Tiling the Bunny

Quad Layouts for Efficient 3D Geometry Representation

Marcel Campen
New York University

Information to be processed by computers ultimately needs to be represented as a plain sequence of numbers. For many types of data a suitable encoding is not too hard to find. An area where this poses a major challenge, though, is the representation of shapes, in particular complex three-dimensional shapes. While primitive shapes, such as cubes or spheres are easily handled, for general *freeform* shapes a proper approach (in particular an efficient one) is far from obvious.

This became a major concern in industries where dealing with non-primitive geometric shapes is at the heart of the trade, the automobile and aircraft industries being prime examples. Extensive work conducted by pioneers such as de Casteljau, Bézier, de Boor, Coons, Ferguson, and Sabin at Citroën, Renault, General Motors, Boeing, the British Aircraft Corporation, and Ford in the 1950s-70s culminated in the standardization of the NURBS representation. It soon became "the standard curve and surface form in the CAD/CAM industry"[A].

The mathematical beauty of this concept, responsible for many practically important properties, comes with an unfavorable aspect once termed "the rectangular tyranny of NURBS" by M. Sabin: the tensor-product form (used to generalize from curves to surfaces) requires each surface piece represented to be four-sided in nature. While this restriction can be avoided or circumvented using other techniques, these either gained less traction ("tensor product surfaces are far more often encountered"[A]), e.g. for reasons of interoperability, or complicate tasks such as smoothly joining multiple surface pieces to a complex surface.

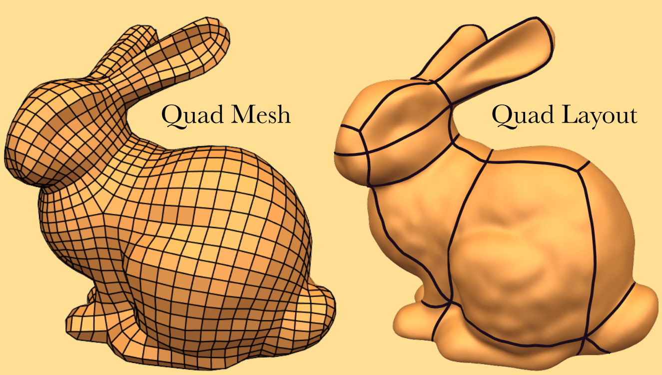
One thus faces the fundamental task of dividing a given surface (whether in form of a mental picture, a real-world workpiece, or a virtual design represented in some different form) into four-sided pieces. These

Quad Meshes vs Quad Layouts

A large variety of methods for the generation of finite element meshes with quadrilateral faces (*quad meshes*) for given surfaces is available. Structurally, these quad meshes are no different than quad layouts (a network of conforming four-sided elements). Why do we need novel methods for quad layout generation?

Meshes and layouts follow different geometric objectives. For instance may the vertices of a quad mesh be expected to sufficiently encode the full surface shape, or to sample the surface at a desired resolution. Quad layouts rather provide a, possibly much coarser and less uniform, partition, a domain, on top of which additional information or structure can be encoded, e.g. polynomial/rational parametric surface patches (NURBS) or regular/adaptive grids for highly structured discretization of the surface.

This conceptual difference necessitates the use of different construction strategies for both classes: while the quality of a quad mesh can be assessed quite locally, a different perspective is necessary to capture global structural and topological aspects for layouts. We note that it is easy to refine a layout to a mesh but in general very hard to coarsen a mesh to a layout of high quality.



can then appropriately be represented in NURBS form.

A problem of the same kind is of central importance in the related area of Finite Element Analysis: for certain scenarios, the use of four-sided elements to discretize a surface proves to exhibit significantly favorable properties in terms of simulation accuracy and convergence behavior in comparison to commonly used simplexes, i.e. triangular elements.

My thesis[B] asks the question how these problems of finding a good *quad layout*, i.e. a partitioning of surfaces into four-sided pieces, can be tackled efficiently, in automatic or semi-automatic ways, as opposed to manual workflows abounding in tedium, as commonly found in pertinent industries to the present day. In this article the findings and the novel algorithmic ideas are summarized. We show how they can be combined to form fully automatic as well as semi-automatic, interactive quad layout generation pipelines. The latter allows for the injection of high-level expert knowledge in complex scenarios where a fully automatic, high-quality solution is yet elusive.

Quad Layouts

The partitioning of geometrically (and possibly topologically) complex surfaces into simpler surface patches is a key principle in a variety of applications involving 3D geometric data. It can, for instance, reduce the complexity of representation, processing, or analysis tasks. In some cases, unstructured arrangements of arbitrarily shaped patches can be sufficient, e.g. to define a texture atlas, a UV map. Many applications, however, require structurally more constrained partitions, or *patch layouts,* so as to be able to properly coordinate the results obtained on individual patches to form a global solution for the entire surface.

A layout type that has a particularly high level of structuredness and at the same time experiences a particularly high level of demand from the application side, is the *conforming quad layout.* In this type of patch layout, each patch is four-sided (*quad*), and adjacent patches share sides never partially but always entirely (*conforming*). Fig. 1 illustrates the difference of conforming and non-conforming quad layouts.

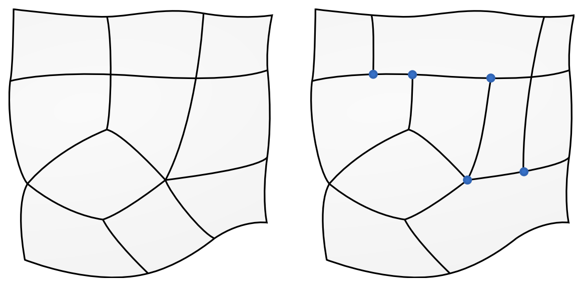


Fig. 1 Left: conforming layout with quad patches. Right: non-conforming quad layout; adjacent quads can share sides partially, causing T-junctions (blue).

Conforming quad layouts, for instance, enable the use of tensor-product surface representations based on NURBS or Bézier patches, of grid-based multi-resolution techniques for the solution of differential equations on surfaces, and of high-quality discrete pixel-based representations of maps from, onto, or between surfaces. Fig. 2 shows an exemplary conforming quad layout. Fig. 3 illustrates the use for map representation.

The Hardness of Quad Layout Generation

As one might expect, finding or constructing a patch layout (of adequate quality) is typically the harder the higher the structural demands. In the case of conforming quad layouts, the requirement of four-sidedness in combination with the requirement of conformity causes structural interdependencies on

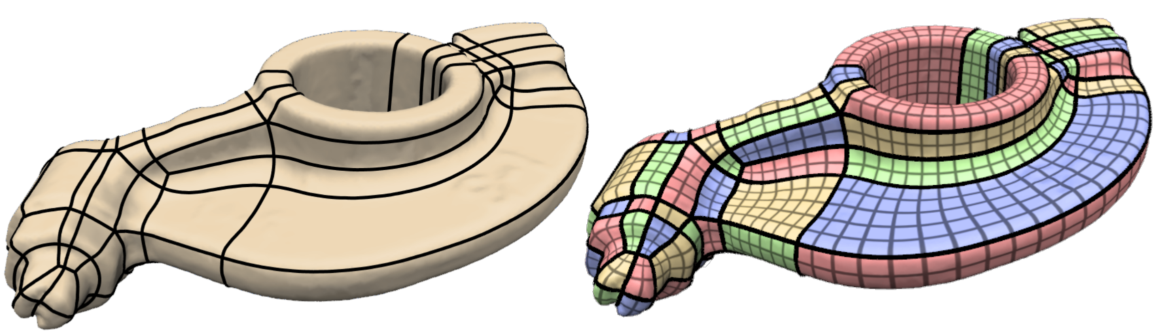


Fig. 2 Left: exemplary conforming quad layout on a surface. Right: a regular grid mapped into each quad, illustrating the use of the quad layout for semi-regular quadrilateral mesh generation.

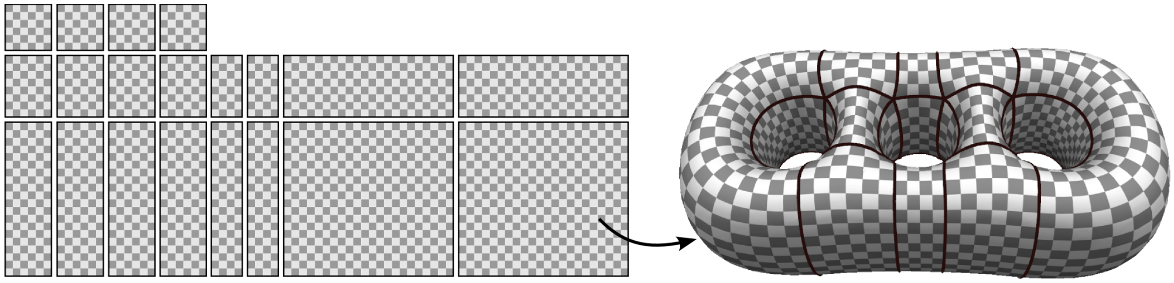


Fig. 3 Using a quad layout to define a piecewise map from a set of simple planar rectangles onto a surface of non-trivial topology. The specific properties of a conforming quad layout lead to simple transitions between the pieces and beneficial continuity properties across patch borders.

a global level, making this case particularly challenging. In the end, the difficulty stems from two main facts:

- **Mixed nature:** the conceptual optimization problem for finding a high quality quad layout is of mixed nature: it has *continuous*, *discrete*, and *combinatorial* or *topological* degrees of freedom.
- **Quality notion:** the notion of *quality* of a quad layout is often complex, application-dependent, sometimes fuzzy, thus hard to formalize.

The mixed types of degrees of freedom (or variables) of the problem are due to its very nature. The number of patches, number of joints where multiple patches meet, the number of patches meeting at a particular joint—these are discrete degrees of freedom. The position of these joints on the surface, the shape of the patches and their borders—these degrees of freedom are continuous. Finally, combinatorial and topological aspects come into play when one asks which route patch borders take over a topologically non-trivial surface and which of the joints they connect to.

The quality of a quad layout is a measure that to some extent depends on the intended application. We can identify some generic aspects which are common to most application scenarios:

- **Geometric fidelity:** patches should be mappable to rectangles with low parametric distortion (intuitively, in Fig. 3 the squares of the planar checkerboard pattern should not be far from being square when mapped to the surface).
- **Structural simplicity:** the number of patches should be small.

Geometric fidelity in terms of low parametric distortion is a key factor in many applications, while structural simplicity gives preference to quad layouts that lead to simpler surface representations, simpler mapping domains, or more flexibility for hierarchical structures (e.g. in the context of generating highly regular meshes for multigrid solvers).

Unfortunately, both aspects tend to be opposing objectives; low distortion can be achieved at the expense of a large number of small patches, while a small number of patches typically requires less nicely shaped patches, i.e. higher parametric distortion. Hence, a good quad layout generally is a compromise, balancing layout simplicity and patch rectangularity, and possibly further application-dependent objectives.

This hardness of the problem of constructing a good quad layout causes manual layouting approaches, which are common in many practical workflows in animation, engineering, and simulation, to often be very time-consuming and tedious, even for experienced experts. This was a major source of motivation for the investigation of methods and techniques that can further the automation of this process.

The core novelty of the proposed methods for quad layout generation is the use of *dual* approaches for the most challenging central part, the settling of the combinatorial/topological degrees of freedom: instead of explicitly constructing the graph of patch borders, we effectively construct its dual graph. A key aspect of this dual perspective is that it enables a robust *incremental* construction of quad layouts: one can ensure that a valid layout can always be reached from arbitrary intermediate states; there are no dead-ends that would require expensive backtracking. This proved to be of significant benefit for efficiency (and thus, in practice, result quality) in automatic as well as interactive layout construction scenarios, as further detailed in the following.

Automatic Layouting

Our key concept to make the complex, mixed-nature problem of quad layout generation tractable, is the division into a set of sub-problems, each of which is of a simpler, more homogeneous nature. Using appropriate strategies, these sub-problems can

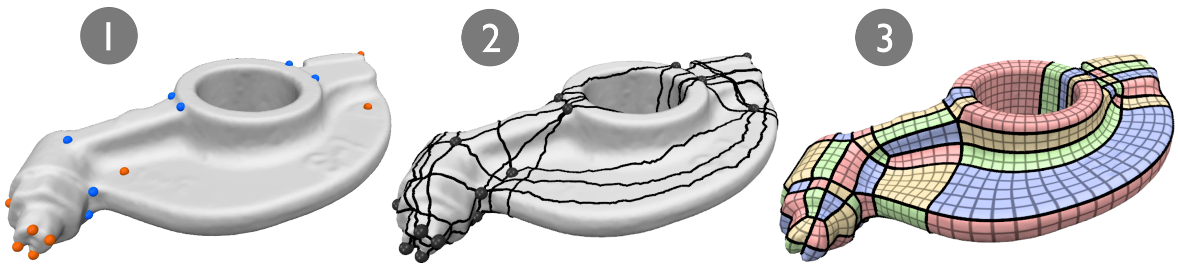


Fig. 4 We construct quad layouts by solving three sub-problems in sequence. In stage 1, the *nodes* (or joints, or vertices) of the layout graph are determined (red and blue dots), fixing the discrete degrees of freedom of the problem. In stage 2, the connectivity in form of the layout's *arcs* (or edges) is generated, fixing the combinatorial and topological degrees of freedom. Note that arc intersections can form additional (regular, degree 4) nodes. In stage 3 the continuous degrees of freedom are optimized, yielding a high-quality *embedding* of the layout's nodes, arcs, and patches in the surface.

be solved in sequence, leading to appropriate solutions of the original problem.

In Fig. 4 these sub-problems, which focus on the discrete, combinatorial, and continuous degrees of freedom, respectively, are illustrated.

The Discrete Degrees of Freedom

In a first stage, we should settle the discrete degrees of freedom, i.e. decide on a node configuration for the layout. This includes the number of nodes and possibly their intended degrees (the numbers of incident patches; also called valence). Before this question is settled, we cannot reasonably argue about anything else, such as layout connectivity or four-sidedness of patches.

It is most important to consider *extraordinary* nodes, i.e. nodes that are supposed to be part of a number of patches that differs from four (the *regular* case) in this context. Regular nodes will emerge automatically where necessary or beneficial at a later stage of our pipeline.

We argue[B] that it is particularly beneficial for layout quality if the set of nodes is derived from a (in a certain sense) least effort redistribution of Gaussian curvature on the surface. Computationally this can be handled by synthesizing global vector fields on the surface, optimized for smoothness (low variation), and deriving nodes and their degrees from the fields' singular points. Fig. 5 shows the node sets determined in this manner on exemplary surfaces.

An interesting property of this approach is that it easily allows for taking the surface's principal directions of curvature (illustrated in Fig. 6 left) into account already in this first stage. This will make it easier in the following stages to achieve that the layout's arcs are aligned with these curvature directions. This alignment is an important component of layout quality in a variety of scenarios; it is beneficial for the precision or approximation quality of surface representations based on the resulting quad layouts, it can reduce aliasing artifacts, and it can improve the planarity of elements in derived quad meshes.

The Combinatorial Degrees of Freedom

Once a desired configuration of nodes has been determined, suitable arcs connecting these nodes need to be constructed. Aiming for a quad layout, we require these arcs to partition the surface exclusively into four-sided patches. The fact that the set of possible arcs to choose from is infinite (even if we

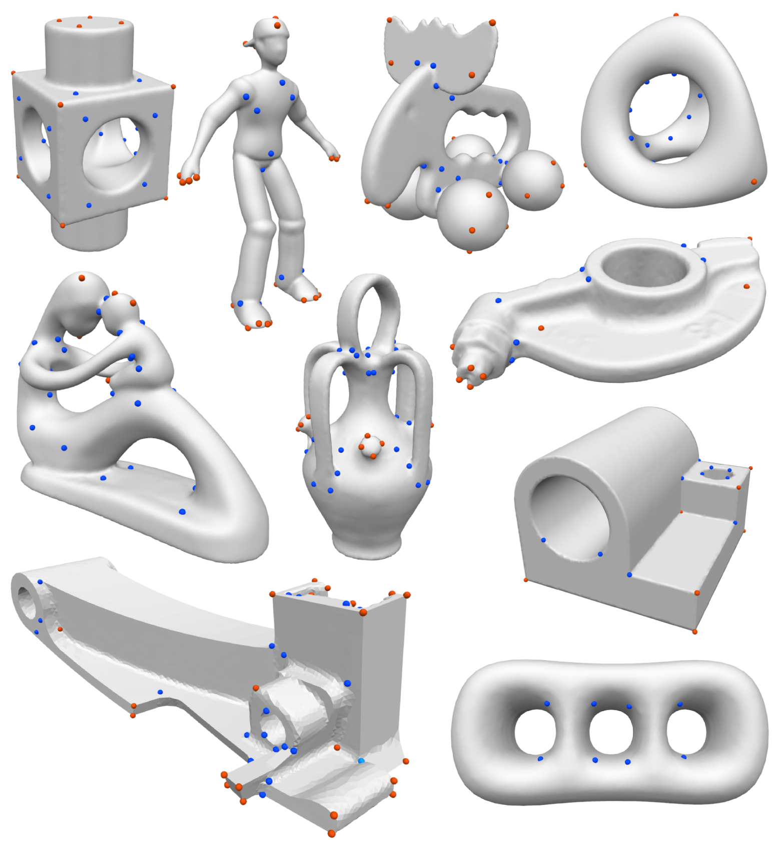


Fig. 5 Extraordinary layout nodes determined on exemplary surfaces in a globally shape-aware manner based on Gaussian curvature. Colors indicate intended node degree (red=3, blue=5, cyan=6).

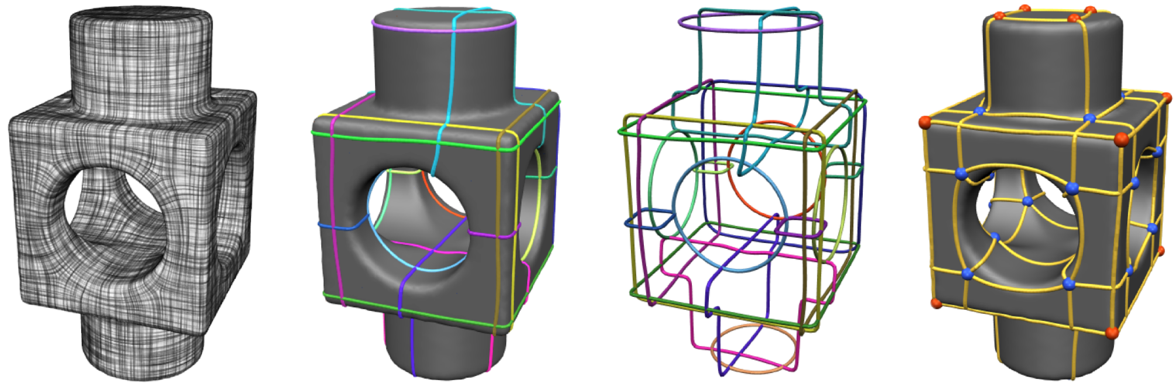


Fig. 6 Essence of the Dual Loops Meshing approach. Based on a cross field of principal directions (left), closed loops are incrementally computed in the form of anisotropic geodesic curves (center left and center right). The dual of the graph formed by these intersecting loops is a pure quad layout by design (right).

disregard their precise geometrical shape and focus only on topology and combinatorics) indicates that this is not an easy problem.

One might picture simple greedy strategies to incrementally add arcs in between nodes, based on some notion of node proximity. Such strategies, based on local considerations and local operations, however, have limited chances of success; without a global view of the situation it is hard to insure arriving at a final state where every patch formed by the arcs is actually four-sided, i.e. a quad.

Creating layouts with arbitrary polygonal patches, or triangular patches, is easily facilitated by local incremental approaches. We have thus seen a number of techniques proposed for the conversion of polygonal or triangular layouts into quad layouts, e.g. by splitting polygons into quads or by merging triangles into quads. Control over the quad layout's structure is limited in this context and yielding a layout of high quality remains an issue.

The key observation in the context of our Dual Loops Meshing method[C] is the fact that an incremental construction of quad layouts is well feasible if we operate in a *dual* setting, in the sense of graph duality. Instead of primal arcs, we conceptually construct dual arcs. Due to the four-sidedness of patches these dual arcs generally form closed loops on the surface. We show that *any* arrangement of loops on a surface that intersect simply and partition the surface into simply-connected regions, does actually form the dual graph of a quad layout. We can thus proceed by incrementally adding loops onto the surface until all regions are split into simply-connected pieces while avoiding non-simple loop intersections. The dual graph of the graph formed by these intersecting loops then is guaranteed to be a quad layout.

The central challenge in this context is of course the choice of dual loops such that they imply a layout that is not only structurally sound, but also of high quality geometrically. We describe an algorithm that constructs dual loops as closed *anisotropic geodesic*[U] curves (shortest paths on a surface, where 'shortest' is in terms of a metric that locally varies depending on a path's direction) on the surface, guided by a field of principal curvature directions (as was also used for the node determination). This is illustrated in Fig. 6. Intuitively, it is due to the shortness of geodesic

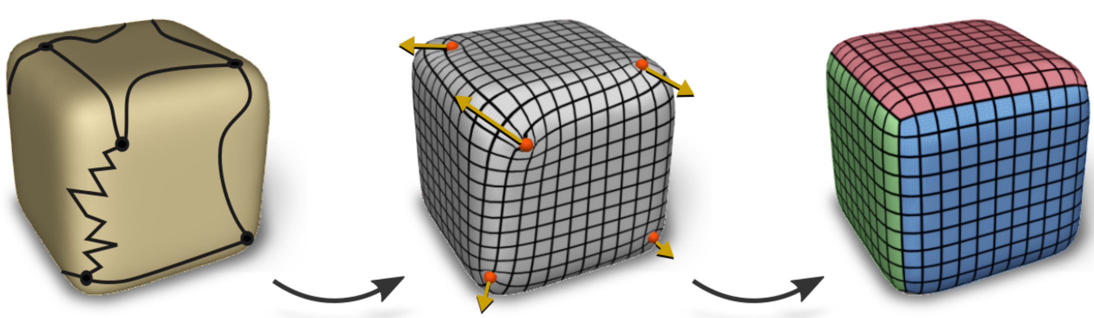


Fig. 7 Our quad layout embedding optimization takes as input a topological description of a quad layout in preliminary shape (left; here a particularly bad, artificial example for demonstration). A global mapping onto an abstract domain is established (middle; visualized by a grid texture) and optimized by a combination of direct, linear optimization techniques with non-linear gradient descent techniques to obtain a high-quality quad layout embedding (right).

curves that the resulting quad layout is simple, and due to the field guidance that it has geometric fidelity. We demonstrate that anisotropic geodesic loops can efficiently be computed in a discrete setting using a dynamic programming approach based on Dijkstra's algorithm.

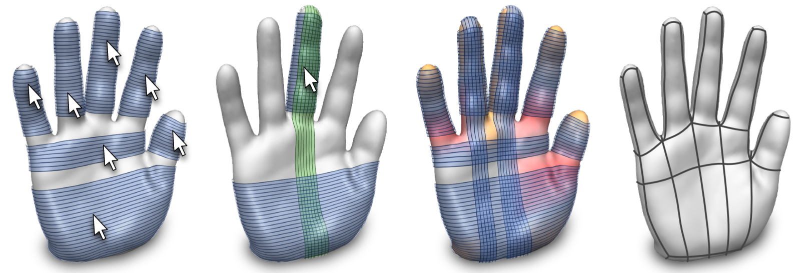


Fig. 10 Interaction modes for quad layout design using Dual Strip Weaving. Elastica strips (visualized in blue or green) are automatically computed and proposed to the user when hovering over the surface. With a single click these strips can be selected, or modified as desired. A heat map (middle right) points the user to regions where more or other strips would benefit layout quality. The primal quad layout (right) is automatically derived from the set of dual strips.

The Continuous Degrees of Freedom

The above described algorithms construct a graph (of nodes and arcs) which is embedded in the surface. The embedding is to be considered preliminary as we did not pay close attention to its precise geometrical properties yet—our focus so far was on guaranteeing structural, topological correctness of the layout. We thus follow up with a stage that optimizes the continuous degrees of freedom: the positions of nodes on the surface, the paths that arcs take over the surface, and even the maps of patches onto rectangles.

Optimizing the embedding of nodes, arcs, and patches individually proves to be of very limited value, due to their tight coupling (nodes necessarily are the end points of arcs, arcs necessarily are the boundaries of patches, …). We thus proposed an integrated approach[G] that performs a global optimization of all continuous degrees of freedom at once. It is based on computing a global mapping of the entire surface onto an abstract domain whose structure is derived from the quad layout. This global mapping effectively encodes the embedding of all nodes, arcs, and patches in a combined manner. This mapping can then be optimized with respect to a variety of objectives, aiming, for instance, for alignment of arcs with principal curvature directions. Fig. 7 illustrates the process. A particular advantage of our approach to this embedding optimization problem is its independence of the preliminary arc embedding's quality. Fig. 8 shows the results of this optimization on a number of exemplary surfaces.

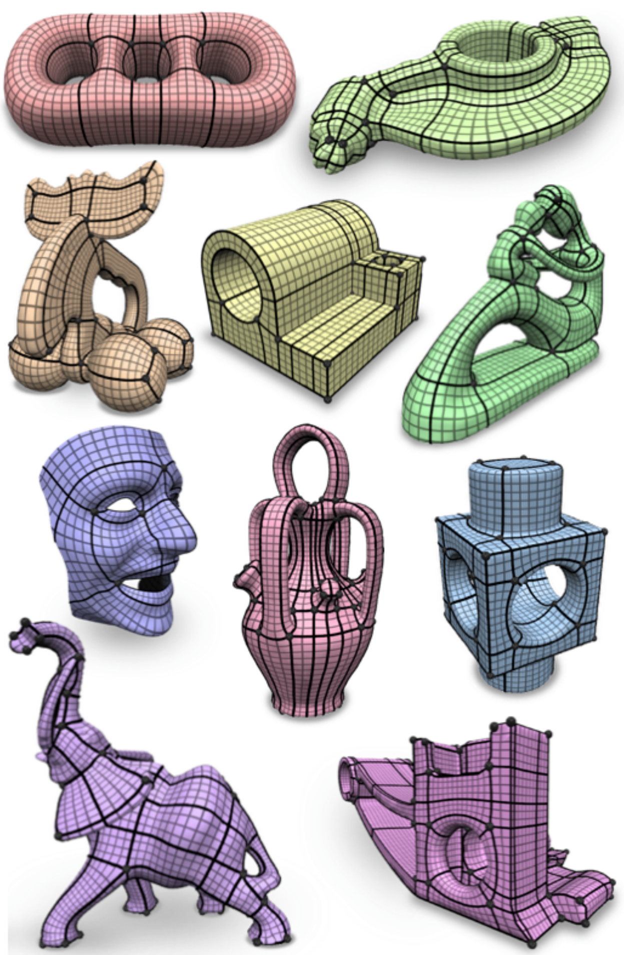


Fig. 8 Examples of optimized quad layouts (black). Patch embeddings are visualized using a grid texture.

Interactive Layouting

In the automatic techniques described above, we took generic layout quality aspects into account. For specific use cases, it can be important to consider more concrete, specialized quality objectives. These, however, can be hard to formalize, subjective, driven by aesthetics, etc. The related field of quad mesh generation faces a similar issue: based on some quality measure a compromise between alignment, orientation, and element shape needs to be made. A solution which found success in practice in this field is the inclusion of the user in the process—instead of aiming for full automation. Several major 3D sculpting and modeling software packages have in recent years been equipped with quad remeshing features which follow this paradigm and rely on high-level user interaction, e.g. regarding the specification of alignment and element sizing. This allows the user to tune the result to meet the given requirements—even if no formal description thereof is available.

Following this paradigm, we presented an interactive quad layout design system, Dual Strip Weaving[F], that explicitly takes the user into the loop to enable respecting application-dependent quality criteria which can be hard to formalize but easy to judge for an expert. Our approach provides high-level interaction tools and visual guides which keep the amount of user effort low.

We operate in the same conceptual dual setting as the automatic approach: the atomic operation is the creation (or deletion, or modification) of an entire dual loop. This insures preservation of structural consistency at all times. To provide full flexibility, however, we do not, as in the automatic approach, predetermine the nodes, but let them emerge freely. This requires a more powerful mathematical model for the dual loops: we use Euler's *elastica* model instead of geodesic loops (incorporating curvature minimization in addition to length minimization). The layout design process, which consists mainly of covering the surface with these dual strips, crossing each other, can be seen in loose analogy to the weaving or plaiting of baskets (cf. Fig. 9). Fig. 10 illustrates the main aspects of the quad layout design process. Fig. 11 shows snapshots of the interactive process on a complex example surface.

Fig. 9 Basket plaiting: a real-world analogy for the Dual Strip Weaving method. (image courtesy of Jonas Hasselrot)

The key novel property of this system is the fully automatic handling of structural consistency aspects, such as conformity and four-sidedness, while at the same time providing the user with full design flexibility.

Post-Thesis Developments

Since the time of finishing the thesis, there have been a number of interesting developments, both on our side and due to other members of the research community, that build on our results, that make use of algorithmic parts, that improve on certain aspects, or that provide alternatives to various parts.

Primal Arc Generation

Recently, progress has been made regarding the determination of layout arcs not in the dual, but directly in the primal setting[I,J,K]. The main challenge in this setting is respecting the global structural constraints inherent to conforming quad layouts. It was demonstrated that this can practically be handled by formulating a global combinatorial (or binary) optimization problem and making certain simplifications and assumptions for the sake of tractability.

Alternatively, a global mixed integer optimization formulation can be used. Given the nodes, a so-called integer grid map[H] can be computed. From this map, arcs between nodes can be derived by following isolines. A major challenge in this context is dealing with the discrete, integer-valued variables in the optimization problem. We recently presented a novel efficient and robust solution for this task[D].

Also in the interactive regime primal approaches were shown to be feasible[L,T] if control over extraordinary nodes is not essential.

Improved Node Generation

Recent improvements in the area of cross field synthesis can naturally be exploited in the node generation stage. This includes multi-resolution techniques for speed-up[M,N] and strategies for increased automation and robustness in principal curvature and feature alignment[E].

Special Layout Classes

While our focus was on being general, being able to construct any valid quad layout, depending on the use case it may be of interest to focus on specific sub-classes of layouts—either because this benefits the case itself, or because it allows for a more controlled, safer, or more robust construction. One example of such restricted classes of quad layouts are layouts induced by curve skeletons of shapes[O].

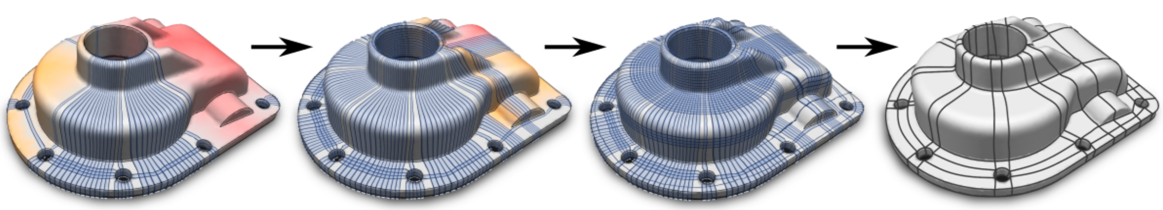


Fig. 11 Snapshots of an interactive layout design session using the Dual Strip Weaving approach. When the surface is sufficiently covered by dual strips (blue), the implied primal layout is of proper quality (right).

Technology Transfer

Cooperations are currently underway with the aim of integrating our quad mesh generation techniques[D,H] into mainstream modeling software and geometry processing libraries. Parts of our quad layout generation algorithms, particularly the embedding optimization and the structure control techniques, are envisioned to find their way into these endeavors.

Future Directions

There are currently major efforts being undertaken by numerous members of the research community to generalize or transfer the concepts and methods used for quad layout and quad mesh generation from the surface regime to the solid regime, from 2D to 3D. The goal is the partitioning of solid volumes into *hexahedral* layouts and meshes, with elements structurally equivalent to cuboids rather than rectangles. This higher-dimensional setting comes with a multitude of additional challenges, so progress is being made slowly. There already is a number of promising early results, e.g. based on frame fields[Q], polycubes[R], or structural simplification[P]. Achieving robustness is particularly challenging in this domain; we are currently working on the application of our recent robust bijective volumetric mapping approach[S] in this context.

The formalization of concrete quality criteria, specialized to application scenarios, is certainly an important area. This could serve as a basis for the development of specialized layout generation algorithms, possibly even providing strict guarantees or bounds regarding the results' quality. It would furthermore be of benefit for the proper quantitative comparison of layouts and algorithms.

Other areas where investigations are currently underway are the generation of quad meshes and layouts for dynamic rather than static shapes, and the generalizability of application scenarios that traditionally strictly rely on conforming layouts to broader classes of layouts that are non-conforming (cf. Fig. 12). This would grant a higher degree of flexibility, potentially allowing for a more efficient construction of layouts of higher quality.

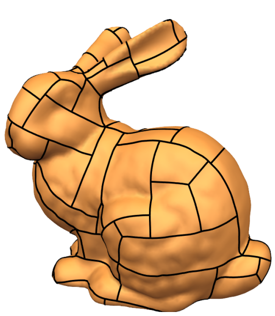


Fig. 12 Example of a *non-conforming* layout.

Acknowledgments

I thank my advisors, coauthors, and colleagues for their continual support. Work on my dissertation was funded in part by the DFG Cluster of Excellence UMIC and the ERC Advanced Grant ACROSS.

References

A FARIN G.: A History of Curves and Surfaces in CAGD. In G. Farin, J. Hoschek, and M.-S. Kim, editors, *Handbook of 3D Modeling and Graphics*, 1-22. Elsevier, 2002

B CAMPEN M.: Quad Layouts: Generation and Optimization of Conforming Quadrilateral Surface Partitions. Shaker Verlag, 2014.

C CAMPEN M., BOMMES D., KOBBELT L.: Dual Loops Meshing: Quality Quad Layouts on Manifolds. *ACM Transactions on Graphics 31*, 4 (2012).

D CAMPEN M., BOMMES D., KOBBELT L.: Quantized Global Parametrization. *ACM Transactions on Graphics 34*, 6 (2015).

E CAMPEN M., IBING M., EBKE H.-C., ZORIN D., KOBBELT L.: Scale-Invariant Directional Alignment of Surface Parametrizations. *Computer Graphics Forum 35*, 5 (2016).

F CAMPEN M., KOBBELT L.: Dual Strip Weaving: Interactive Design of Quad Layouts using Elastica Strips. *ACM Transactions on Graphics 33*, 6 (2014).

G CAMPEN M., KOBBELT L.: Quad Layout Embedding via Aligned Parameterization. *Computer Graphics Forum 33*, 8 (2014).

H BOMMES D., CAMPEN M., EBKE H.-C., ALLIEZ P., KOBBELT L.: Integer-Grid Maps for Reliable Quad Meshing. *ACM Transactions on Graphics 32*, 4 (2013).

I RAZAFINDRAZAKA F. H., REITEBUCH U., POLTHIER K.: Perfect matching quad layouts for manifold meshes. *Computer Graphics Forum 34*, 5 (2015),.

J PIETRONI N., PUPPO E., MARCIAS G., SCOPIGNO R., CIGNONI P.: Tracing field-coherent quad layouts. *Computer Graphics Forum 35*, 7 (2016).

K ZHANG S., ZHANG H., YONG J.-H.: Automatic quad patch layout extraction for quadrilateral meshes. *Computer-Aided Design and Applications 13*, 3 (2016)

L TAKAYAMA K., PANOZZO D., SORKINE-HORNUNG A., SORKINE-HORNUNG O.: Sketch-based generation and editing of quad meshes. *ACM Transactions on Graphics 32*, 4 (2013).

M JAKOB W., TARINI M., PANOZZO D., SORKINE-HORNUNG O.: Instant field-aligned meshes. *ACM Transactions on Graphics 34*, 6 (2015).

N EBKE H.-C., SCHMIDT P., CAMPEN M., KOBBELT L.: Interactively Controlled Quad Remeshing of High Resolution 3D Models. *ACM Transactions on Graphics 35*, 6 (2016).

O USAI F., LIVESU M., PUPPO E., TARINI M., SCATENI R.: Extraction of the quad layout of a triangle mesh guided by its curve skeleton. *ACM Transactions on Graphics 35*, 1 (2015).

P GAO X., DENG Z., CHEN G.: Hexahedral mesh re-parameterization from aligned base-complex. *ACM Transactions on Graphics 34*, 4 (2015).

Q JIANG T., HUANG J., WANG Y., TONG Y., BAO H.: Frame field singularity correction for automatic hexahedralization. *Transactions on Visualization and Computer Graphics, 20*, 8 (2014).

R FANG X., XU W., BAO H., HUANG J.: All-hex meshing using closed-form induced polycube. *ACM Transactions on Graphics 35*, 4 (2016).

S CAMPEN M., SILVA C., ZORIN D.: Bijective Maps from Simplicial Foliations. *ACM Transactions on Graphics 35*, 4 (2016).

T MARCIAS G., TAKAYAMA K., PIETRONI N., PANOZZO D., SORKINE-HORNUNG O., PUPPO E., CIGNONI P.: Data-driven Interactive Quadrangulation. *ACM Transactions on Graphics 34*, 4 (2015).

U CAMPEN M., HEISTERMANN M., KOBBELT L.: Practical Anisotropic Geodesy. *Computer Graphics Forum 32*, 5 (2013).

Marcel Campen *is a postdoctoral researcher at New York University. Contact him at campen@nyu.edu.*