
Mesh Generation Based On Surface Mosaics

Xiaowei Ma, Yu-Kun Lai, and Ralph R. Martin

Cardiff School of Computer Science and Informatics, Cardiff University, UK
{X.Ma, Yukun.Lai, Ralph.Martin}@cs.cardiff.ac.uk

1 Introduction

Mesh generation is a key issue in the numerical solution of partial differential equations (PDEs) for both finite element methods (FEMs) and computational fluid dynamics (CFD). This paper proposes a new method for quadrilateral (dominant) surface mesh generation, primarily aimed at CFD. Our method is based on the idea of surface mosaics [1, 2] from the computer graphics community—by analogy with Roman mosaics, surface mosaics cover an arbitrary surface, instead of images [3], with small square tiles. An algorithm in [1] shows how to generate equal-sized rectangular tiles over a 3D model (represented as a triangulated surface mesh) so as to almost cover it, without overlaps. The orientation of the tiles is controlled by a vector field which is interpolated from a few input control vectors. Fig. 1 gives an example of this output surface mosaics on a wing section model (m6). Taking such a surface mosaic as input, an initial quad mesh is generated by connecting the tile centroids, taking into account both distances and angles between adjacent centroids (Sec. 2). This strategy results in quad elements with good shape and which are well aligned with a vector field interpolated from the input control vectors (and hence features of the geometry, if control vectors are in agreement), properties which are highly desirable for downstream analysis.

A small number of non-quad elements is also generated; post-processing (hole-filling and cleanup) is applied to eliminate these where appropriate (Sec. 3). Various techniques exist [4, 5] to fill holes. We simply use certain patterns to replace the holes—most holes typically are one of a few kinds. To cleanup the mesh, our approach searches for problem patterns, following those in [8], although we have modified their approach somewhat, allowing us to take the vector field into account.

There are many mesh generation methods available. The three main approaches unstructured mesh generation include the octree approach, the advancing front approach and the Delaunay approach. Please refer to [6] for

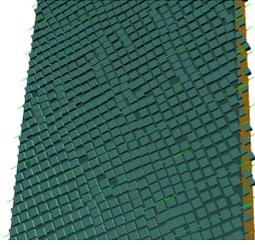


Fig. 1. An example of surface mosaics.

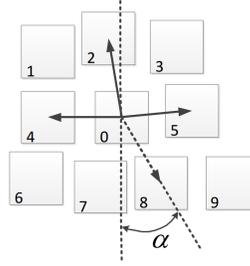


Fig. 2. Our connecting strategy.

an extensive review. Surface mosaics seem promising for mesh generation because: firstly, in an ideal case, both the centroids of the tiles in a surface mosaic, and the mesh points in surface mesh generation ideally should be distributed on a regular grid. Secondly, the ability to align the grid with key features of the target model is desirable in both applications. The mesh generated by our approach is not fully quadrilateral, but it has a good performance in terms of quad element quality and alignment to the preferred directions.

2 Connection strategy

The connection stage builds a mesh from a surface mosaic. Our algorithm considers each tile in turn and the one currently being considered is referred to as the *key tile*. In Fig. 2, tile 0 is a key tile. For each key tile, our algorithm determines the 8 nearest neighbouring tiles of a key tile, considering only tiles whose normal vectors are roughly in the same direction for robustness. Tile i ($i = 1, 2, \dots, 8$) are the neighbours of the key tile in Fig. 2.

Three steps are used to connect a key tile to appropriate neighbours:

- Find all neighbours within a certain distance.
- In each direction ('up', 'down', 'left' and 'right', as decided by the orientation of the key tile), select exactly one neighbour.
- In each direction, connect the key tile and selected neighbour to generate an edge of the mesh, if and only if the relationship between these two tiles satisfies certain angle and distance constraints.

The distance constraint simply ensures that the distance between the tile and the neighbour should be less than some maximum allowed length L . The angle constraint ensures that the angle between the line joining the centroids of the key tile and its neighbour, and either local coordinate direction given by the vector field, is less than a maximum allowed angle θ . Suitable values for L and θ will be discussed in Sec. 4.

Fig. 3(a) shows an intermediate mesh produced using the above connection strategy for a wing section model. No optimization has yet been applied. The mesh contains polygonal holes as well as triangle elements. Further operations

are now applied to fill in the holes, followed by topological cleanup to reduce the numbers of triangles.

3 Post-processing

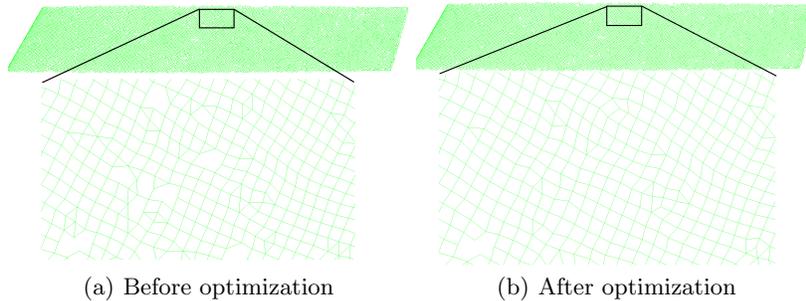


Fig. 3. Output mesh of our approach (close-ups are shown in the bottom).

To fill in the holes, the patterns used are illustrated in Fig. 4, where the top row shows patterns detected in the initial mesh, and their replacements are shown in the bottom row. In our current software, holes which do not fit the patterns can either be left untouched, or filled with triangles using a trivial strategy, depending on downstream requirements.

After hole filling, further topological cleanup is applied to reduce the number of triangles. More specifically, we use a local window to search for and replace several particular patterns, in particular those in Fig. 5 (Fig. 5(a) and (d) are similar to ones in [8]). We consider the local direction of the vector field for the pattern in Fig. 5(b). Note that these pattern replacement operations can be applied in various different orders. The order we use is to cycle over all elements, first fixing (b), then (c), (a) and (d) in turn; two iterations of this process are applied for topological cleanup.

This optimisation step removes the majority of non-quad elements (see Sec. 4), while the property that edges are well aligned with the vector field is largely preserved. However, some non-quad elements still exist, as the number of patterns used to replace holes is limited. We intend to investigate further patterns and rules in future.

4 Results

The quality metrics used are the following:

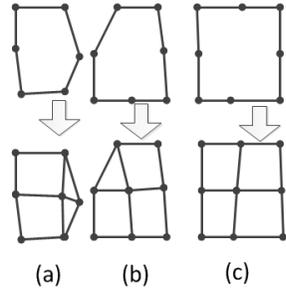


Fig. 4. Patterns used to replace holes.

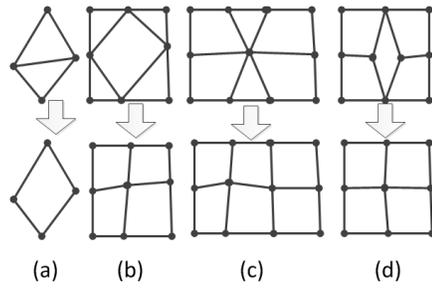


Fig. 5. Topological cleanup operations.

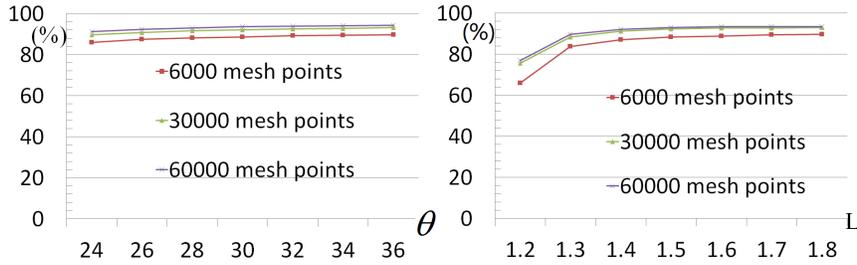


Fig. 6. Mesh quality, Q_r changes with angle tolerance θ (left), and L (right).

- The ratio $Q_r = n_q/n_e$, where n_q is the number of quads in the mesh and n_e is the number of total mesh elements. When we calculate Q_r , any remaining polygonal holes are simply covered with triangles.
- The shape measures $Q_g = \beta_1 h_{max} h_s / S_{min}$ for quads (1 for squares) and $T_g = \beta_2 h_s^2 / S$ for triangles (1 for equilateral triangles) (see [7] for details).

The average of T_g in all our tests is between 0.91 and 0.93, which means the average quality of triangles is quite high.

4.1 Sensitivity to parameters

In this section, all results relate to final meshes after topological optimization.

Fig. 6 shows how quality measure Q_r changes with angle tolerance θ (L is fixed at 1.6) and L (θ is fixed at 30°) on a wing section model (m6) for different numbers of output mesh points (6000, 30000 and 60000). From Fig. 6 we can see that Q_r is quite high, and is stable with respect to choice of θ , slightly improving with increased θ . Q_g very slightly deteriorates as changes of either θ or L , but is also stable at around 0.8. $\theta = 30^\circ$ and $L = 1.6$ seems to be a reasonable compromise and we used it in other tests.

4.2 Benefits of topological optimization

Topological optimization (cleanup) successfully increased the Q_r (from around 80% to around 90%) at the cost of a small drop in average shape quality (a drop less than 0.02) for the wing section model (m6) with $\theta = 30^\circ$ and $L = 1.6$. Results on other models are similar. Finally, Fig. 7 shows two meshes produced with 30000 mesh points using $L = 1.6$ and $\theta = 30^\circ$.

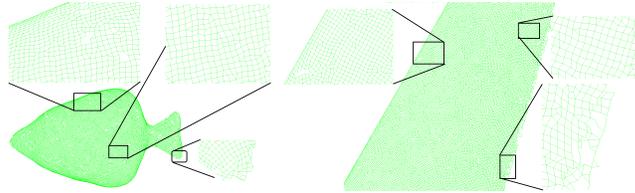


Fig. 7. Output meshes of fish (left) and wing (right) models.

Acknowledgements

This work is funded by Airbus Operations Ltd and the Chinese Scholarship Council (CSC). We also wish to thank Scott Shaw of Airbus for encouragement and support during this work.

References

1. Lai Y.K., Hu S.M., Martin R.R. (2006) Surface Mosaics. *Visual Comput.* 22(9-10):604-611
2. Passos V.D. and Walter M. (2008) 3d mosaics with variable-sized tiles. *The Visual Computer*, 24(7-9):617-623
3. Haeberli P. (1990) Paint by numbers: Abstract image representation. In *Proceedings of SIGGRAPH*, 207-578
4. Davis J., Marschner S., Garr M. Levoy M. (2002) Filling holes in complex surfaces using volumetric diffusion. *First International Symposium on 3D Data Processing, Visualization and Transmission*, 428-438
5. Schilling A., Bidmon K., Ove S., Ertl T. (2008) Filling Arbitrary Holes in Finite Element Models. *Proceeding of International Meshing Roundtable*, 231-248
6. Baker T.J. (2005) Mesh generation: art or science? *Prog. Aerosp. Sci.* 41:29-63
7. Frey P.J., George P.-L. (2000) *Mesh generation: application to finite elements*. ISTE, London, UK
8. Sampl P. (2000) Semi-Structured mesh generation based on medial axis. *Proceedings of 9th International Meshing Roundtable*, 21-32
9. Leatham M., Stokes S., Shaw J.A., Cooper J., Appa J., Blayloc T.A. (2000) Automatic Mesh Generation for Rapid-Response Navier-Stokes Calculations. *AIAA Paper 2000-2247*