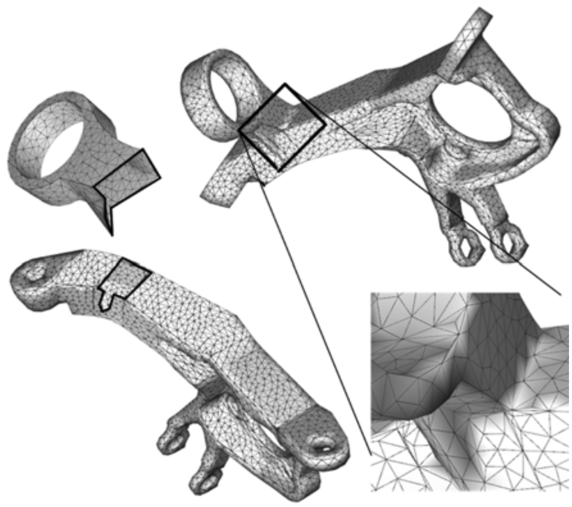

Fig. 5. left:two free placement engineer parts. right:stitched model without local optimization.

Fig. 6 shows one example of mesh composition using our strategy. A pig with fore camel foot is modeled by our method. An optional local optimization stage is taken to generate a smooth transition from the stitching line to both body and legs. Fig. 1 is another example.

Out algorithm is also applicable to mesh repair. Fig. 4 is an example. We only need to mark the end points of the crack, either manually or by feature recognition algorithms, the repairment of the crack is performed by our stitching method automatically.

5. Conclusion and future work

In this paper, we present a stitching strategy to generate new models from simple polygonal surface patches. We achieve at a transformation that best fit the seam of the stitching boundary, and merge them with the same error measurement. A local optimization framework is then provided, which merely affect the neighborhood of the stitching line. Using our strategy, mesh composition can obtain a nice quality surface near the stitching line with low computational cost.

Although we intend to keep most of the original

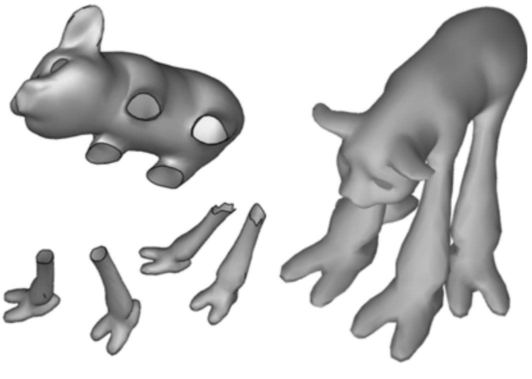

Fig. 6. A complex model composition of a body of pig and four foot from a camel.

surface unchanged, for some discrete patches having quite different sampling rates, the optimization procedure may not achieve at a natural combination of two surfaces. Some preprocess or postprocess about resampling are required. In the Future work, we would discuss some issues on the tessellation near the stitching boundary. With the CSG primitives getting more general, to build topologically complex models, more work on CSG representation should also be deeply researched besides the error-minimizing stitching.

Acknowledgment

This work is partially supported by the National Basic Research Program of China (Project No.2010CB328002), National Natural Science Foundation of China (Project No.60925007, 60833007, 60703062), and the National High Technology Research and Development Program of China (Project No.2007AA01Z318).

Appendix

We are calculating a R and $\mathbf{t}$ to minimize (4). Our derivation is under the framework of [10]. We've checked that although the objective function and local coordinate are different, quaternions are still useful for calculating R. We do all the calculation in a local coordinate to the centroids of the polyline's midpoints

$$\bar{\mathbf{p}}=\frac{1}{n-1}\sum_{i=1}^{n-1}\frac{\mathbf{p}_i+\mathbf{p}_{i+1}}{2},\quad \bar{\mathbf{q}}=\frac{1}{n-1}\sum_{i=1}^{n-1}\frac{\mathbf{q}_i+\mathbf{q}_{i+1}}{2}$$

All the vertices are recorded in a new coordinate as

$$\mathbf{p}'_i=\mathbf{p}_i-\bar{\mathbf{p}},\ \mathbf{q}'_i=\mathbf{q}_i-\bar{\mathbf{q}}$$

Then the first term of (4) becomes

$$\sum_{i=1}^{n-1}\left\|\frac{\mathbf{p}'_i+\mathbf{p}'_{i+1}}{2}-R\frac{\mathbf{q}'_i+\mathbf{q}'_{i+1}}{2}\right\|^2$$

$$-2t'^T\sum_{i=1}^{n-1}\left(\frac{\mathbf{p}'_i+\mathbf{p}'_{i+1}}{2}-R\frac{\mathbf{q}'_i+\mathbf{q}'_{i+1}}{2}\right)+(n-1)\|\mathbf{t}'\|^2$$

Where $\mathbf{t}'=\mathbf{t}-\bar{\mathbf{p}}+R\bar{\mathbf{q}}$. The middle term of the above expression is zero, and total error is minimized with $\mathbf{t}'=0$. That is, the optimal translation $\mathbf{t}'=\bar{\mathbf{p}}-R\bar{\mathbf{q}}$.

Here comes to optimize the rotation R, we remove all the terms which are constant or only respect to translation $\mathbf{t}$, and get the reduced objective function of total error

$$\frac{1}{12}\sum_{i=1}^{n-1}(\mathbf{p}'^T_i R\mathbf{q}'_i+\mathbf{p}'^T_{i+1}R\mathbf{q}'_{i+1}+5\mathbf{p}'^T_i R\mathbf{q}'_{i+1}+5\mathbf{p}'^T_i R\mathbf{q}_i') \tag{A.1}$$

Follow the derivation in section 4 of [10], we using quaternion d representing rotation R, and substitute it into (A.1). On the property of quaternions, (A.1) could be represented as a quadratic in d: d^TNd. Where

$$N=\begin{pmatrix} tr(M) & \Delta^T \\ \Delta & M+M^T-tr(M)I_3 \end{pmatrix}$$

where $\Delta=[A_{23},A_{31},A_{12}]^T$, $A_{ij}=(M-M^T)_{ij}$ and

$$M=\sum_{i=1}^{n-1}(\mathbf{p}'_i\mathbf{q}'^T_i+\mathbf{p}'_{i+1}\mathbf{q}'^T_{i+1}+5\mathbf{p}'_i\mathbf{q}'^T_{i+1}+5\mathbf{p}'_{i+1}\mathbf{q}'^T_i)$$

Then the optimization problem turns to extract the principle eigenvector $e=(e_0,\ e_1,\ e_2,\ e_3)^T$ of matrix N. Finally the rotation R could be reconstructed by the quaternion $e=(e_0, e_1, e_2, e_3)$

$$\begin{pmatrix} e_0^2+e_1^2-e_2^2-e_3^2 & 2(e_1e_2-e_0e_3) & 2(e_1e_3+e_0e_2) \\ 2(e_1e_2+e_0e_3) & e_0^2+e_2^2-e_1^2-e_3^2 & 2(e_2e_3-e_0e_1) \\ 2(e_1e_3-e_0e_2) & 2(e_2e_3+e_0e_1) & e_0^2+e_3^2-e_1^2-e_2^2 \end{pmatrix}$$

More theoretical details are in [10].

References

[1] Henning Biermann, Daniel Kristjansson, Denis Zorin: *Approximate Boolean Operations on Free-form Solids*. Proceedings of ACM. SIGGRAPH 2001, 185-194, 2001.

[2] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin: *Modeling by example*. ACM Trans. Graph. 23(3), 652-663, 2004.

[3] Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.Y.: *Mesh editing with poisson-based gradient field manipulation*. ACM Trans. Graph. 23(3), 644-651, (2004).

[4] X Huang, H Fu, Okc Au, CL Tai: *Optimal boundaries for Poisson mesh merging*. Proceedings of the 2007 ACM symposium on Solid and physical modeling, 30-35.

[5] Lin J., Jin X., Wang C.: *Sketch based mesh fusion*. Advances in Computer Graphics, 4035, 90-101, 2006.

[6] Sharf A., Blumenkrants M., Shamir A., Cohen-Or D.: *Snap-Paste: an interactive technique for easy mesh composition*. The Visual Computer 22, 9, 2006, 835-844.

[7] Lin J., Jin X., Wang C., Hui.K: *Mesh Composition on Models with arbitrary Boundary Topology*. IEEE Trans. Visualization and Computer Graphics, 14(3), 2008.

[8] P. Borodin, M. Novotni, and R. Klein: *Progressive gap closing for mesh repairing*. In J. Vince and R. Earnshaw, editors, Advances in Modelling, Animation andRendering, 201-213, Springer Verlag, 2002.

[9] A Guziec, G Taubin, F Lazarus, B Horn: *Cutting and Stitching: Converting Sets of Polygons to Manifold Surfaces*. IEEE Trans. Visualization and Computer Graphics, 7(2), 2001.

[10] B. K. P. Horn: C*losed-form solution of absolute orientation using unit quaternions*. Journal of the Optical Society of America, 4(4), 629-642, 1987.

[11] Alexa M., Cohen-Or D., Levin D.: *As-rigid-as-possible*

shape interpolation. In Proceedings of ACMSIGGRAPH (2000), 157-164.

[12] Besl, P., And Mckay, N. 1992: *A method for registration of 3D shapes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 239-256.

[13] Jeehyung Lee, Thomas Funkhouser: *Sketch-Based Search and Composition of 3D Models*. EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling, 2008.

[14] T. Kanai, H. Suzuki, J. Mitani, and F. Kimura: *Interactive mesh fusion based on local 3d metamorphosis*. In Graphics Interface, 148-156, 1999.

[15] Kraevoy V., Sheffer A.: *Cross-parameterization and compatible remeshing of 3D models*. ACM Transactions on Graphic23, 3, 861-869, 2004.

[16] Schreiner J., Asirvatham A., Praun E. Hoppe H.: *Inter-surface mapping*. ACM Transactions on Graphics, 23(3), 870-877, (2004).

[17] Fu, H., Tai, C.L., Zhang, H.: *Topology-free cut-and-paste editing over meshes*. In: Proceedings of the 3rd International Conference on Geometric Modeling and Processing (2004).

[18] Sorkine, O.,Cohen-Or, D., Lipman, Y., Alexa, M., Rossl, C., AND Seidel, H.-P. 2004. *Laplacian surface editing.* In Symposium of Geometry Processing.ACM SIGGRAPH/ Eurographics, 2004, 179-188.

[19] Levy, B. 2003. *Dual domain extrapolation*. ACM TOG 22(3), 364-369.

[20] Tao Ju. 2004. *Robust repair of polygonal models*. ACM TOG 23(3,) 888-895.

[21] Stephan Bischoff, Leif Kobbelt: *Structure Preserving CAD Model Repair*. Computer Graphics Forum. 24(3), 527-536, 2005.

[22] G. Barequet and M. Sharir. *Filling gaps in the boundary of a polyhedron*. Computer-Aided Geometric Design, 12(2), 207-229, 1995.

[23] G. Barequet and S. Kumar. *Repairing CAD models*. In Proc. IEEE Visualization, 363-370, 1997.

[24] K. S. Arun, T. S. Huang, S.D.Blostein: *Least-squares fitting of two 3-D point sets*, IEEE Trans. Pattern Analysis and Machine Intelligence, 9(5), 698-700, 1987.

[25] Kimmel, R., Sethian, J. A. 1998. C*omputing geodesic paths on manifolds*. Proc. of National Academy of Sci. 95(15) (July), 8431-8435.

[26] Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S., Hoppe, H. 2005. Fast exact and approximate geodesics on meshes. ACM Trans. Graph. 24, 3, 553-560.

[27] Museth, K., Breen, D., Whitaker, R.,Barr, A. 2002. *Level set surface editing operators*. ACM Trans. Graph. 21(3), 330-333.

[28] Rhaleb. Z, Christian. R, Zachi. K, Hans-Peter. S. *Harmonic Guidance for Surface Deformation*. Comp. Graph. Forum. 24(3), 2005.

[29] V. Surzahsky and C. Gotsman. Explicit surface remeshing. Proceedings of the EUROGRAPHICS / ACM SIGGRAPH Symposium on Geometry Processing, 20-30, 2003

[30] Taubin, G. 2000. *Geometric signal processing on polygonal meshes*. In State of the Art Report, Eurographics, 81-9

[31] Nealen, A., Igarashi, T., Sorkine, O., Alexa, M.: *Laplacian mesh optimization.* In: Proceedings of GRAPHITE 2006, 381-389, ACM Press.

Lifeng Zhu, PhD. candidate of the School of Electronics Engineering and Computer Science, Peking University. He received his BE degree at School of Mathematics, Fudan University in 2007. His research interests include geometric modeling and processing, shape analysis and computer graphics.

Guoping Wang received his BS and MS degrees from Harbin Institute of Technology, Harbin, China and his PhD degree from Fudan University in 1987, 1990 and 1997, respectively, all in mathematics. Dr. Wang has been a faculty member of the School of Electronics Engineering and Computer Science, Peking University, Beijing, China since 1999. He is now a full professor and the Director of HCI & Multimedia Lab. He is also the Associate Director of the Institute of Soft ware. His research interests include computer graphics, multimedia, virtual reality, geometrical modeling, CAD, and human-computer interaction.

Shengren Li, lsr@graphics.pku.edu.cn, master candidate of the School of Electronics Engineering and Computer Science, Peking University. His research interests include computer graphics and geometric modeling processing.

Lifeng Zhu

Shengren Li

Guoping Wang