Title:
Automatic Quad Patch Layout Extraction for Quadrilateral Meshes

Authors:
Sen Zhang, stephengenius@gmail.com, Tsinghua University
Hui Zhang, huizhang@tsinghua.edu.cn, Tsinghua University
Jun-hai Yong, yongjh@tsinghua.edu.cn, Tsinghua University

Keywords:
Quad Patch Layout, Safety Turning Area, Binary Integer Programming, Global Optimal Solution

DOI: 10.14733/cadconfP.2015.62-66

Introduction:

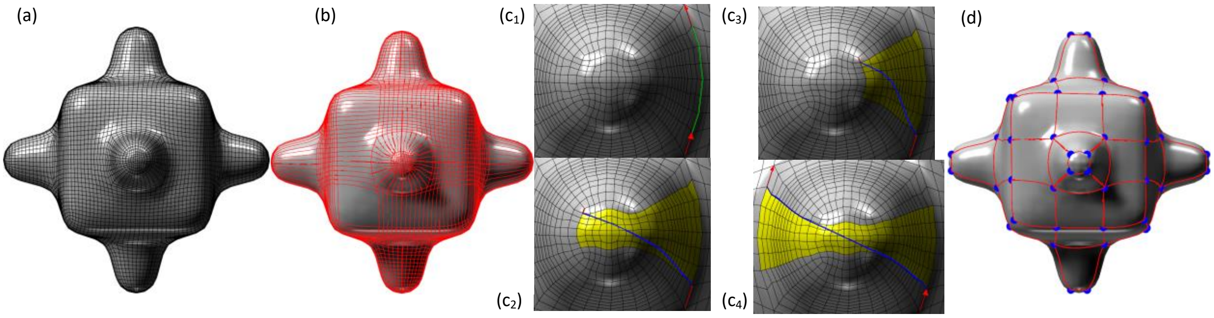

Fig. 1: Main steps of the proposed algorithm. (a) CubeBlob model, (b) The initial separatrices (red lines), Fig (c1) - (c4) are an example to find all possible separatrices related to one port (red arrows), where the *safety turning area* are shown in yellow areas, and the geodesic paths in the *safety turning area* are shown in blue lines. The green lines in (c1) are a unique separatrix for its *safety turning area*. (d) The desired patch layout with the minimal energy, in which the boundaries are shown in red lines, the intersections of separatrices are shown in blue points.

Quadrilateral mesh (quad mesh for short) is one of the most popular shape representations in computer graphics, which is widely used in CAD/CAM, numerical simulation and other areas. Among quad meshes, semi-regular ones, which have few numbers of irregular vertices, have been gained more attentions [1]. Significant progress has been made in generation and processing of quad meshes during the last decades. However, most semi-regular quad mesh generation algorithms focus on minimizing the number of the irregular vertices. When analyzing the global structure of a generated quad mesh, as it is depicted in Fig. 1(b), it does not exhibit a coarse and well-shaped patch layout. In practice, a coarse patch layout seen in Fig. 1(d) is highly desirable to support operations such as texturing, adaptive sizing and so on.
Generally speaking, there are two main types of methods for extracting the quad patch layout of a quad mesh: one type is based on mesh segmentation, and the other is based on simplifying separatrices and connectivity graph. Bommes et al. [1] gave a more comprehensive survey of the quad mesh processing.

Mesh Segmentation: Vieira and Shimada [2] segmented the mesh while iterating between region growing and surface fitting. [3], [4] and [5] all segmented the mesh with the help of geometric measure. Benko and Varady [3] segmented the mesh by approximating each patch by geometrical primitives. Cohen-Steiner et al. [4] drove the distortion error down through repeated clustering using the concept of geometric proxies. Wu and Kobbelt [5] extended this method by allowing planes, spheres, cylinders and rolling-ball blend patches. Myles et al. [6] used a greedy algorithm to generate coarse quadrilateral patches which were appropriate to fit with T-splines. Eppstein et al. [7] partitioned the mesh into structured quadrilateral patches with the help of motorcycle graph proposed in [8].
However, the methods based on mesh segmentation might change the number and distribution of the irregular vertices, which could bring in some deformation distortions during extracting patch layout.

Separatrices and Connectivity Graph Simplification: Bommes et.al [9] used GP-Operators, to greedily simplify the local helical quadrilateral structures. Then they used a convex mixed-integer quadratic programming formulation [10] to generate reliable quad mesh, which could achieve high-quality coarse quad layout by globally searching. Campen et al. [11] proposed the algorithm which was consisted in the direct construction of a simple connectivity out of a prescribed cross-field to produce a good base complex of the mesh. Tarini et al. [12] used two atomic operators, "delete" and "open" moves, to disentangle the separatrices and connectivity graph by selecting separatrices that were as short as possible. In the algorithm, the separatrices were selected by a depth-first searching method, and although in practice, the produced structure dramatically improved over the input graph, the strategy was greedy and in theory it cannot guarantee to get the most optimized result. Besides, the possible operations were high coupling with the energy defined on separatrices in this method, once the energy definition was changed, it needs to re-detect all possible operations and lead to a high time consuming.
The methods based on simplifying separatrices and connectivity graph mostly maintained the consistency of the irregular vertices, but they used a greedy strategy to select the redirected separatrices, which might obtain partial optimal solutions instead of the global optimal solution.

<u>Main Sections:</u>
A quad mesh embedded in 3D can be represented as M = (V, E, Q), where V, E, Q are vertices, edges and quads respectively.
Given a closed quad mesh, a vertex is *regular* if and only if its valance is 4, otherwise it is *irregular*. Topologically a regular vertex is the crossing of two coordinate lines in a 2D Cartesian grid and therefore a right-hand local coordinate system could be built in counter-clockwise order with u, v, -u, -v [1] as depicted in Fig. 2.
Port: A port is the outgoing edge adjacent to an irregular vertex (seen in Fig. 2).
Separatrix: A separatrix contains a directed edge sequence and two endpoints which are restrained to be irregular vertices (seen in Fig. 2). The inner vertices passed by the separatrix are all regular.

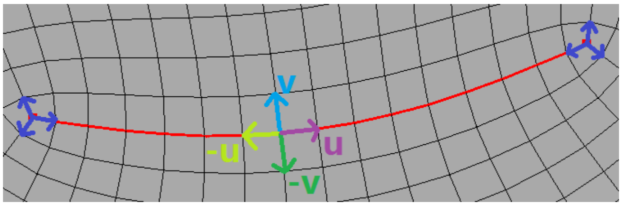

Fig. 2: A right-hand local coordinate system with u, v, -u, -v built at a regular vertex. Ports at an irregular vertex are shown as blue arrows. A separatrix is shown as a red line.

Connectivity Graph: If every port of any irregular vertex is associated with a particular separatrix, all separatrices and their crossings form a connectivity graph of the quad mesh.
Considering the separatrices as the boundaries of the patches, one connectivity graph segments the quad mesh into a patch layout (seen in Fig. 1). Obviously, since the corners in the patch layout are either irregular vertices or the crossings of the separatrices, and the number of irregular vertices is the intrinsic property of a quad mesh, the complexity of the patch layout is directly determined by the

Proceedings of CAD'15, London, UK, June 22-25, 2015, 62-66

number of crossings. At the same time, as a long separatrix has a greater probability of generating crossings, finding short separatrices in the quad mesh and using them to form the connectivity graph can help us to reduce the number of the crossings and further to achieve a coarse and well-shaped patch layout. What is more, we focus on extract a quad patch layout since a quad topology of the mesh is more preferred in many applications.

A patch layout is called *quad* if and only if every patch in it is bounded by four different separatrices.

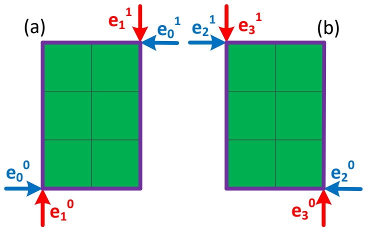

Fig. 3: Four types of safety turning areas. In the rectangle area of (a), which is bounded by purple lines, and the ports (e_0^0, e_0^1) (shown in blue arrows), which have the same local parametric direction and connect the rectangle area at the bottom-left and top-right corners respectively, can form a safety turning area. Likewise, ports (e_1^0, e_1^1), (e_2^0, e_2^1) and (e_3^0, e_3^1) can also form valid safety turning areas with the rectangle area respectively.

Safety Turning Area: A safety turning area is the combination of a rectangle area in the quad mesh and two ports of irregular vertices. The vertices in the rectangle area are all regular. The two ports (called "diagonal ports") connect the rectangle area at the diagonal corners and have the same local parametric direction.

All four types of safety turning area are shown in Fig. 3. There are two main reasons for the definition of safety turning area:

- Considering the separatrices in a safety turning area, "safety turning" means that the edge sequence can be freely changed in the rectangle area, since there is no irregular vertex in it.
- That the diagonal ports have the same local parametric direction makes the separatrices in the safety turning area along the local direction of the cross field, which efficiently reduces the probabilities of forming non-quad patches.

Supposing that in the rectangle area of a safety turning area, along the local parametric direction (called "x-dir") of the port, the length of the rectangle area is L_{wid}. And along the orthogonal parametric direction (called "y-dir") of the port, the length of the rectangle area is L_{ext}. Obviously, the minimal length of separatrices in the safety turning area is ($L_{wid} + L_{ext}$).

Considering the edge sequence of the separatrices in the rectangle area,

- If the number of the edges is one for every column (i.e., the edges in a separatrix and parallel to "x-dir"), the separatrix is called *x-monotone*.
- If the number of edges is one for every row (i.e., the edges in a separatrix and parallel to "y-dir"), the separatrix is called *y-monotone*.

A *monotone* separatrix satisfies both the x-monotone and y-monotone constraints, seen in Fig. 4.

We assume the input quad mesh have uniform edge length, all monotone separatrices in the same safety turning area and associated with the same diagonal ports share a common length. What is more, the common length is just the shortest length, seen in Fig. 4(c) and Fig. 4(d).

As a consequence, considering two ports in the mesh, if a safety turning area associated with them can be found, one monotone separatrix can be selected as the candidate separatrix for all separatrices connecting them, since the monotone one is the shortest.

At the same time, if two safety turning areas have overlapped area, the candidate separatrices of them should be selected carefully, since bad choose may bring in crash for quad layout detection. The geodesic path associated the diagonal ports in the safety turning area could help us selected proper separatrices, since they could efficiently avoid unnecessary crossings, seen in Fig. 5.

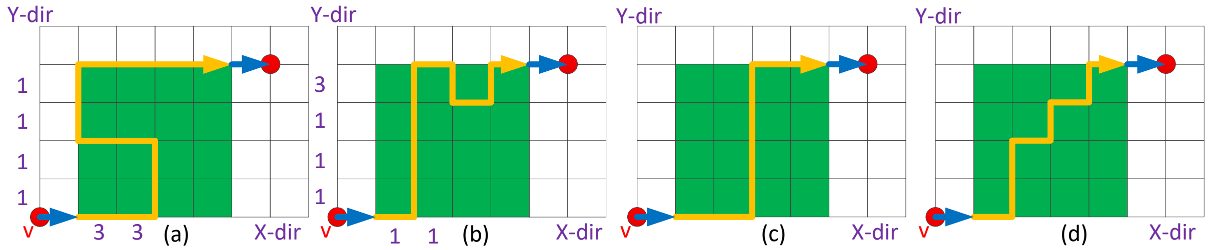

Fig. 4: Monotone Separatrix. The red points in the figure are denoted as irregular vertices, and the green quads form the rectangle area of the safety turning area. In (a) and (b), the edges of separatrices (yellow lines) in the rectangle area are not monotone in “x-dir” and “y-dir” respectively. In (c) and (d), the separatrices (blue lines and yellow lines) satisfy both x-monotone and y-monotone constraints. Although the edge sequences of separatrices are different in (c) and (d), they have the same and minimal length.

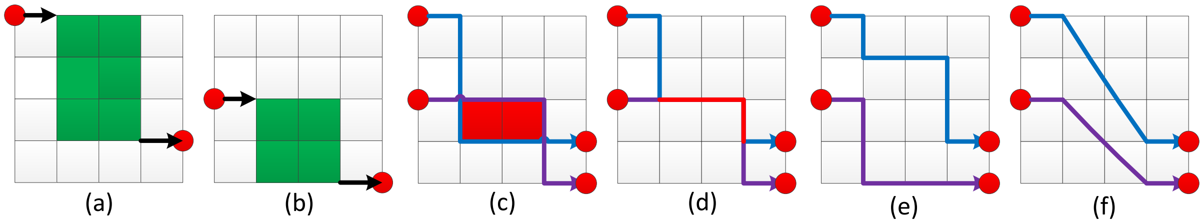

Fig. 5: Geodesic Path could efficiently avoid unnecessary crossings. In (c) and (d), the blue line is the candidate separatrix selected from the safety turning area in (a), and the purple line is the candidate separatrix selected from the safety turning area in (b). However, the red quads in (c) will bring in a non-quad patch in the layout, and red lines in (d) will bring in crash for quad layout detection, since the separatrices are overlapped. Instead of analyzing the safety turning area to select proper separatrix, geodesic path in (e) could help us to choose proper ones in (f).

In this paper, based on simplifying the separatrices and the connectivity graph, we present an automatic method to extract a coarse and well-shaped quad patch layout in a quad mesh. The key idea is to firstly find all the candidates of the shortest separatrices in the quad mesh based on safety turning areas, and then with the help of a binary integer programming solver, globally minimized the total energy of the selected separatrices until they could extract a quad patch layout. Fig. 1 provides a quick overview over the stages of the procedure. More results are shown in Fig. 6. Tab. 1 shows the time used for the examples, where N_1 is the number of the quad patches formed by initial separatrices, N_2 is the number of the desired quad patches, M is the times of calling binary integer programming solver during the globally optimization, and T is the time used by our method.

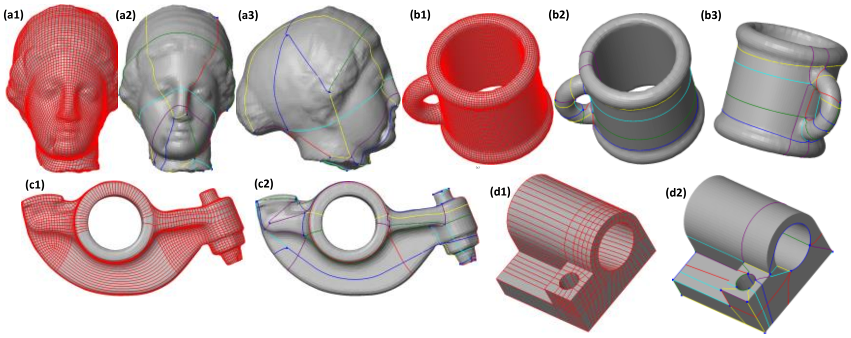

Fig. 6: More results of quad layout extraction. The boundaries of the desired patch layout are shown in colored lines and they are all the geodesic paths in *safety turning areas*.

Model	N_1	N_2	M	*T (s)*
CubeBlob [Fig. 1]	5600	78	1	0.16
Cup [Fig. 6(b)]	14983	23	1	0.15
Igea [Fig. 6(a)]	8183	28	2	0.25
Rockarm [Fig. 6(c)]	4524	101	2	0.24
Joint [Fig. 6(d)]	498	52	3	0.34

Tab. 1: Timing results.

The main contributions of this paper are as follows.

- With the help of safety turning area, some candidate separatrices could be selected to stand for all shortest separatrices, which greatly reduce the number of separatrices considering for the layout.
- We formulate the problem of finding the proper candidate separatrices to extract quad patch layout as a binary integer programming problem. Compared with the method by a greedy strategy, it efficiently avoids obtaining partial optimal solutions and enables us to find the global optimal solution.
- With the help of the geodesic path, a simple but efficient method is used to quickly determine whether a solution of the binary programming problem can extract quad patch layout.

References:

[1] David, B.; Bruno, L.; Nico, P.; Enrico, P.; Claudio, S.; Marco, T.; Denis, Z.: Quad-Mesh Generation and Processing: A Survey, Computer Graphics Forum, 32(6), 2013,1-26. http://dx.doi.org/10.1111/cgf.12014

[2] Miguel, V.; Kenji, S.: Surface Mesh Segmentation and Smooth Surface Extraction through Region Growing, Computer Aided Geometric Design, 22(8), 2005, 771-792. http://dx.doi.org/10.1016/j.cagd.2005.03.006

[3] Pal, B.; Tanmas, V.: Direct Segmentation of Smooth, Multiple Point Regions, Geometric Modeling and Processing, 22(8), 2002, 169-178. http://dx.doi.org/10.1109/GMAP.2002.1027508

[4] David, C. S.; Pierre, A.; Mathieu, D.: Variational Shape Approximation, Computer Aided Geometric Design, 23(3), 2004, 905-914. http://dx.doi.org/10.1145/1186562.1015817

[5] Jianhua, W.; Leif, K.: Structure Recovery vis Hybrid Variational Surface Approximation, Computer Graphics Forum, 24(3), 2005, 277-284. http://dx.doi.org/10.1111/j.1467-8659.2005.00852.x

[6] Ashish, M.; Nivo, P.; Denis, K.; Denis, Z.: Feature-aligned T-meshes, ACM Transactions on Graphics, 29(4), 2010, 117. http://dx.doi.org/10.1145/1778765.1778854

[7] David, E.; Michael, T. G.; Ethan, K.; Rasmus, T.: Motorcycle Graphs: Canonical Quad Mesh Partitioning, Computer Graphs Forum, 27(5), 2008, 1477-1486. http://dx.doi.org/10.1111/j.1467-8659.2008.01288.x

[8] David, E.; Jeffrey, G. E.: Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions, Discrete and Computational Geometry, 22(4), 1999, 569-592. http://dx.doi.org/10.1145/276884.276891

[9] David, B.; Timm, L.; Leif, K.: Global Structure Optimization of Quadrilateral Meshes, Computer Graphics Forum, 30(2), 2011, 375-384. http://dx.doi.org/10.1111/j.1467-8659.2011.01868.x

[10] David, B.; Marcel, C.; Hans-Christian, E.; Pierre, A.; Leif, K.: Integer-Grid Maps for Reliable Quad Meshing, ACM Transactions on Graphics, 32, 2013, 4. http://dx.doi.org/10.1145/2461912.2462014

[11] Marcel, C.; David, B.; Leif, K.: Dual Loops Meshing: Quality Quad Layouts on Manifolds, ACM Transactions on Graphics, 31(4), 2012, 110:1-110:11. http://dx.doi.org/10.1145/2185520.2185606

[12] Marco, T.; Enrico, P.; Daniele, P.; Nico, P.; Paolo, C.: Simple Quad Domains for Field Aligned Mesh Parameterization, ACM Transactions on Graphics, 30(6), 2011, 142. http://dx.doi.org/10.1145/2070781.2024176

[13] David, B.; Henrik, Z.; Leif, K.: Mixed-Integer Quadrangulation, ACM Transactions on Graphics, 28(3), 2009, 77. http://dx.doi.org/10.1145/1531326.1531383