EUROGRAPHICS 2011 / M. Chen and O. Deussen
(Guest Editors)

Volume 30 (2011), Number 2

Global Structure Optimization of Quadrilateral Meshes

David Bommes Timm Lempfer Leif Kobbelt

RWTH Aachen University

Abstract

We introduce a fully automatic algorithm which optimizes the high-level structure of a given quadrilateral mesh to achieve a coarser quadrangular base complex. Such a topological optimization is highly desirable, since state-of-the-art quadrangulation techniques lead to meshes which have an appropriate singularity distribution and an anisotropic element alignment, but usually they are still far away from the high-level structure which is typical for carefully designed meshes manually created by specialists and used e.g. in animation or simulation. In this paper we show that the quality of the high-level structure is negatively affected by helical configurations within the quadrilateral mesh. Consequently we present an algorithm which detects helices and is able to remove most of them by applying a novel grid preserving simplification operator (GP-operator) which is guaranteed to maintain an all-quadrilateral mesh. Additionally it preserves the given singularity distribution and in particular does not introduce new singularities. For each helix we construct a directed graph in which cycles through the start vertex encode operations to remove the corresponding helix. Therefore a simple graph search algorithm can be performed iteratively to remove as many helices as possible and thus improve the high-level structure in a greedy fashion. We demonstrate the usefulness of our automatic structure optimization technique by showing several examples with varying complexity.

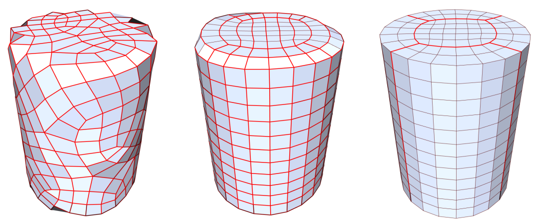
Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Hierarchy and geometric transformations, Curve, surface, solid, and object representations

1. Introduction

For more sophisticated geometric modeling and processing applications like, e.g., CAD/CAM and numerical simulation, quad meshes are often preferred over triangle meshes. However, the generation and handling of quad meshes is significantly more difficult due to the anisotropic nature of quadrilaterals. While for high quality triangle meshes it is usually sufficient to have a fairly regular vertex distribution, good quad meshes have additional orientation and consistency constraints to satisfy.

In fact the optimization of quad meshes is an inherently global problem since local changes in the mesh structure usually propagate globally across the mesh. This is not the case for triangle meshes where mesh optimization can be performed based on local operations.

Recently strong methods for the generation of quad meshes have been proposed which yield meshes with good orientation and alignment properties. However, while the resulting meshes are looking quite pleasing at the first glance,



(a) (b) (c)

Figure 1: *Comparing different structural quality: (a) A completely unstructured mesh with bad quads and a dense base-complex (in red). (b) Appropriate singularities and oriented quads improve the mesh, but due to a quad-loop winding down the cylinder the base-complex is still dense. (c) While preserving singularities and orientations, the base-complex is optimized and topologically equivalent to a cube*

a more careful analysis of their global structure reveals that they still do not exhibit a patch layout as it is known from meshes emerging from 3D modeling systems. In practice such a clean patch layout is highly desirable to support standard operations like e.g. texturing, NURBS fitting or adaptive sizing.

The major reason for this imperfection is that mesh singularities are usually placed based on geometric considerations but otherwise fairly independently from each other. An important consequence of this is that geodesically neighboring singularities are not properly connected to form a nicely shaped patch layout which would correspond to a coarse base complex. The most noticeable phenomenon where the lack of consistency in the global structure can be observed, is the occurrence of helical configurations (see Figure 1(b)).

In this paper we are proposing an algorithm that takes such an automatically generated quad mesh as input and converts it into a quad mesh with improved global structure. This improvement is entirely based on an optimization of the discrete graph structure of the mesh and not on the continuous optimization of the geometric embedding.

In particular, we are presenting a very general operator which changes the quad mesh structure without affecting the number and distribution of singularities and without introducing non-quad faces. Nevertheless the operator is flexible enough to significantly modify the global structure of the quad mesh.

Based on this operator we develop a simple greedy procedure to effectively remove helical configurations from the input mesh and thus improve the global structure.

1.1. Related work

Quad-remeshing techniques have a long tradition within the graphics community and nice surveys exist [AUGA05]. Early works tried to generate oriented quadrilateral elements by explicitly tracing lines along the principal curvature directions [ACSD*03, MK04], resulting in quad-dominant meshes. More recent parametrization based techniques are very successful in generating curvature oriented all-quadrilateral meshes [RLL*06, KNP07, HZM*08, BZK09, ZHLB10]. They are able to automatically find adequate singularity positions, e.g. by non-linearly smoothing the cross field induced by the principal curvature directions [KNP07], optimizing a non-linear objective function [RLL*06, HZM*08, ZHLB10] or solving a mixed-integer problem [BZK09]. Typically these methods can generate quadrilateral meshes with a nice angle and edge-length distribution as well as adequate singularities. However, the quality of the induced base-complex is often not sufficient, motivating our base-complex optimization technique. All input meshes used in our paper were generated with methods from this class.

Instead of using the principal curvature directions as a guiding, another class of algorithms directly exploits a base-complex with specified topology to generate all-quadrilateral meshes via a global parametrization. Here the base-complex is constructed manually [TACSD06, BVK08] or derived automatically from a Morse-Smale complex [DBG*06]. However, automatically constructing high-quality base-complexes comparable to manually designed ones is still an open problem.

Recently the generation of feature-aligned T-meshes was studied [MPKZ10]. Although completely different techniques are used to generate a different mesh type, this work has similar intentions as ours. Instead of removing helices, Myles et al. try to align singularities directly within the parametrization to simplify the base-complex. We found that automatically choosing singularity pairs for alignment often is complicated or even ambiguous and can lead to contradicting conditions. However, since misaligned singularities lead to helical structures within the quadrangulation, we concentrated on removing them instead and thus implicitly align appropriate pairs of singularities without the necessity to resolve the ambiguities.

Quad mesh decimation techniques transform a high-complexity quadrilateral mesh into a low complexity one by incrementally applying a set of discrete operators.

Stimulated by the work of Daniels et al. [DSSC08] different operators and objective functions were developed, which e.g. interleave poly-chord collapse and quadrilateral collapse [DSSC08] (a.k.a. quadclose [Kin97] or quad-vertex merge [DSC09a]), use only localized operations [DSC09a], apply ring collapses [SDW*10] or maximize homeometry through an extended set of local operators combined with tangential smoothing [TPC*10].

The above methods have proven to be very useful in improving unstructured quadrilateral meshes, convert triangle meshes in quadrilateral meshes with good individual element quality or even building level of detail hierarchies [DSC09b]. However they are not designed to work on structured quadrilateral meshes with singularities placed by a global optimization method. The inherent problem is that despite the global poly-chord collapse, all of those local operators introduce singularities when applied on a regular grid. Consequently for highly structured inputs the optimization driven by the local operators gets stuck in local minima leaving singularities at unexpected positions, instead of the desired behavior of generating adequate chains of operations to preserve the original singularities.

Although the base-complex corresponds to a simplified version of the input quad mesh, we do not address the quad mesh decimation problem. Our main goal is to simplify the base complex as a means to global structure optimization not necessarily the quad mesh itself. For this task different operators are necessary which are designed to preserve the regular grid structure without introducing additional singularities.

2. Structure Optimization

2.1. Definitions

A quadrilateral mesh $M = (V, E, Q)$ is a set of vertices $v_i \in V$ embedded in $\mathbb{R}^3$ with adjacency information encoded as edges E and quadrilaterals Q.

A vertex v_i is called *regular* if it has valence 4 otherwise it is a *singular* vertex. Topologically a regular vertex is the crossing of two coordinate lines in a 2D Cartesian grid and therefore we can easily build a right-handed local coordinate system at such a vertex by cyclically labeling the adjacent edges in counter-clockwise order with u, v, $-u$ and $-v$ as depicted in Figure 2. However, notice that such a labeling is only possible within a singularity-free local region since e.g. walking counter-clockwise around a valence 3 singularity would mean that a formerly labeled u edge becomes a v edge contradicting with the initial label.

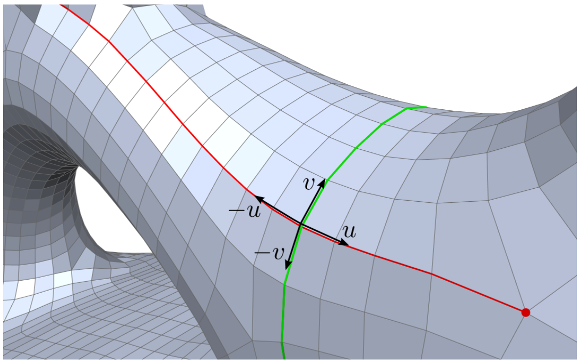


Figure 2: *Each regular vertex induces a natural coordinate system by counter-clockwise labeling the outgoing edges with $u, v, -u, -v$. Parametric lines, as shown in red and green can be extended until they end in a singularity (red point).*

A *parametric line* is generated by tracing a local coordinate direction through regular vertices or more formally a connected sequence of edges, such that two subsequent edges e_i and e_j are always connected through a regular vertex where both edges belong to the same local parametric direction, i.e. they are either $\{u, -u\}$ or $\{v, -v\}$ (see Figure 2). Finally a *regular parametric loop* is a closed parametric line where all traversed vertices are regular.

We will also use the common notion of the dual mesh $M^d = (V^d, E^d, Q^d)$ where an isomorphism identifies each dual vertex with the centroid of a primal face, each dual edge with a primal edge rotated counter-clockwise by $\frac{\pi}{2}$, each dual face with a primal vertex and the adjacency information is automatically inherited from the primal mesh by the above isomorphism. An important property of the dual of a quadrilateral mesh is that all vertices are regular (valence 4). This property is the reason why using dual parametric lines instead of primal ones is advantageous. For each primal parametric line we can always identify two parallel dual parametric lines, while the contrary is not always true due to the fact that primal parametric lines end at singularities. Consequently, using dual parametric lines or dual *parametric loops*, which are *quad-loops* in the primal meshes and called *poly-chords* in [DSSC08], increase the set of candidates for our grid preserving operator (see Section 2.3).

2.2. The Base Complex

In the following sections we will propose an algorithm to improve the *base complex* $\mathcal{B}(M) = (\mathcal{V}, \mathcal{E}, \mathcal{Q})$ of a given quadrilateral mesh which is also a quadrilateral mesh with $\mathcal{V} \subset V$. The base complex is the union of all parametric lines which start and end at singular vertices. Figure 1 shows three different quadrilateral meshes, where the base complex is highlighted in red. An inverse way of constructing the base complex is to iteratively remove all regular parametric loops. Each of these steps corresponds to the merging of neighboring quad-loops. This modification obviously preserves the quadrangular structure of the input, which proves that the base complex of a quadrilateral mesh is guaranteed to be a quadrilateral mesh as well.

Providing a high-quality quadrilateral mesh with a coarse base complex is of great interest, since a coarse base complex induces a simple patch layout which is desired for e.g. fitting of NURBS-patches or as a base mesh for subdivision. In general, computing quadrangulations which provide on the one hand a nice stretch distribution in terms of angles and anisotropic edge lengths and on the other hand a coarse base complex is an unsolved problem. Parametrization based techniques usually lead to nicer stretch distributions due to well adapted singularities and edge orientations, but unfortunately they often posses a rather fine base complex. On the contrary decimation based algorithms are able to generate coarse base complexes, however, this benefit usually comes at the cost of inappropriate placed singularities or edge orientations, inducing high stretch distributions in a finer subdivision.

Our strategy is to start with a quadrangulation already equipped with appropriate singularities and a nice stretch distribution and then try to improve the base complex as much as possible while keeping the singularities fix. Notice that apart from the base complex there is no other straightforward coarse quadrangulation with the same singularities as in the input mesh. Due to the global topological restrictions we cannot define a concept analogous to the Delaunay triangulation to achieve a coarse quadrangulation of the singularities.

In the next section we will propose a novel operator which is fundamental for our base complex optimization, since it offers a new class of global operations which preserve quadrilaterals and are optionally able to preserve singularities.

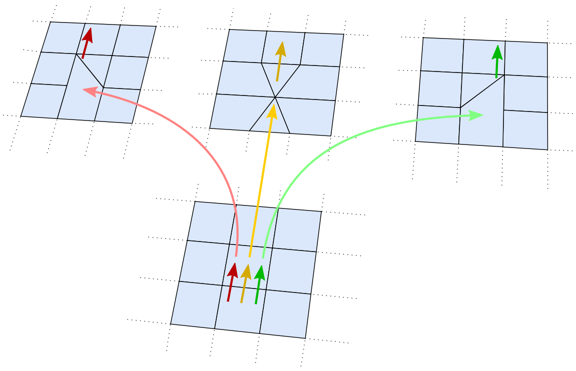
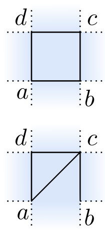


Figure 3: *The three atomic operations of a dual half-edge: A* shift left *step (red arrow) releases the vertex on the left side of the dual half-edge and shifts it towards the next vertex, generating a triangle and a pentagon. In a* collapse *step (yellow) the edge is* collapsed *into a single vertex. The* shift right *(green) is the counterpart of* shift left*, releasing the right vertex. After applying one step we move to the next dual halfedge as indicated by the arrows.*

2.3. Grid-Preserving Operators

Changing the local connectivity within a quadrilateral mesh without introducing non-quadrilateral elements or new singularities is a delicate task. And even worse, no local operation exists to perform such a modification. However, since such an operation is highly desirable, it is worth to examine the problem in more detail.

Assume that we have a closed quadrilateral mesh without boundaries and that we want to change the connectivity within a single quadrilateral with points a, b, c and d such that a is connected to c instead of b, as depicted in the figure to the right.



The problem is that after executing this edge-flip, we end up with a triangle and a pentagon. If the corresponding quad-loop is self-intersection free, one solution would be to propagate the edge-flip along the whole (always closed) quad-loop such that in the end the triangle and the pentagon will cancel out. Unfortunately not all quad-loops are intersection free and even if they are, this combined operation is completely determined by the quad-loop structure and leaves no freedom to control which areas of the mesh should preferably be modified. This property is in conflict with the requirement to protect parts of the mesh which contain important features or regions of good quality.

To obtain more degrees of freedom we propose to combine the above edge-flip operation with a collapse operation in such a way, that we can create a much larger variety of possible operators, but still can guarantee to preserve the quadrangular structure of the input mesh. Figure 3 shows the three necessary atomic operations, namely *shift left*, *collapse* and *shift right* which are visualized with a red, yellow and green arrow respectively. All three operations can be associated with a dual halfedge and combined along a dual path in the way shown in the finite automaton in Figure 4 in order to form a valid grid-preserving operator (*GP-operator*). The most important property of such a *GP-operator* is that it does not introduce new singularities or non-quadrilateral elements.

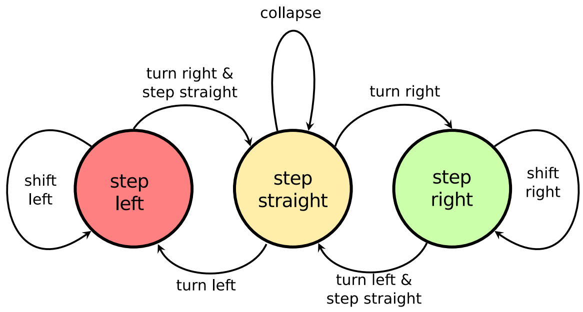


Figure 4: *The finite-automaton describes all valid possibilities to combine the three atomic operations. Each closed dual path on the mesh, which is closed within the the finite-automaton preserves the all-quadrilateral structure without introducing new singularities.*

This means, if we start at one mesh edge in the *step left* state we can do as many *shift left* steps as we want by following the dual path in the same direction where all crossed edges are shifted. To leave the *step left* state, within a face we can turn right and change the state to *step straight*. From here we can either move straight and collapse as many edges as desired, or turn right and apply the *shift right* operator, or again turn left and apply the *shift left* operator. Altogether, using this state machine, we can traverse a dual path which is assembled of straight steps, sidesteps to the left and sidesteps to the right, but we can never step back, i.e. turn twice into the same direction (cp. Figure 5).

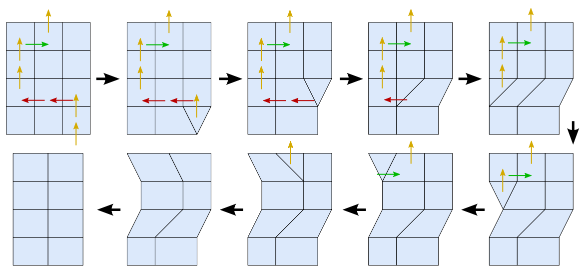
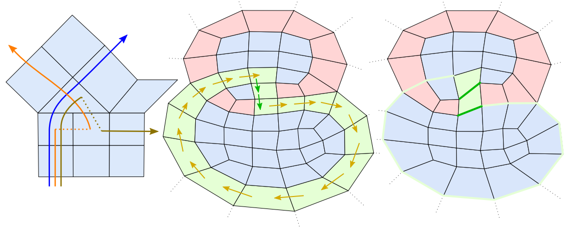


Figure 5: *Example of a valid dual path combining the three atomic operations according to the state machine. In the absence of singularities the resulting topology is equivalent to the removal of a single column of quads (cp. last step).*

While this might seem to be quite restrictive it fortunately is not. The reason is that we can exploit the singularities



(a) (b)

Figure 6: *(a) Side steps (dashed lines) can control the walking direction by navigating between singularities. (b) The dual path through the green quadrilaterals, consisting of* collapse *steps (yellow) and* shift right *steps (green), is a valid GP-operator (left). Executing the corresponding atomic operations results in a new quadrilateral mesh with the same singularities (right). Notice that the GP-operator has closed the red quad-loop.*

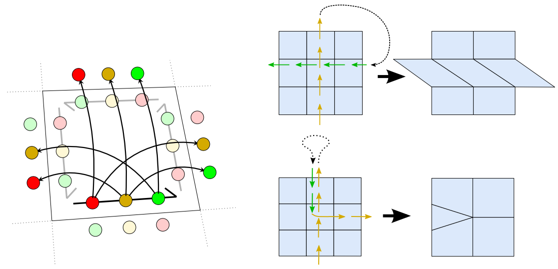
within the mesh to change the walking direction, e.g. walking around two valence three singularities is the same as turning by an angle of π in a regular grid. Consequently, navigating between and around the singularities offers a large variety of possible paths. Figure 6 (a) gives an example of this behavior.

To guarantee that in the end all triangles and pentagons cancel out, it is necessary that the dual path is closed within the state machine, meaning that there is a transition from the state at the last dual half-edge to the state at the first dual half-edge. Notice that this is exactly the case when the closed dual path circuits a group of singularities such that the total rotation becomes an integer multiple of 2π. For illustration, Figure 6 (b) shows such a path and the resulting quadrangulation after applying the corresponding operations.

Going back to our introductory question, we are now able to give a more satisfying answer. If we want to shift the edge between a and b to an edge between a and c while maintaining a quadrangulation without additional singularities, we can start at the edge between a and b with the state *shift right* and walk along any closed dual path compatible to the state machine and perform the induced atomic operations. Which one of those candidate operations is the best strongly depends on the application in mind.

A natural choice is to minimize the overhead, i.e. the number of additional atomic operations which are necessary to close the path. This can be found by enumerating all possible paths generated by the state-machine with increasing length until the shortest cycle is found. Obviously this approach leads to an exponential complexity which is useless for practical applications.

The state-machine graph: In order to efficiently find a cycle which is compatible to the state machine, we first assemble a directed graph, as depicted in Figure 7 (a). In this graph all cycles are compatible with the state-machine by



(a) (b)

Figure 7: *(a) Illustration of the state-machine graph: By creating three vertices for each dual half edge we can encode the different states* shift left *(red),* collapse *(yellow) and* shift right *(green). Adding directed edges corresponding to transitions within the finite-state automaton we obtain a graph where all paths that belong to chains of operations are compatible with the finite-state automaton by construction. (b) The upper part of the figure shows a valid while the lower one depicts an invalid crossing configuration.*

construction. The idea is that the graph possesses three different vertices for each dual halfedge of the quadrilateral mesh which encode the three different states. Adding directed edges which reproduce the transitions of the state machine as illustrated in Figure 7 (a) we achieve a directed graph with the desired property. All cycles in this graph correspond to dual paths on the quadrilateral mesh which are closed within the mesh as well as in the state machine.

In this graph a shortest cycle through a start vertex can be found by a simple and efficient breadth-first search. However there is one drawback compared to the explicit exponential algorithm of the state-machine. Since the graph is static, it does not capture the changes made by previous operations of the same path. Clearly we cannot *shift* an edge which was already collapsed, although such a path exists in the graph. Therefore we have to do a post-evaluation of the cycle in order to check whether it belongs to a realizable set of operations or not. If it is not realizable, we iteratively modify the graph and perform new searches, until we have found a valid cycle or the algorithm terminates without finding one. In contrast to the breadth-first search the iterative process cannot guarantee to find a shortest path. However, as our practical experiments showed, it is at least a good compromise between quality and performance.

Illegal configurations within a cycle are typically induced by a corresponding dual path on the quadrilateral mesh that visits a face more than once, e.g. by performing more than one operation on a single edge or first shifting an edge and then performing any other operation while walking through the face. The only two exceptions where it is allowed to visit a face twice are first collapsing through a face in two orthogonal directions and second collapsing through a face in one direction and then shifting through the face in the orthogonal direction. In both cases the static graph structure still leads to valid paths.

If an illegal cycle is found, we first identify the first illegal configuration where a face is visited twice, leading to a pair of graph vertices v_i and v_j, which are in conflict by visiting v_i first. To modify the graph, we remove all graph vertices and adjacent edges which are incompatible for the path up to vertex v_i and then restart the search from v_i.

Feature and singularity preservation: A nice property of the graph representation is that we can exclude all unwanted atomic operations by simply removing the corresponding graph vertex and all its adjacent edges from the graph. This is for example useful to disallow the merging of neighboring singularities or the shifting of feature edges.

Moreover it is possible to disallow the merging of singularities which are not directly connected. Such a merging could possibly happen if the breadth-first search leads to a cycle which collapses several edges connecting two singularities. Of course we do not want to forbid the collapse of all edges between the singularities. Therefore such illegal configurations are identified in the post-evaluation phase and as before we restart the search with a modified graph, where the last collapse leading to the illegal merge was removed. The same procedure can be used to prevent that two distinct feature lines collapse into one.

Using GP-operators: In summary the concept of a GP-operator offers a variety of different structural modifications, which by construction do not introduce new singularities or non-quadrilateral elements. Notice that the well-known poly-chord collapse used in [DSSC08] is one special case of a GP-operator which only consists of edge collapses.

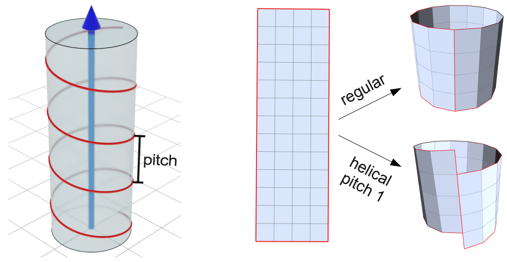
Here we suggested to extend a desired local operation to a full GP-operator by the minimal number of additional operations. However, depending on the desired structural optimization many other choices are conceivable, leading to other graph search algorithms like e.g. a Dijkstra or Hamiltonian cycle.

A nice feature of the graph based construction is the flexibility to optionally guarantee the preservation of singularities and/or (sharp) features of the input quadrangulation by just removing some of the graph vertices.

In the following sections we will use GP-operators to improve the quality of the base complex by identifying and repairing helical mesh configurations.

2.4. Topological Helices in Quadrilateral Meshes

The most intuitive way to think of topological helices in quadrilateral meshes, which we will call *q-helices*, is, to imagine their construction out of a rectangular part of the Cartesian grid as illustrated in Figure 8 (b). First start creating a cylinder in the usual way, by keeping one side of the rectangle fixed in space, wrapping the opposite side of the rectangle around the first one and glueing together pairs of boundary vertices which belong to equal parametric lines.



(a) (b)

Figure 8: *(a) A left-handed helix winds up the blue axis. (b) By wrapping a rectangular quad-patch and and glueing two sides, we can create a cylinder. Shifting the sides against each other before glueing, we end up with a topological q-helix equipped with the same properties as in the continuous case.*

If we instead connect vertices from different parametric lines of the rectangle, we are able to create a single new parametric line, which winds upwards or downwards in the grid with a constant orthogonal offset. Hence, we have constructed a discrete helical structure. In this structure we can identify all the properties of a usual helix. The *pitch* h of the helix is is the distance between two neighboring windings, while the *turn length* τ is the arc length of a single turn. For a q-helix both values are integers, since all distances are measured in the grid-metric of the quadrangular mesh, which means that all edges (and dual edges) have a length of one. The winding number γ which counts the number of turns can be computed by dividing the total length l by the length of one turn $\gamma = l/\tau$.

The *orientation* of a helix is reflected in the sign of the pitch. Following the right hand grip rule, a right-handed helix has a positive pitch, while the pitch of a left-handed helix is negative. Notice that the handedness of a helix is an intrinsic geometric property and does not depend on the chosen coordinate system.

After describing the construction of q-helices, in the next paragraph we will derive a criterion which can be used to identify q-helices in quadrangular meshes. Some example helices are shown in Figure 9.

As discussed in the previous section, we want to work with helices of the dual mesh. More precisely a q-helix $H^d = [e^d_0, \ldots, e^d_n]$ with pitch $h \in \mathbb{Z}$, turn length $\tau \in \mathbb{Z}$ and winding number $\gamma \in \mathbb{R}$ is an ordered set of connected dual edges e^d_i forming a dual parametric line and fulfilling the following q-helix property:

Within a q-helix it is equivalent to either walk τ steps along the helix or alternatively do h side-steps to the left.

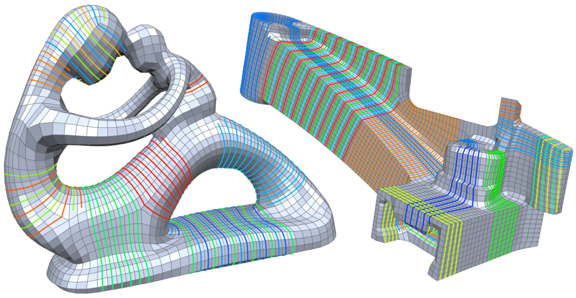


Figure 9: *Exemplarily some q-helices are shown as colored dual parametric lines. Notice that q-helices with a pitch greater than 1 often form bundles of interleaved helices.*

Here equivalent means that not only the position but also the orientation coincides.

Mathematically the above statement implies that there is a regular region without interior singularities around H^d, where it is possible to choose a consistent frame. Consequently q-helices cannot have any self-intersections.

For our mesh optimization task not all helices which fulfill the above definition are of interest. Therefore it is useful to define the interesting subset to be so called *minimal q-helices*. They are characterized through two properties: For a minimal q-helix the pitch h is always smaller or equal to the turn length τ and secondly there is no subset of dual edges belonging to the q-helix, which form a separate q-helix with smaller pitch. The first criterion excludes approximately half of all q-helices, because for each q-helix there exists an orthogonal q-helix living in the same regular region, where the values of pitch and turn length are exchanged. The second criterion excludes helices which contain other helices with smaller pitch, not well suited for our optimization.

As illustrated in the introductory example in Figure 1, q-helices subdivide the base-complex into narrow stripes. Therefore in the next section we will discuss how to remove them from the quadrilateral mesh.

2.5. Removing q-helices

To remove a q-helix we can apply exactly the inverse operation of the construction example of Figure 8 (b), which means cutting the mesh along the helix, shifting the vertices of one side of the cut, and glueing them with their new partners. However, on a closed mesh the situation is a little bit more complicated. In order to preserve the quad-structure we have to compute a full GP-operator, as introduced in Section 2.3, where the desired shifting operations are a sub-part of the complete operation. Furthermore we have to make sure that no other shifting of horizontal edges within the cylindrical mesh area of picture Figure 8 (b) are done by the GP-operator. Since the graph construction of the GP-operator does not allow multiple operations on a single edge, we repair helices with pitch > 1 iteratively by applying the following algorithm.

Removing a q-helix H with pitch 1 can be done in four steps.

1. Set up the graph G representing the state-machine for the input quadrilateral mesh.
2. Identify an open dual path $D = [d_0, \ldots, d_m]$ consisting of shift steps which are necessary to remove the helix.
3. Remove all vertices from G which correspond to shift operations which are in conflict with the correction of D, i.e. all shifting steps of horizontal edges in the cylindrical region which do not belong to D.
4. Execute the iterative path search described in Section 2.3 from vertex d_m to vertex d_0 in G to extend D to a GP-operator. If such an operator exists, perform the induced atomic operations. Otherwise it was not possible to remove H.

In general we have different possibilities to choose the correction path D. Each possibility is a column of quadrilaterals within the cylindrical region. We randomly choose one of those candidates and only in cases where we do not find a path, we iteratively test the other ones.

3. Algorithm

Given the above GP-operator to remove a single q-helix, it is straightforward to design a greedy algorithm which removes as many q-helices as possible.

An important side condition within this algorithm is that we forbid all operations which worsen mesh areas which have a nice topological structure. More precisely we identify all dual edge-loops without self-intersections, i.e. all minimal q-helices with pitch 0, and only search for GP-operators which do not destroy them by shifting a neighboring parallel edge. Furthermore we disallow increasing the pitch of all present q-helices with a winding number greater or equal 2. This somehow arbitrary choice is justified by the observation that helices with at least two complete windings most likely increase the base-complex quality and therefore it is often advantageous to protected them from worsening. Both modifications can easily be done by removing graph vertices as explained in section 2.3.

Altogether our base-complex optimizing greedy algorithm works in the following way (cp. Figure 10):

1. Identify all minimal q-helices $\{H_i\}$ within the input quadrilateral mesh.
2. Greedily remove the helix with the largest winding number with the algorithm explained in Section 2.5.
3. Apply smoothing to reduce the geometric distortion introduced by shift steps.
4. Go back to step 1. until there is no removable q-helix left.

A naive search for q-helices would first check for all dual

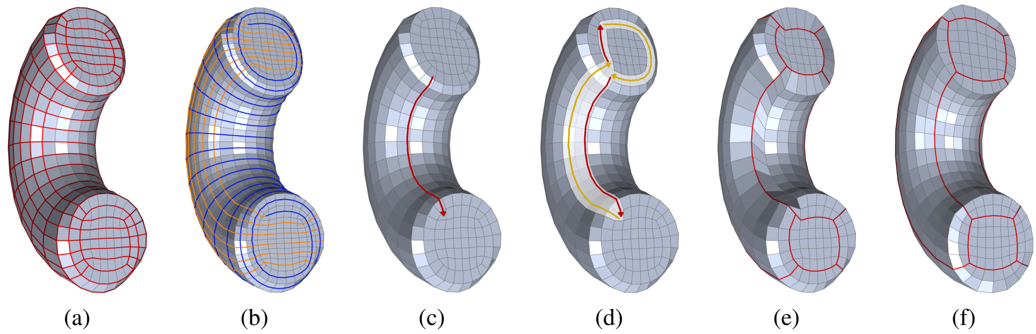


Figure 10: *Algorithm example: Figure (a) shows the input mesh with a fine base-complex. Two q-helices (blue and yellow curve) are identified in (b) and the correction path shown in (c) and belonging to the blue helix is extended to a GP-operator in (d). Figure (e) shows the mesh after applying all induced atomic operations. This single operation is sufficient to remove both helices leading to the desired (coarse) base complex. Finally tangential smoothing improves the per element quality (f).*

vertices whether their orthogonal dual parametric lines intersect each other. If this is the case, the first intersection is a q-helix candidate and we can verify whether the necessary conditions of Section 2.4 are fulfilled and extend the q-helix in both directions as far as possible. By precomputing for each dual half-edge the corresponding parametric line, a local position index on this line and the next self-intersection on this parametric line, the detection of q-helices becomes much faster.

For the smoothing we apply a very simple explicit variant of [ZBX05] as done in [DSSC08] which is able to handle features appropriately. In general it would be possible to leave this step out, however shift operations will locally create unaesthetic angles. Therefore if not only the topological result is of interest, a tangential relaxation is preferable.

	Input			Output		
Model	#Hel	#F	#F in BC	#Hel	#F	#F in BC
FANDISK	19	764	408	5	506	144
DRILLHOLE	24	3077	1368	7	1948	216
ROCKERARM	17	3180	1226	3	1678	178
FERTILITY	46	3357	2271	1	2387	526
BOTIJO	42	8395	4957	7	5472	1034
LEVER	49	7886	5578	10	5850	907
JET	52	36472	23303	23	31296	1492

Table 1: *Statistics of the base-complex optimization: We compare the number of helices # Hel, the number of quadrilaterals of the mesh # F and the number of quadrilaterals of the base-complex # BC between the input and the optimized mesh of several models.*

4. Results

For the evaluation of our base-complex optimization technique, we apply the method to several quadrilateral meshes generated with the method of [BZK09]. As a quantitative evaluation we compare the number of helices and the quality of the base-complex of the input mesh against the optimized mesh as shown in Table 1. The quality of the base-complex is measured by the number of its quadrilateral patches , i.e. the number of quadrilaterals that remain after removing all regular parametric lines. All results were computed within a few minutes on a standard PC.

For all meshes most of the q-helices could be removed leading to a significant reduction of the base-complex size. On the FANDISK model the optimization method reduces the size of the base-complex from 408 to 144 quadrilaterals. Furthermore we experimentally collapsed all face-loops that did not lead to singularity merges or collapsing features. In this experiment the base complex could be even reduced to 90 quadrilaterals, as shown in the right most picture of the FANDISK in Figure 11. However, this reduction comes at the cost of moving the valence five singularity onto the feature line on top of the FANDISK which is not optimal and induces unwanted stretch.

Another additional experiment was performed on the FERTILITY model, where the right most picture in Figure 11 shows the result of a base-complex optimization where the merging of singularities was allowed. Here the size of the base-complex could be reduced from 526 to 222 but again the overall distortion of the patches increased as a result of the merged singularities. Whether such aggressive reductions are useful depends strongly on the desired application.

The BOTIJO and the LEVER model both have a larger number of singularities leading to rather many separating lines despite the removal of most of the helices. But still the decoupling of quad-loops is advantageous for many applications enabling for example a better control of anisotropic edge-lengths.

Limitations. The presented algorithm works in a greedy fashion and therefore it is no surprise that we cannot guarantee optimality. Due to the iterative graph search it is even not guaranteed to find a suitable GP-operator if one exists.

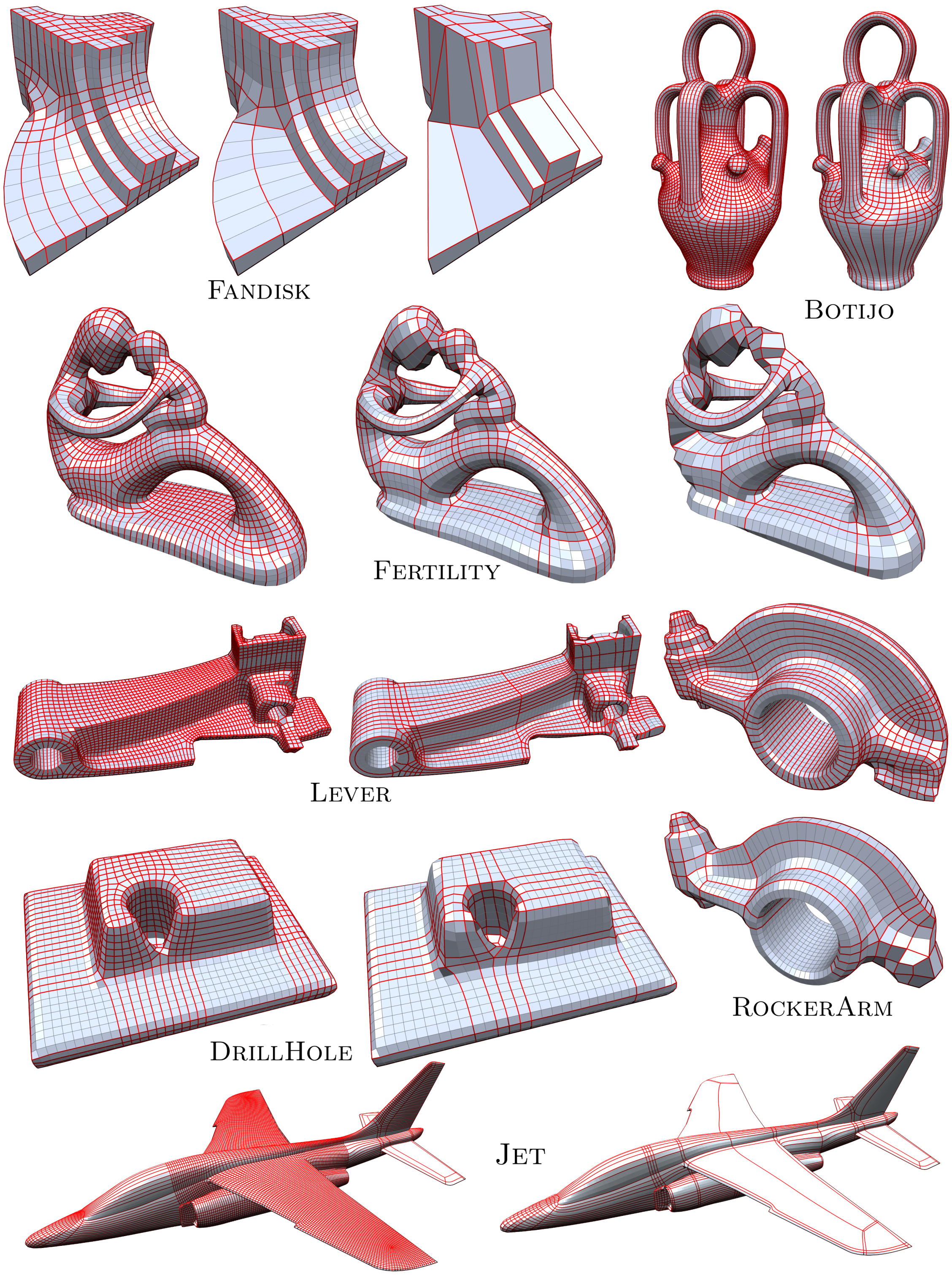


Figure 11: *Comparison of various example meshes before and after our base-complex optimization. The red lines indicate the base-complex, i.e. all parameter lines emanating from the singular vertices. For the* FANDISK *model the third result is a maximal reduction of quadloops without merging singularities, while the third picture of* FERTILITY *was created by allowing singularity merges within the helix removal step.*

Our experiments showed that prioritizing the q-helices by their winding-number usually leads to good results, but we also experienced counter examples where a different ordering performed better.

Furthermore the resulting base-complex is strongly dependent on the number and placement of singularities in the input, since we do not change them. In particular for unstructured quadrilateral meshes like the cylinder in Figure 1 (a) it cannot be expected to achieve a coarser base-complex without adequately adjusting the singularities.

While the topological optimization is completely robust and parameter free, the mesh smoothing may occasionally lead to geometric instabilities. Replacing the explicit smoothing by a superior parametrization based method which e.g. exploits the optimized base-complex could be an interesting research topic for the future.

5. Conclusion

We have presented a fully automatic method which optimizes the base-complex of a given quadrilateral mesh by greedily removing helical structures without destroying regular parts. Removing a single helix is done by applying the proposed GP-operators, which combine atomic operations in order to perform edge flips without introducing new singularities or non-quadrilateral elements. Finding such a GP-operator is equivalent to a graph search problem.

We believe that the concept of a GP-operator has much more potential than just removing helical structures and may stimulate further work in this area. Interesting extensions include different graph search algorithms and GP-operators for the alignment of singularities.

Acknowledgements

This project was funded by the DFG Cluster of Excellence on Ultra-high Speed Mobile Information and Communication (UMIC), German Research Foundation grant DFG EXC 89, http://www.umic.rwth-aachen.de, and the Aachen Institute for Advanced Study in Computational Engineering Science (AICES). Additionally we would like to thank EADS, *AIM@SHAPE* and Carlos Hernández http://www.tsi.enst.fr/3dmodels for providing us with datasets, Jan Möbius for the geometry processing framework http://www.OpenFlipper.org and the reviewers for their competent and helpful comments.

References

[ACSD*03] ALLIEZ P., COHEN-STEINER D., DEVILLERS O., LEVY B., DESBRUN M.: Anisotropic polygonal remeshing. *ACM Trans. Graph. 22*, 3 (2003), 485–493. 2

[AUGA05] ALLIEZ P., UCELLI G., GOTSMAN C., ATTENE M.: *Recent Advances in Remeshing of Surfaces*. Research report, AIM@SHAPE Network of Excellence, 2005. 2

[BVK08] BOMMES D., VOSSEMER T., KOBBELT L.: Quadrangular parameterization for reverse engineering. In *Mathematical Methods for Curves and Surfaces* (2008), pp. 55–69. 2

[BZK09] BOMMES D., ZIMMER H., KOBBELT L.: Mixed-integer quadrangulation. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), ACM, pp. 1–10. 2, 8

[DBG*06] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J. C.: Spectral surface quadrangulation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (2006), pp. 1057–1066. 2

[DSC09a] DANIELS II J., SILVA C. T., COHEN E.: Localized quadrilateral coarsening. In *SGP '09: Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), Eurographics Association, pp. 1437–1444. 2

[DSC09b] DANIELS II J., SILVA C. T., COHEN E.: Semi-regular quadrilateral-only remeshing from simplified base domains. In *SGP '09: Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), Eurographics Association, pp. 1427–1435. 2

[DSSC08] DANIELS J., SILVA C. T., SHEPHERD J., COHEN E.: Quadrilateral mesh simplification. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–9. 2, 3, 6, 8

[HZM*08] HUANG J., ZHANG M., MA J., LIU X., KOBBELT L., BAO H.: Spectral quadrangulation with orientation and alignment control. *ACM Trans. Graph. 27*, 5 (2008), 1–9. 2

[Kin97] KINNEY P.: Cleanup: Improving quadrilateral finite element meshes. In *6th International Meshing Roundtable* (1997), pp. 449–461. 2

[KNP07] KÄLBERER F., NIESER M., POLTHIER K.: Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum 26*, 3 (Sept. 2007), 375–384. 2

[MK04] MARINOV M., KOBBELT L.: Direct anisotropic quad-dominant remeshing. In *PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 207–216. 2

[MPKZ10] MYLES A., PIETRONI N., KOVACS D., ZORIN D.: Feature-aligned t-meshes. *ACM Trans. Graph. 29*, 4 (2010), 1–11. 2

[RLL*06] RAY N., LI W. C., LÉVY B., SHEFFER A., ALLIEZ P.: Periodic global parameterization. *ACM Trans. Graph. 25*, 4 (2006), 1460–1485. 2

[SDW*10] SHEPHERD J. F., DEWEY M. W., WOODBURY A. C., BENZLEY S. E., STATEN M. L., OWEN S. J.: Adaptive mesh coarsening for quadrilateral and hexahedral meshes. *Finite Elem. Anal. Des. 46*, 1-2 (2010), 17–32. 2

[TACSD06] TONG Y., ALLIEZ P., COHEN-STEINER D., DESBRUN M.: Designing quadrangulations with discrete harmonic forms. In *Proc. SGP* (2006), Eurographics Association, pp. 201–210. 2

[TPC*10] TARINI M., PIETRONI N., CIGNONI P., PANOZZO D., PUPPO E.: Practical quad mesh simplification. *Computer Graphics Forum (Special Issue of Eurographics 2010 Conference) 29*, 2 (2010), 407–418. 2

[ZBX05] ZHANG Y., BAJAJ R., XU G.: Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. In *In Proceedings, 14th International Meshing Roundtable* (2005), John Wiley & Sons, pp. 449–468. 8

[ZHLB10] ZHANG M., HUANG J., LIU X., BAO H.: A wave-based anisotropic quadrangulation method. *ACM Trans. Graph. 29*, 4 (2010), 1–8. 2