# Demystifying Quadrilateral Remeshing 

Daniele Panozzo<br>ETH Zurich

Surface manipulation and representation is becoming increasingly important, with applications ranging from special effects for films and videogames to digital fabrication and architectural design. Despite significant research efforts, there is still a large technological gap between the acquisition of 3D models and the tools used to process, edit, and render them. Acquired 3D objects do not possess a high-level structure; they often start as a "point cloud" of surface points from which a triangle mesh is derived (see Figure 1). Conversely, the majority of the 3D models used in movies, games, and CAD applications are smooth surfaces with a shape that is controlled by a coarse grid of quadrilaterals. Depending on the application, the quadrilaterals can be either subdivided or used as control points for defining high-order polynomial surfaces.

The conversion of a triangle mesh into a quadrilateral mesh is a difficult problem that has been deeply studied during the last decade. Many different approaches have been proposed and transferred from academia to commercial modeling and CAD software. However, there are still many open problems to solve to provide a fully automatic pipeline that converts an unstructured model into a highlevel representation that can be directly used in a conventional 3D modeling pipeline.

This article explains why it is important to represent surfaces using quadrilateral meshes, while providing an overview of the conversion techniques that I invented to retopologize triangular meshes. (For a complete literature overview, interested readers are referred to the quadrilateral meshing survey written by Bommes and his colleagues. ${ }^{1}$ ) After showing how these algorithms can be applied to contemporary problems in architectural geometry, their limitations and interesting future research directions are discussed.

## Quadrilateral Remeshing

The creation of triangle meshes has been extensively studied in the last 30 years, and many robust algorithms are currently available in productionquality software libraries. A triangle mesh is the preferred choice for real-time rendering because it is well-suited for the classic rasterization-based rendering pipeline used by modern GPUs. Its main limitation is that it does not contain any global structure, and it is thus difficult to edit or use as a control grid for higher-order surfaces. This is why quadrilateral meshes are ubiquitous in the movie industry and in CAD applications: they naturally represent a pair of directions, and they exhibit global structures called edge loops (see Figure 2), which are chains of edges that can be selected by


Figure 1. 3D object acquisition. (a) A triangle mesh, (b) a coarse quadrilateral mesh representing the same object, and (c) its Catmull-Clark subdivision limit surface.


Figure 2. Edge loops allow quick and precise selection of semantically meaningful parts of a 3D model. A single click on any part of a loop will select the entire chain of edges. Such loops might be closed or open.
picking any edge in the loop, increasing the efficiency of modelers, riggers, and animators.

The de facto approach to model smooth surfaces in CG movies is the Catmull-Clark subdivision, which is a recursive algorithm that produces a smooth surface with a shape that is completely determined by a coarse control grid. Finally, quadrilateral meshes are often preferred over triangular meshes for discretizing partial differential equations (PDEs) because of their numerical properties.

## Mesh Quality

The quality of a triangular mesh is usually determined by the shape and size of the triangles; equilateral and uniformly sized triangles are often preferred for rendering smooth surfaces, where they provide an even interpolation of the per-vertex normals. High-quality triangular meshes also exhibit better numerical properties when used for solving partial differential equations.

Although similar quality metrics could be defined for quadrilateral meshes, they are not useful quality measures because quadrilateral meshes are not directly used to represent 3D shapes. A quadrilateral mesh is usually used as a control grid for a subdivision or NURBS surface (Figure 1). In both cases, the quality of the final surface is mainly affected by the distribution of singularities that are the vertices touched by more or less than four edges. These vertices are particularly important because they are the only ones where the quadrilateral mesh is not a regular grid. The limit surface of a Catmull-Clark subdivision is only $C^{1}$ smooth on singular points, and it is challenging to smoothly connect more than four NURBS that meet at a single point.

A singular vertex can have a different singularity index, depending on the number of edges touching
it, and the total number and index of the singularities is a topological invariant that depends on the genus of the surface. ${ }^{2}$

The quality of a quadrilateral mesh is thus measured by the following:

- the number of singularities used (the lower the better);
- the geometric location of the singularities (the generated artifacts in the smooth surface should be hidden or not disturbing); and
- the alignment of the edges, aka edge flow (edges should be aligned with meaningful features, thus allowing artists to quickly select semantically meaningful edge loops).

The last two requirements are difficult to evaluate objectively and are difficult to incorporate in an optimization algorithm. Many quadrilateral meshing algorithms incorporate user input to allow users to adjust the result depending on their preferences.

## Computer-Assisted Quadrilateral Remeshing

Traditional modeling tools allow us to retopologize a surface by manually placing its vertices and edges. This procedure is time consuming and prone to errors; because the number and placement of singularitites is a global feature of the mesh, it cannot be modified using local operations such as insertion of vertices and edges. Thus, if a mistake is made while retopologizing a part of a shape, it is not possible to fix it locally, and the entire model has to be retopologized from scratch. Surprisingly, this procedure is still the main method used by artists and CAD engineers to design coarse quadrilateral control grids, and computer-assisted quadrilateral remeshing methods are only recently starting to be used instead.


Figure 3. Local operations used to coarsen a dense quadrilateral mesh.


Figure 4. Local method example. (a) An input triangular mesh is converted into (b) a coarse quadrilateral grid with a series of local operations. The high-resolution details of the original surface are stored as a displacement map over each quadrilateral, and (c) they can be efficiently rendered in an OpenGL-based rendering system.

There are three major families of quadrilateral meshing algorithms:

- Local methods adapt simplification strategies developed for triangular meshes to quads. Although they tend to introduce a large number of singularities, their efficiency, simplicity, and stability make them ideal for processing large and noisy datasets.
- Global methods use global optimization strategies to optimize directly the number and placement of the singularities. They generate high-quality results, at the expense of efficiency and controllability.
- Interactive methods integrate local optimization algorithms with smart user interfaces, providing semiautomatic tools that are both efficient and highly controllable.

Local methods. I proposed two local methods to convert a triangle mesh into a coarse quadrilateral mesh. ${ }^{3,4}$ They first trivially pair every two triangles in a quadrilateral, thus producing a dense quad-
rilateral mesh, and then reduce the number of quads using simple and efficient local operations. Each local operation affects a small part of the quadrangulation, reducing the number of quads by collapsing one of them or improving the shape of the quads by rotating an edge (see Figure 3).

Although this technique does not provide explicit control over the placement and number of introduced singularities, it was the first completely automatic method able to create coarse quadrilateral meshes.

The algorithm is simple, robust, and as opposed to some of the methods presented in the next sections, it can be applied to dense meshes. The 3D positions of the quadrilateral layout can be optimized so that the induced smooth surface (obtained with Catmull-Clark subdivision) is as close as possible to the original geometry, whose details can be stored as a displacement map (see Figure 4). Note that the created mesh is adaptive-that is, it concentrates the elements in the regions with more geometric details. This feature can be disabled for applications that benefit from uniform elements.


Figure 5. Five major steps of a global, parametrization-based, anisotropic meshing algorithm, from left to right. (a) Constraints (green quads) are provided by an artist, and (b) they are converted into a frame field, (c) which is deformed together with the shape until it becomes a cross field. (d) The deformed mesh is quadrangulated using a global parametrization approach and (e) then warped back to its original shape.

Global methods. Global methods rely on a global optimization to decide the number and position of the singularities. In this section, I focus on my algorithm that generalizes previous work by allowing adaptive and anisotropic quadrilateral meshes. ${ }^{5}$ Figure 5 illustrates the algorithm's five major steps:

1. Edge alignment constraints are manually defined by a user or extracted from principal curvature directions. The user-specified constraints are a sparse set of quadrilaterals, with size and edge alignment that will be preserved in the final quadrilateral mesh and interpolated in the remaining regions.
2. The constraints are interpolated over the surface, defining a set of directions over the entire surface. Different algorithms can be used for this task, each providing different control over the type of constraints that the user can specify. ${ }^{5,6}$
3. The shape is deformed depending on the interpolated constraints, enlarging regions where a higher density of quadrilateral has been prescribed. ${ }^{5}$
4. The deformed directions are used as gradients for a global parametrization, whose principal axes are aligned with the specified directions in the least-square sense. The integer isolines of the parametrization naturally define a uniform quadrilateral mesh on the surface. David Bommes and his colleagues introduced this method in 2009, ${ }^{7}$ and many variants and extensions of this algorithm have been proposed in the last four years. ${ }^{8}$
5. The deformation (step 2) is reverted, revealing the final quadrangulation that follows the constraints specified by the user in the first step.

The major advantage of global methods is that they provide precise control over the singularities of the resulting quadrilateral mesh that can be
either prescribed manually or automatically optimized (see Figure 6). Their main disadvantages are the extreme implementation complexity and that they might fail to generate a pure quadrilateral mesh. Depending on the constraints, these methods might introduce triangles or pentagons close to singularities.

Step 2 can be further enhanced by extracting symmetries from the original object and enforcing them during the interpolation of the alignment constraints to create a symmetric quadrilateral mesh. ${ }^{9}$ This is particularly useful for man-made CAD objects that often exhibit multiple bilateral symmetries (see Figure 7).

Interactive Methods. Exact control of every edge of the quadrangulation is required in certain applications, such as the creation of extremely low-poly videogame characters or the creation of high-end movie characters. In these cases, fully automatic algorithms are not applicable because they do not provide sufficient control. The current solution is to manually draw every quad, a task that can consume multiple days of manual labor on complex models.

To tackle this problem, I introduced a new interaction paradigm ${ }^{10}$ to efficiently quadrangulate a surface using a sketch-based interface, while having accurate control over each vertex and edge wherever necessary. In contrast to the parametrization-based approaches, the users draw a subset of the edges they want to have in the final mesh, and the system takes care of filling the regions between the strokes with quadrilateral elements.

The basic mode allows us to freely draw sketches on a surface: when a loop is formed, the region enclosed by the loop is tessellated (see Figure 8). For every region, it is possible to change the number of edges on each of its sides and change the connectivity in its interior.

Figure 6
Global method example. An artist manually drew the quadrilateral mesh on hands and feet plus a sparse set of constraints (green quads). The algorithm produces a complete quadrilateral mesh that preserves the singularities specified by the artist and adds new ones to ensure a smooth interpolation of the constraints. The five images correspond to the five steps in Figure 5. Red and blue bullets depict singularities.


To further improve the efficiency, the system provides a set of tools for the most common cases. A brush tool can be used to quickly create a strip of quadrilaterals, the autocompletion tool automatically fills the region below the cursor, and the cylinder tool creates a circular pattern around cylindrical regions such as fingers and limbs.

## Impact and Applications

Automatic and assisted remeshing algorithms are now available in commercial modeling softwares such as ZBrush and 3DCoat. They are daily used by hundreds of artists and engineers to create coarse and dense quadrilateral grids for high-end videogames, movie visual effects, and engineering applications. Quadrilateral meshing algorithms are also deeply affecting architectural designs,
enabling architects to design and build free-form structures that were impossible to realize with traditional techniques.

## Planar Panelization

Glass and steel constructions are common because of their combination of functionality and specific aesthetic. However, their construction is expensive if curved glass panels are used. Tessellating a surface with flat quadrilaterals is a challenging problem that requires expensive numerical optimization.
With slight variations to the algorithms presented earlier, it is possible to tessellate a surface with planar quadrilaterals. Starting from a planar tessellation, a building can be constructed by replacing each face with a flat glass panel that is much less expensive to manufacture. Mathemati-
cally, the edges of a quadrilateral mesh with flat faces define a conjugate field, ${ }^{11}$ which is challenging to design. I recently proposed a simple and efficient algorithm ${ }^{6}$ that allows architects to quickly experiment with different planar tessellations by simply specifying a set of desired alignment constraints (see Figure 9).

## Free-Form Masonry Structures

Another way in which quadrangulation could be used is the design and tessellation of free-form masonry structures. These structures consist of unsupported stone blocks and stand thanks to their special geometry where all blocks are in static equilibrium. Masonry structures have been widely used in the past by following simple patterns for the construction of arcs and vaults, and only recently they are being used in modern architecture to realize free-form designs.

I proposed a completely automatic pipeline that converts a sketch of a surface into a masonry model consisting of hexagonal blocks. ${ }^{12,13}$ An automatic quadrilateral remeshing algorithm creates a coarse force layout that is used to transform the given sketch into a self-supporting surface (see Figure 10). The same algorithm is then used to split the surface into a staggered grid of blocks. The block pattern is a quadrilateral mesh, where specific edges are removed to create a staggering effect that increases the interlocking between the pieces, simplifying the construction and improving the structural properties of the masonry building. We validated our approach by fabricating and assembling small-scale models of masonry buildings.

## Open Problems

Quadrilateral meshing is still an open problem with many challenges left to address. The plethora of new applications that use it are adding more constraints to a problem that is already computationally challenging.


Figure 7. Symmetric quadrilateral meshes of a collection of objects with several bilateral symmetries. The blue and red spheres represent singular vertices, which are themselves symmetric.

## Code Complexity

The majority of the quadrangulation methods are complex to code, and they must be maintained and supported by highly trained programmers. This is a major problem that is both slowing down the research in this area (because only a few research groups have access to the state-of-the-art quadrangulation algorithms) and drastically increasing the cost for companies to embrace these new technologies.

To ameliorate this problem, I recently released a full implementation of the global method I presented earlier, ${ }^{5}$ which is available in libigl, a simple C++ geometry processing library (https://github .com/libigl/libigl). The source code and binaries of the interactive retopology system SketchRetopo ${ }^{10}$ are also available online to foster future research in this area. An important venue for future work in this area is the design of alternative methods that are simple to implement and maintain, while having a quality comparable with the current approaches.


Figure 8. Quadrilateral meshes created with the SketchRetopo system. The yellow curves have been manually designed by an artist, while the black quads are automatically created by the system. The yellow lines can be interactively adjusted, while seeing the effect on the final quadrilateral mesh in real time.


Figure 9. Planar quadrilateral meshes are used in architectural geometry to design free-form glass and steel structures. (a) The designer specifies a set of alignment constraints, and (b) the constraints are interpolated in a conjugate direction field (c) that is automatically converted into a mesh with planar faces.


Figure 10. An input surface is automatically transformed into a masonry 3D model. The equilibrium of the surface is represented by two planar graphs that encode the directions and magnitudes of all forces. The generated blocks are 3D printed and assembled into a physical model of the surface that stands in compression without using glue or reinforcements.

## Manual versus Automatic Quadrangulation

The ideal quadrangulation tool would allow a user to automatically tessellate a surface and then refine it by adding additional constraints that can be incorporated in the quadrangulation in real time. Unfortunately, this is not possible with any existing method. Completely automatic methods do not support exact constraints, and they are too slow to provide interactive feedback, while interactive methods cannot automatically quadrangulate an entire surface without user input. Combining the two approaches is far from trivial and would probably require a completely novel solution strategy, but it would provide the ideal solution for character retopology and CAD applications.

## Volumetric Tessellation

The advantages of quadrilateral meshes over triangle meshes similarly extend to the volumetric case, where hexaedral meshes (such as sets of cubes) are preferred over tetrahedral meshes. This problem shares many similarities with quadrangulation, while being considerably more challenging. The singularities can be both vertices or segments inside the volume. This is an exciting topic, with many problems to solve and only a few
problem-specific algorithmic solutions that cannot be applied to arbitrary shapes.

## Quadrangulation as a Design Tool for Architectural Patterns

Many architectural applications require the design of semiregular patterns on surfaces, such as the design of flat panelizations, force-flow-aligned beam structures, and brick patterns. Only recently, researchers started to adapt existing quadrangulation methods to these problems, and I expect that a plethora of interesting problems and designs will be proposed as architects increasingly discover the potential of these novel design tools.

The advent of 3D printing and digital fabrication is increasing the demand for tools to quickly and efficiently acquire and edit geometric data. Although 3D modeling requires a considerable amount of skill and artistic education, the 3D acquisition and editing of models is a much simpler task accessible to the masses. The tools presented in this article are striving to streamline the geometry processing pipeline, with the goal of fully automating the conversion of a point cloud
into a high-quality coarse control grid that can be used to edit and digitally fabricate variations of the scanned model. Many of the basic building blocks required to make this idea a reality are now available, and I look forward to the research challenges that lie ahead.

## References

1. D. Bommes et al., "State of the Art in Quad Meshing," Computer Graphics Forum (Proc. Eurographics STARS), vol. 31, no. 2, 2012.
2. N. Ray et al., "N-Symmetry Direction Field Design," ACM Trans. Graphics (Proc. ACM Siggraph), vol. 27, no. 2, 2008, article no. 10.
3. M. Tarini et al., "Practical Quadrilateral Mesh Simplification," Computer Graphics Forum (Eurographics), vol. 29, no. 2, 2010, pp. 407-418.
4. D. Panozzo et al., "Automatic Construction of Quad-Based Subdivision Surfaces using Fitmaps," IEEE Trans. Visualization and Computer Graphics, vol. 17, no. 10, 2011, pp. 1510-1520.
5. D. Panozzo et al., "Frame Fields: Anisotropic and NonOrthogonal Cross Fields," ACM Trans. Graphics (Proc. ACM Siggraph), vol. 33, no. 4, 2014, article no. 134.
6. O. Diamanti et al., "Designing $N$-PolyVector Fields with Complex Polynomials," Computer Graphics Forum, vol. 33, no. 5, 2014, pp. 1-11.
7. D. Bommes, H. Zimmer, and L. Kobbelt, "MixedInteger Quadrangulation," ACM Trans. Graphics (Proc. ACM Siggraph), vol. 28, no. 3, 2009, article no. 77.
8. H.-C. Ebke et al., "QEx: Robust Quadrilateral Mesh Extraction," ACM Trans. Graphics (Proc. ACM Siggraph Asia), vol. 33, no. 6, 2013, article no. 168.
9. D. Panozzo et al., "Fields on Symmetric Surfaces," ACM Trans. Graphics (Proc. ACM Siggraph), vol. 31, no. 4, 2012, article no. 111.
10. K. Takayama et al., "Sketch-Based Generation and Editing of Quad Meshes," ACM Trans. Graphics (Proc. ACM Siggraph), vol. 32, no. 4, 2013, article no. 97.
11. Y. Liu et al., "General Planar Quadrilateral Mesh Design Using Conjugate Direction Field," ACM Trans. Graphics (Proc. ACM Siggraph Asia), vol. 30, no. 6, 2011, article no. 140.
12. D. Panozzo, P. Block, and O. Sorkine-Hornung, "Designing Unreinforced Masonry Models," ACM Trans. Graphics (Proc. ACM Siggraph), vol. 32, no. 4, 2013, article no. 91.
13. M. Deuss et al., "Assembling Self-Supporting Structures," ACM Trans. Graphics (Proc. ACM Siggraph Asia), vol. 33, no. 6, 2014, article no. 214.

Daniele Panozzo is a senior researcher at ETH Zurich. Contact him at panozzo@inf.ethz.ch.

Contact department editor Jim Foley at foley@cc.gatech.edu.

## ADVERTISER INFORMATION

## Advertising Personnel

Marian Anderson: Sr. Advertising Coordinator
Email: manderson@computer.org
Phone: +1 7148162139 |Fax: +1 7148214010
Sandy Brown: Sr. Business Development Mgr.
Email sbrown@computer.org
Phone: +1 7148162144 | Fax: +1 7148214010

## Advertising Sales Representatives (display)

Central, Northwest, Far East:
Eric Kincaid
Email: e.kincaid@computer.org
Phone: +1 2146733742
Fax: +1 8888868599
Northeast, Midwest, Europe, Middle East:
Ann \& David Schissler
Email: a.schissler@computer.org, d.schissler@computer.org
Phone: +1 5083944026
Fax: +1 5083941707

Southwest, California:
Mike Hughes
Email: mikehughes@computer.org
Phone: +1 8055296790
Southeast:
Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1 9733044123
Fax: +1 9735857071
Advertising Sales Representatives (Classified Line)
Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1973 3044123
Fax: +1 9735857071

Advertising Sales Representatives (Jobs Board)
Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1 9733044123
Fax: +1 9735857071

