COMPUTER-AIDED DESIGN & APPLICATIONS, 2016
VOL. 13, NO. 3, 409–416
http://dx.doi.org/10.1080/16864360.2015.1114399

Automatic Quad Patch Layout Extraction for Quadrilateral Meshes

Sen Zhang, Hui Zhang and Jun-Hai Yong

Tsinghua University, China

ABSTRACT
In this paper, we present an automatic method to extract quad patch layout with monotone boundaries in a quadrilateral mesh. Our approach simplifies the separatrices and connectivity graph topologically compared with the previous work. The key idea of our method is firstly to find all the candidates of the short separatrices based on the *safety turning areas*, and secondly, the problem of finding proper separatrices to extract quad patch layout is formulated as a binary integer programming problem. In the process of solving this problem, each solution is detected whether to extract quad patch layout, and finally the one with the minimum energy is determined as the global optimal solution. The produced quad patch layout is well-shaped and the corresponding base complex is coarse, which can be further used for many mesh processing applications, such as texturing, NURBS fitting and so on.

KEYWORDS
Safety turning area; binary integer programming; global optimal solution

1. Introduction

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 [3]. 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.

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 short 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.

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.

1.1. Related Works

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. [3] gave a more comprehensive survey of the quad mesh processing.

Mesh Segmentation: Vieira and Shimada [16] segmented the mesh while iterating between region growing

CONTACT Jun-Hai Yong yongjh@tsinghua.edu.cn
This article was originally published with errors. This version has been corrected. Please see Erratum (http://dx.doi.10.1080/16864360.2016.1139765)

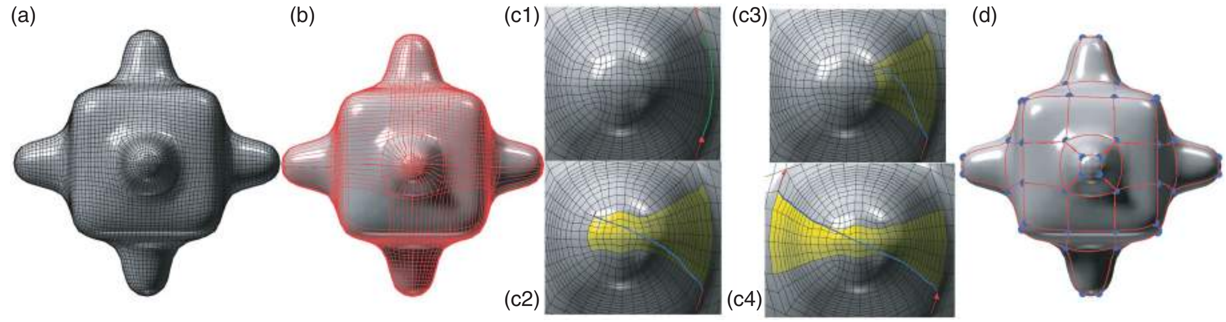

Figure 1. Main steps of the proposed algorithm. (a) CubeBlob model, (b) The initial separatrices (red lines), Fig (c1) - (c4) are an example to find all possible separatrices related to one port (red arrows), where the *safety turning area* are shown in yellow areas, and the geodesic paths in the *safety turning area* are shown in blue lines. The green lines in (c1) are a unique separatrix, where Lwid is 0. (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.

and surface fitting. [5], [6] and [17] all segmented the mesh with the help of geometric measure. Benko and Varady [5] segmented the mesh by approximating each patch by geometrical primitives. Cohen-Steiner et al. [6] drove the distortion error down through repeated clustering using the concept of geometric proxies. Wu and Kobbelt [17] extended this method by allowing planes, spheres, cylinders and rolling-ball blend patches. Myles et al. [12] used a greedy algorithm to generate coarse quadrilateral patches which were appropriate to fit with T-splines. Eppstein et al. [9] partitioned the mesh into structured quadrilateral patches with the help of motorcycle graph proposed in [8]. Gunpinar et al. [10] extended the motorcycle graph, and with the help of a patch growing strategy, the bi-monotone patches were achieved.

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. [2] used GP-Operators, which were the generalization of the poly-chord collapse operations, to greedily simplify the local helical quadrilateral structures, and what was more, each simplification process changed the local quadrilaterals while maintaining quad-consistency as well as irregular vertices. Then they used a convex mixed-integer quadratic programming formulation [4] to generate reliable quad mesh, which could achieve high-quality coarse quad layout by globally searching. Campen et al. [7] proposed the algorithm which was consisted in the direct construction of a simple connectivity out of a prescribed cross-field. In the algorithm, a simple operator was used in the process to greedily add a candidate "dual-loop" to generate a quadrilateral mesh which reproduced the cross-field's singularities, and it produced a good base complex of the mesh. Li et al. [11] used two global operations, "re-sampling" and "re-distribution", to optimize the shape of the patch layout in quad mesh, and finally make the mesh possess, as much as possible. Peng et al. proposed a connectivity editing framework for quad [13] and quad-dominant [14] meshes, which allowed user to edit mesh connectivity by controlling the number and distribution of the irregular vertices and irregular faces, and then they could illustrate the advantages and disadvantages of different strategies for quad/quad-dominant mesh design. Tarini et al. [15] used two atomic operators, "delete" and "open" moves, to disentangle the separatrices and connectivity graph by selecting separatrices that were as "short" as possible. Once the energy of all separatrices could not be deduced, the patch layout of the quadrilateral mesh is generated. 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.

The rest of the paper is organized as follows. Some problem definitions are listed in section 2, *safety turning areas* and their candidate separatrices are found in section 3. A binary integer programming problem of finding the proper separatrices to extract quad patch layout is introduced in section 4. Experiments are shown in section 5, and future work is discussed in section 6.

2. Problem Definitions

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 [14] as depicted in Fig. 2.

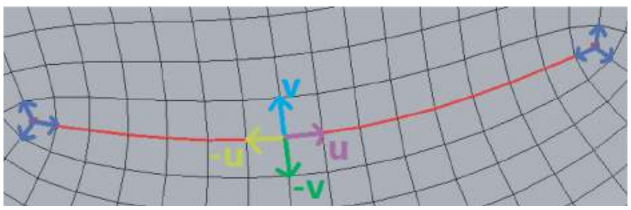

Figure 2. A right-hand local coordinate system with u, v, -u, -v built at a regular vertex. Ports at an irregular vertex are shown as blue arrows. An initial separatrix is shown as a red line.

Port: A port is the outgoing edge adjacent to an irregular vertex (seen in Fig. 2). An irregular vertex of valence n has n distinct ports.

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.

Typically, the edge sequence of an *initial separatrix* (seen in Fig. 2) starts from a port of an irregular vertex, followed by the edge of a regular vertex in the same local parametric direction, i.e., they are either {u,-u} or {v,-v}, and ends at an irregular vertex (maybe the same irregular vertex as the start vertex).

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. Taking a connectivity graph as boundaries, a quad mesh could be segmented into many patches, which means that a connectivity graph is corresponding to a *patch layout*.

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

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.

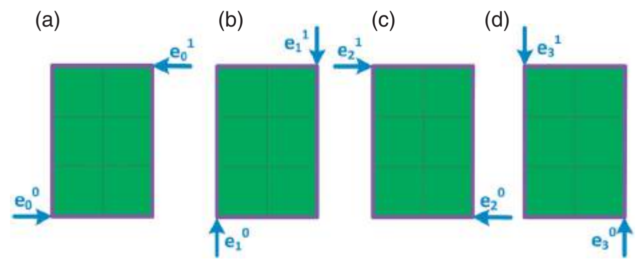

Figure 3. Four types of safety turning areas. In the rectangle area of (a), which is bounded by purple lines, and the diagonal ports (e_0^0, e_0^1) (shown in blue arrows), which have the same local parametric direction and connect the rectangle area at the bottom-left and top-right corners respectively,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

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

For diagonal ports in the quad mesh, if the rectangle areas appear at the left part of its "x-dir", the corresponding safety turning areas are called *left*, otherwise, if the rectangle areas appear at the right, the safety turning areas are called *right*. For example, in Fig. 4, for diagonal ports (e_0^0, e_0^1) and (e_3^0, e_3^1), the safety turning areas are *left*, and for diagonal ports (e_1^0, e_1^1) and (e_2^0, e_2^1), the safety turning areas are *right*.

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*.

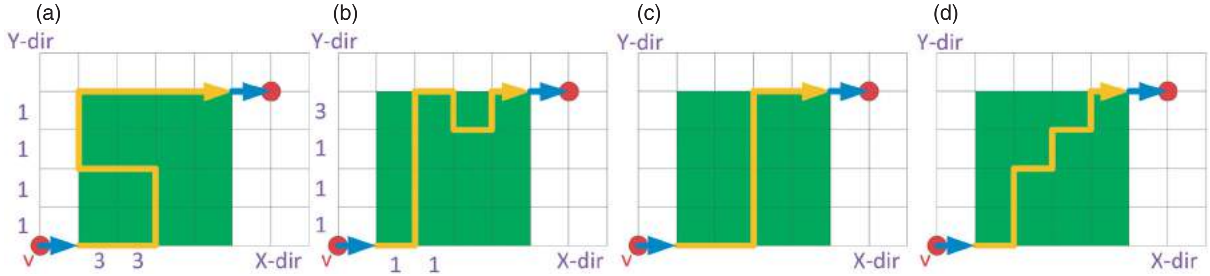

Figure 4. Monotone Separatrix. The red points in the figure are denoted as irregular vertices, and the green quads form the rectangle area of the safety turning area. In (a) and (b), the edges of separatrices (yellow lines) in the rectangle area are not monotone in "x-dir" and "y-dir" respectively. In (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.

- 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, and then the Lwid and Lext of the safety turning area and the length of separatrix can be simply measured by the number of the edges. Take the safety turning area associated to diagonal ports ($e_0{}^0$, $e_0{}^1$) in the Fig. 3(a) as an example, the Lwid is 3 and Lext is 2. For every *initial separatrix*, Lwid is 0.

Theory 1: *In a safety turning area, only the monotone separatrices could be considered as the desired shortest separatrices, and one of the monotone separatrices can be selected to be a candidate for others.*

Proof: In a safety turning area, although there are many separatrices connecting the irregular vertices through the diagonal ports, only the monotone separatrices pass exactly Lwid times of column edges and Lext times of row edges. Since the length of separatrices could not be smaller than is Lwid + Lext, the monotone separatrices could be considered as the desired shortest separatrices. What is more, because all monotone separatrices connect the same irregular vertices through the same ports, and they share a common length, one of them could be selected to be a candidate for others. **End.**

As the definitions above, it is easy to know that the complexity of a patch layout is determined by the number of crossings in its corresponding connectivity graph. A large number of crossings mean a complex patch layout, and a small number of crossings bring in a coarse patch layout (seen in Fig. 1). Meanwhile, a long separatrix has a greater probability of generating a large quantity of crossings, so finding the shortest 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.

3. Safety Turning Area Detection

For a given quad mesh, a naïve method to solve the problem of finding the shortest separatrices is that firstly finding all separatrices and then removing the longer and duplicate ones. However, the number of all separatrices is so huge that they are unable to be enumerated.

Fortunately, according to theory 1, 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 shortest separatrix for all other separatrices. So the problem of finding all separatrices could be considered to be the problem of finding all safety turning areas in the mesh. Since the number of safety turning areas is much smaller than the number of separatrices, it can be greatly reduce the complexity of the problem.

For a given port, based on the definition of safety turning area, we could simply step by step increase the L_{wid} and L_{ext} to find all safety turning areas related to it. In order to further reduce the complexity of the problem, a good condition to stop increasing L_{wid} and L_{ext} is necessary. It can be seen that not all pairs of ports could find safety turning areas associated with them, e.g. Fig. 5. For the port e_0 and other ports (e_1, e_2, e_3), there are three combinations for pairs of ports: {(e_0, e_1), (e_2, e_3)}, {(e_0,

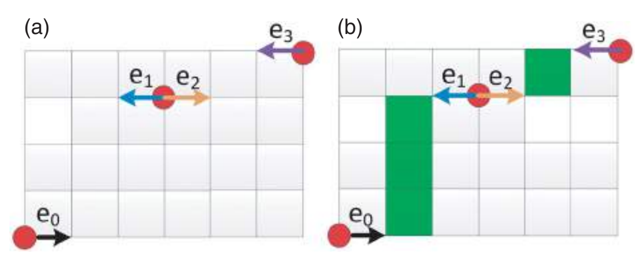

Figure 5. Pair of ports selection. In all sub figures, irregular vertices are shown as red points, rectangle areas of safety turning areas are shown as green areas, and ports are shown as colored arrows. (b) the safety turning areas of pairs of ports {(e_0, e_1), (e_2, e_3)}.

$e_2), (e_1, e_3)\}, \{(e_0, e_3), (e_1, e_2)\}$, but only $\{(e_0, e_1), (e_2, e_3)\}$ could find safety turning areas in Fig. 5(b). Consequently, the condition to stop increasing L_{wid} (or L_{ext}) is to meet another port, which could help us quickly find all safety turning areas. At the same time, the method described above should be used twice for every port, since the safety turning areas are formed by *left* and *right* ones as defined.

It has been proofed that one monotone separatrix can be selected to be a candidate for others in a safety turning area, and what is more, the candidate cannot be simply randomly selected, since a bad selection may bring in non-quad or degenerated patches. As shown in Fig. 6, if two safety turning areas have overlapped areas, bad candidate selections in Fig. 6(c) and (d) bring in bad patches. Generally, in order to avoid these bad selections, we should analyze the safety turning areas to make sure that the candidate monotone separatrices follow requirements as blow:

- If the separatrices could be found without passing through overlapped areas, they are more preferred to be selected as candidates.
- If not, selected the candidate separatrices whose number of crossings is as small as possible.

However, sometimes there may be more than 2 safety turning areas overlapped each other, analyzing them to select good candidates is very difficult. Geodesic path associated the diagonal ports in the safety turning area could help us quickly determine the selection, since they could efficiently avoid unnecessary crossings, seen in Fig. 6(f).

- In most cases, the geodesic paths do not have crossings. Since they will cut the rectangle areas of safety turning areas into several parts, candidate separatrices without crossings can be selected from non-overlapped parts (yellow parts in Fig. 6(e)).
- Otherwise, in the areas which are far away the crossings of geodesic paths, the candidate separatrices pass through the non-overlapped parts, and in the areas around the crossings, we could enumerate all possible situations, and choose the best ones.

4. Binary Integer Programming Problem

As discussed above, the separatrices in a safety turning area share a common length ($L_{wid} + L_{ext}$), so for a separatrix P_i, its energy $En(P_i)$ can be defined in a simple way $En(P_i) = L_{wid}(P_i) + L_{ext}(P_i)$.

However, the candidate separatrices in the safety turning areas turn L_{wid} times in the rectangle areas, and they are misaligned to the local cross field in the quad mesh. In order to punish the misalignment, weighed coefficients are added into the definition of the energy of a separatrix, shown in Eqn. (1).

$$En(P_i) = L_{ext}(P_i) + \alpha L_{wid}(P_i) \quad \alpha \geq 1 \tag{1}$$

Suppose the ports of the quad mesh are $E = \{e_0, e_1, \ldots, e_m\}$, the candidate separatrices are $P = \{P_0, P_1, \ldots, P_n\}$, the energy of the connectivity graph En is Eqn. (2).

$$En = \sum_{i=0}^{n} \delta_i En(P_i) \quad \delta_i \in \{0,\ 1\} \tag{2}$$

Where δ_i is 1 means separatrix P_i is contained in the connectivity graph, while 0 means is not.

Besides minimizing the energy En, there are two constraints should be satisfied.

- Every port should exactly be associated with only one separatrix.
- The extracted patch layout should be quad.

Let $e_i \in P_j$ stands for port e_i is associated with separatrix P_j, then for every e_i, the first constraints could be

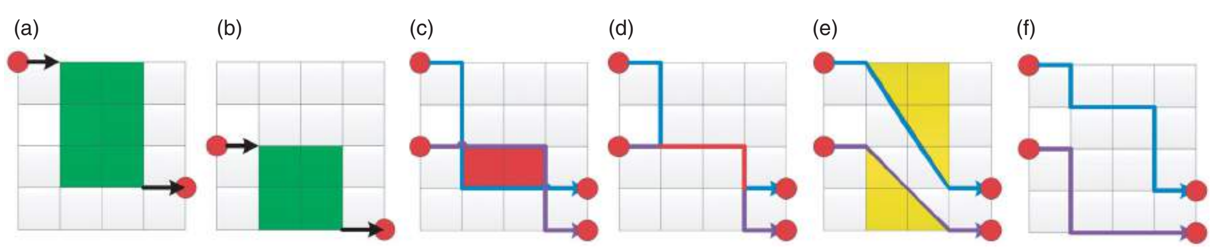

Figure 6. Geodesic Path could efficiently avoid unnecessary crossings. Suppose four irregular vertices exist in an area, and two pairs of them can form safety turning areas. Then one pair is shown in (a), and the other is shown in (b). In (c) and (d), the blue line is the candidate separatrix selected 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 degenerated patch, 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).

formulated as Eqn. (3).

$$\sum_{e_i \in P_j} \delta_j = 1 \quad 0 \le j \le n \tag{3}$$

Although the second constraint is hard to be explicitly formulated, it can be realized by iteratively adding constraints. Assume $\delta^k = \{\delta_0{}^k, \delta_1{}^k, \ldots, \delta_n{}^k\}$ stands for some connectivity graph, and it contains non-quad patches, in order to reject this selection, one more constraint should be added into the problem as Eqn. (4).

$$\sum_{\delta_j^k=1} \delta_j < \sum_{t=0}^{t=n} \delta_t^k \quad 0 \le j \le n \tag{4}$$

Conclude from above, the problem of extracting the quad patch layout forming by shortest separatrices starts from a binary integer programming problem as Eqn. (5).

$$\begin{aligned} \min \quad & En = \sum_{i=0}^{n} \delta_i (L_{ext}(P_i) + \alpha L_{wid}(P_i)) \quad \alpha \ge 1 \\ & st. \sum_{e_i \in P_j} \delta_j = 1 \quad 0 \le j \le n \quad \delta_i \in \{0, 1\} \end{aligned} \tag{5}$$

Checking the optimal solution, if it extracts quad patch layout, the global optimal solution is achieved. Otherwise, one more constraint, shown in Eqn. (4), is added into Eqn. (5), and it will be solved again. The checking and adding constraints are iteratively executed until the global optimal solution is generated. Obviously, more times of iteration, more time will be consumed.

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.

Table 1. Time Consuming.

Model	N_1	N_2	M	T (s)
CubeBlob [Fig. 1]	5600	78	1	0.16
Cup [Fig. 9(b)]	14983	23	1	0.15
Igea [Fig. 9(a)]	8183	28	2	0.25
Rockarm [Fig. 9(c)]	4524	101	2	0.24
Joint [Fig. 9(d)]	498	52	3	0.34

5. Experiments

In our experiments, the input quad meshes can be generated by mixed-integer quadrangulation [1], which results uniform edge length as required by our method.

The environment of our experiments is Intel(R) Core(TM) i5 CPU @ 3.00GHZ and 3.2GHZ, 16G RAM Memory and the binary integer programming optimizer are provided by function “bintprog” in Mosek.

Fig. 7 shows a comparison result between our method and [15], and the number of patches generated by us is smaller than them. Compared with their method, we use a similar energy definition for separatrices, but based on the safety turning area, we make enumerating all the

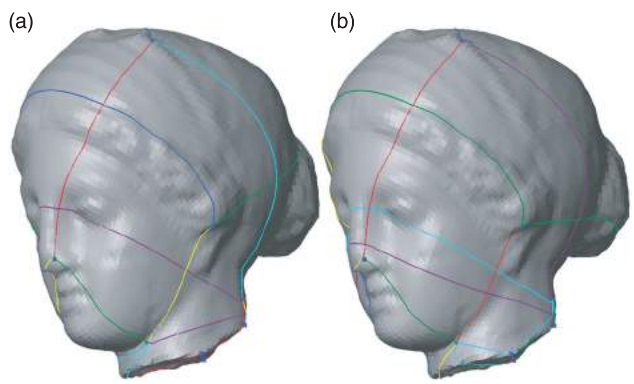

Figure 8. Two results have the same smallest energy.

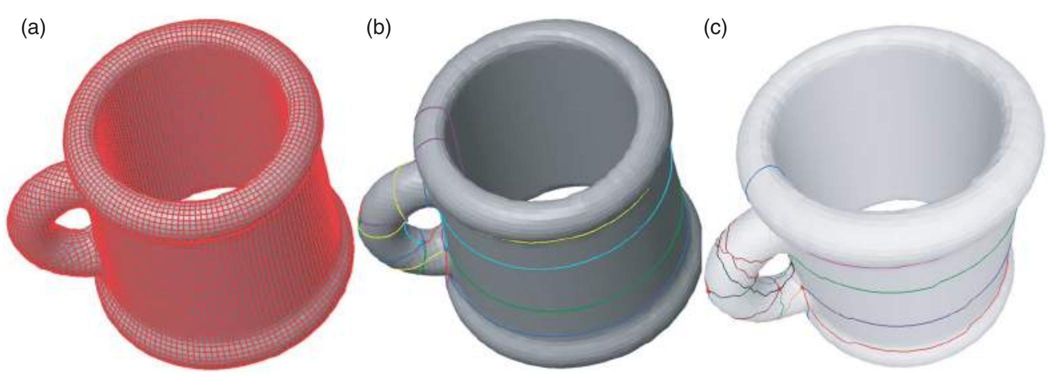

Figure 7. Comparison. Red lines in (a) are initial separatrices and color lines in (b) and (c) are the separatrices selected respectively by our method and method in [15]. In (b), the separatrices are shown by the geodesic path in their corresponding safety turning areas, and the number of quad patches is 23. In (c), the separatrices are shown by the edge sequences on the mesh, which bring in unsmooth separatrices, and the number is 25. The result in (c) may fall into a local optimal solution.

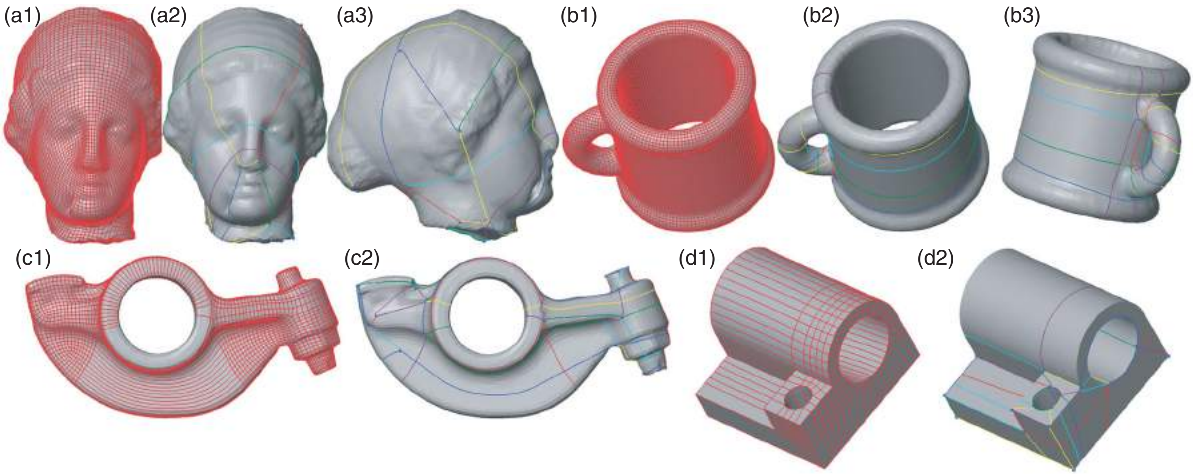

Figure 9. More results of quad layout extraction. The boundaries of the desired patch layout are shown in colored lines and they are all the geodesic paths in *safety turning areas*.

shortest separatrices possible. And then by solving the binary integer programming problem, the final solution could be guaranteed to be the global optimal solution. It overcomes the main shortcomings of the greedy strategy used by others, which may fall into obtaining local optimal solution.

Another interesting result is shown in Fig. 8, in which two different quad patch layouts are extracted, and at the same time, they have the same smallest energy.

Some models with quad patch layout are shown in Fig. 9.

6. Conclusion and Future Work

In this paper, an automatic method is proposed to extract wells-shaped quad patch layout with monotone boundaries in quad meshes. Based on safety turning area, all shortest separatrices in the quad mesh could be found at first, and then the global optimal solution could be achieved by a binary integer programming optimizer, which efficiently avoid the problem falling into obtaining local optimal solution. At the same time, with the help of the geodesic path, we are able to quickly determine whether a solution of the binary integer programming problem can extract quad patch layout.

However, as the binary integer programming problem is NP-Hard, and that the quad topology could not be explicitly formulated into the problem makes us use binary integer programming solver more than once, time consuming becomes the main limitation of our method. And what is more, in some complicated cases, the number of candidate separatrices is still large, which may also bring in more time consuming.

Consequently, the main future work is to speed up the convergence efficiency, or to find a proper constraint of quad topology which can be added into the binary integer programming problem explicitly. At the same time, how to extend our method to deal with open quad meshes and introduce more constraints for measuring the quality of the patch layout is also our future work.

Acknowledgements

This work was supported by the National Key Technologies R&D Program of China (2015BAF23B03, the National Nature Science Foundation of China (61373070, 61272235), International Science & Technology Cooperation Program of China (2013DFE13120), and Tsinghua University Initiative Scientific Research Program (2012Z02170).

ORCID

Sen Zhang http://orcid.org/[0000-0002-7804-1457]
Hui Zhang http://orcid.org/[0000-0001-6563-9890]
Jun-Hai Yong http://orcid.org/[0000-0002-4326-4167]

References

[1] Bommes D.; Zimmer H.; Kobbelt L.: Mixed-Integer Quadrangulation, *ACM Transactions on Graphics*, 28(3), 2009, 77. http://dx.doi.org/10.1145/1531326.1531383
[2] Bommes D.; Lempfer T.; Kobbelt L.: 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
[3] Bommes D.; Levy B.; Pietroni N.; Puppo E.; Silva C.; Tarini M.; Zorin D.: Quad-Mesh Generation and Processing: A Survey, *Computer Graphics Forum*, 00(0), 2013, 1–26. http://dx.doi.org/10.1111/cgf.12014
[4] Bommes D.; Campen M.; Ebke H. C.; Alliez P.; Kobbelt L.: Integer-Grid Maps for Reliable Quad Meshing, *ACM Transactions on Graphics*, 32, 2013, 4. http://dx.doi.org/10.1145/2461912.2462014

[5] Benko P.; Varady T.: 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
[6] Cohen-Steiner D.; Alliez P.Desbrun M.: Variational Shape Approximation, *Computer Aided Geometric Design*, 23(3), 2004, 905–914. http://dx.doi.org/10.1145/1186562.1015817
[7] Campen M.; Bommes D.; Kobbelt L.: 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
[8] Eppstein D.; Erickson J.: 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] Eppstein D.; Goodrich M. T.; Kim E.Tamstorf R.: 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
[10] Gunpinar E.; Suzuki H.; Ohtake Y.; Moriguchi M.: Generation of Bi-monotone Patches from Quadrilateral Mesh for Reverse Engineering, *Computer-Aided Design*, 45, 2013, 440–450. http://dx.doi.org/10.1016/j.cad.2012.10.027
[11] Li Y. F.; Wang W. P.; Ling R. T.; Tu C. H.: Shape Optimization of Quad Mesh Elements, *Computers and Graphics*, 35, 2011, 444–451. http://dx.doi.org/10.1016/j.cag.2011.03.037
[12] Myles A.; Pietroni N.; Kovacs D.; Zorin D.: Feature-aligned T-meshes, *ACM Transactions on Graphics*, 29(4), 2010, 117. http://dx.doi.org/10.1145/1778765.1778854
[13] Peng C. H.; Zhang E.; Kobayashi Y.; Wonka P.: Connectivity Editing for Quadrilateral Meshes, *ACM Transactions on Graphics*, 30(6), 2011, 141:1–141:12. http://dx.doi.org/10.1145/2070781.2024175
[14] Peng C. H.; Wonka P.: Connectivity Editing for Quad-Dominant Meshes, *Computer Graphics Forum*, 32, 2013, 43–52. http://dx.doi.org/10.1111/cgf.12171
[15] Tarini M.; Puppo E.; Panozzo D.; Pietroni N.; Cignoni P.: Simple Quad Domains for Field Aligned Mesh Parameterization, *ACM Transactions on Graphics*, 30(6), 2011, 142. http://dx.doi.org/10.1145/2070781.2024176
[16] Vieira M.; Shimada K.: 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
[17] Wu J. H.; Kobbelt L.: 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