Remeshing Triangle Meshes with Boundaries

Yong Wu, Yuanjun He, and Hongming Cai

Department of Computer Science & Technology, Shanghai Jiao Tong University, China
wuyong916@sjtu.edu.cn

Abstract. This paper proposes a spherical parameterization based remeshing approach to converting a given unstructured triangle mesh with boundaries into one having subdivision connectivity. In order to preserve the boundaries of original meshes, some special strategies are introduced into the remeshing procedure.

1 Introduction

Triangle meshes with subdivision connectivity are important for many multiresolution applications in graphics field. However, most meshes haven't this feature. So there are demands to transform an arbitrary mesh into one with subdivision connectivity. This transformation is called remeshing, which can be understood as an approximation operator $M - S$ that maps from a given irregular mesh M to a regular mesh S with subdivision connectivity. The resulting mesh is called a remesh of the original one. In this section we will give an overview of the most important work.

In [1], Eck et al. have presented a remeshing algorithm. The resulting parameterization is optimal for each base triangle but not smooth across the boundary between two base triangles. Moreover, runtimes for this algorithm can be long due to a lot of harmonic map computations. Lee and co-workers [2] develop a different approach to remeshing irregular meshes. Their method can be used to remesh meshes with boundaries, but the authors don't discuss how to preserve the boundaries.

In this paper, we present an approach for remeshing triangle meshes with boundaries. In order to preserve the boundaries, some special strategies are introduced into the subdividing procedure.

2 Remeshing

2.1 Framework of Our Remeshing Method

As described in Fig.1, our remeshing method contains seven steps.

Step 1: **Closing boundaries.** Before mapping M onto the unit sphere, we triangulate all boundary regions to generate a genus-0 triangle mesh M_Ψ.

Step 2: **Spherical parameterization**. After obtaining M_Ψ, we use Praun's method [3] on M_Ψ to generate a spherical parameterization mesh.

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3515, pp. 335 – 338, 2005.

Step 3: **Simplifying.** We iteratively execute half-edge contraction operation on C_Ψ to generate the initial base mesh C_t^0.

Step 4: **Optimizing.** To reduce the distortion of the remesh, we insert some new interior vertices to C_t^0 so as to obtain an optimal base mesh C^0 with triangles of similar size.

Step 5: **Subdividing.** The subdivision operation is iterated on C^0 until the error between S_Ψ^m and M_Ψ is below the user specified threshold ε. Here S_Ψ^m is the remesh corresponding to the spherical subdivision mesh C^m.

Step 6: **Sampling.** After obtaining C^m, we find the corresponding spatial point in original surface M_Ψ for each vertex of C^m. The resulting mesh is S_Ψ^m.

Step 7: **Reconstructing boundaries.** After deleting all the vertices and triangles inside the boundary regions from S_Ψ^m, we obtain the remesh S^m.

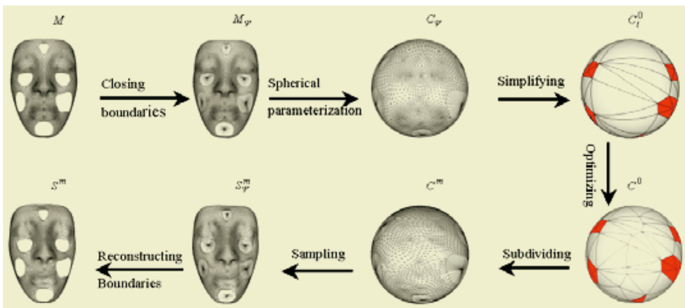
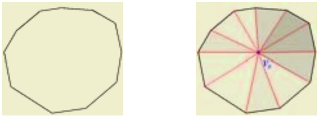


Fig.1. Frame of our remeshing method

2.2 Closing Boundaries

Since Praun's method [3] is only adapted to genus-0 meshes, all the boundaries of M have to be closed before parameterizing. In order to convenience further subdividing operation, we close the boundaries by a new strategy instead of traditional triangulation method. Fig.2 illustrates the new strategy, which inserts one new vertex v_B inside the region decided by a boundary that consists of a set of vertices $B = \{B_0, B_1, \hbar, B_n = B_0\}$. Here v_B is called one BIV (Boundary Interior Vertex) of M.

While inserting v_B, we have to find an appropriate spatial position for v_B to prevent triangles from overlapping each other. Since it is impossible to develop a universal method to decide v_B for the boundaries with arbitrary shape, we simply specify v_B to the average value of positions of all the boundary vertices. Then we scan each boundary edge $e(B_i, B_j)$ and construct a triangle $f(v_B, B_i, B_j)$ by anticlockwise order. After all boundary edges have been visited, we examine if some triangles of $T(v_B)$ overlap each other. If yes, we relocate the spatial position of v_B. Since the boundary number of a mesh is generally small, the users can complete the relocation operation manually.



(a) (b)

Fig. 2. Our triangulation strategy. (a) Open boundary. (b) Close boundary

(a) C_Ψ (b) C_t^0 (c) C^0

Fig. 3. Constructing process of the spherical base mesh

2.3 Constructing the Spherical Base Mesh

After mapping the closed mesh M_Ψ onto the unit sphere by Praun's method [3], we obtain the spherical parameterization mesh C_Ψ with the same connectivity as M_Ψ.

We start the construction of the base mesh C^0 by marking some special vertices of C_Ψ, which will be kept undeleted during the following mesh simplification. Then Gardland's half-edge contraction method [4] is used iteratively on C_Ψ to generate the initial base mesh C_t^0 with only marked vertices (Fig 3). Since Gardland's method selects the contraction edge by Quadric Error Metric, which doesn't consider how to optimize the triangle size, C_t^0 should be further optimized so as to generate a better base mesh C^0 (Fig 3-(c)).

2.4 Subdividing

After obtaining C^0, we iteratively execute subdividing operation on C^0 to generate the spherical subdivision mesh C^m. The subdivision level m should satisfy the inequality $H(S_\Psi^m, M_\Psi) \le \varepsilon$, where ε is the user specified threshold and $H(S_\Psi^m, M_\Psi)$ is the Hausdorff distance between the remesh S_Ψ^m and the original mesh M_Ψ (cf. Fig.1). Since S_Ψ^m is generated by using C^m to sample C_Ψ and M_Ψ, we use a vertex relocation operation to adapt the vertex distribution of C^m to that of C_Ψ, which will improve the visual appearance of S_Ψ^m.

2.5 Sampling the Original Surface

After obtaining the spherical subdivision mesh C^m, we need to find the corresponding spatial point in original surface M_Ψ for each vertex of C^m. This procedure is named as sampling the original surface. And the resulting mesh is S_Ψ^m. In this paper, we use the barycentric coordinates method to compute the corresponding spatial positions of vertices in C^m. After replacing each vertex of C^m by the corresponding spatial point, we obtain the spatial mesh S_Ψ^m. Then the remesh S^m is generated by deleting all BIVs and their 1-ring neighbor triangles from S_Ψ^m.

3 Experimental Results

We have implemented the remeshing approach and applied it to several triangle models with different number of boundaries. The original meshes are mapped onto the unit sphere by Praun's method. Fig 4 shows the remeshes of the Mask model (8,288 triangles, 7 boundaries) and the Bunny model (69,630 triangles, 2 boundaries).

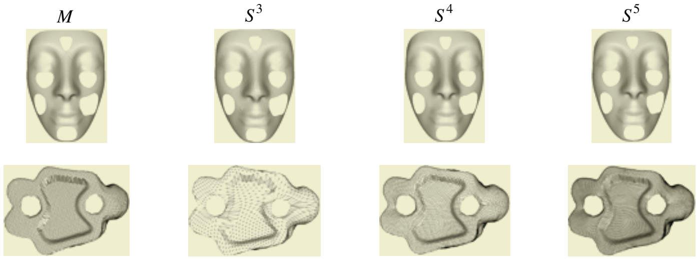


Fig. 4. The remeshing process of two different models

4 Conclusion

We have proposed an algorithm for remeshing triangle meshes with arbitrary number of boundaries. The experimental results show that our method can not only make the number of irregular vertices in the remesh as small as possible, but also preserve the boundaries of the original mesh well.

References

1. M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution Analysis of Arbitrary Meshes. In ACM Computer Graphics (SIGGRAPH '95 Proceedings), pages 173–182, 1995.
2. A. Lee,W. Sweldens, P. Schr¨oder, L. Coswar, and D. Dobkin. Multirelosution Adaptive Parameterization of Surfaces. In ACM Computer Graphics (SIGGRAPH '98 Proceedings), pages 95–104, 1998.
3. Praun, E., and Hoppe, H. Spherical Parameterization and Remeshing. In ACM Computer Graphics (SIGGRAPH '03 Proceedings), pages 340-349, 2003.
4. M. Garland and PS Heckbert. Surface Simplification Using Quadric Error Metrics. In ACM Computer Graphics (SIGGRAPH '97 Proceedings), pages 209-216, 1997.