

Quantized Global Parametrization

Marcel Campen*
RWTH Aachen University

David Bommes*
RWTH Aachen University

Leif Kobbelt*
RWTH Aachen University

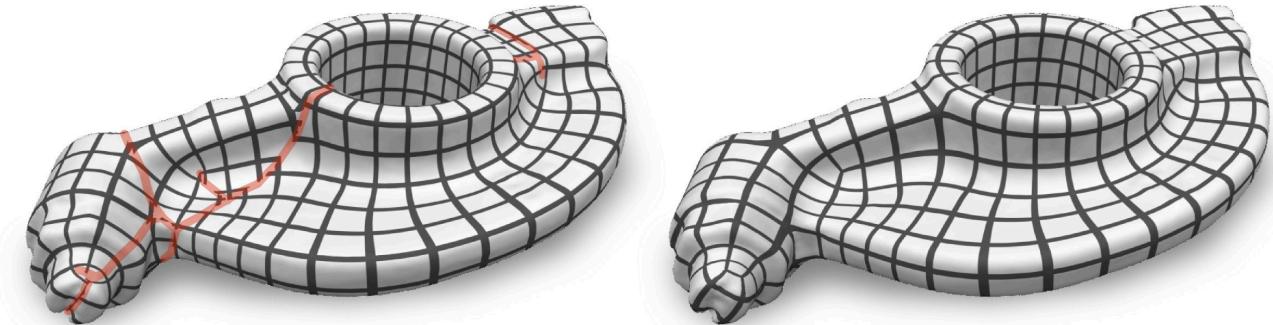


Figure 1: Left: Global parametrization with continuous rigid transitions; notice the mismatch of the depicted isolines across the red cuts. Right: Global parametrization with quantized (or integral) transitions computed robustly and efficiently by our method.

Abstract

Global surface parametrization often requires the use of cuts or charts due to non-trivial topology. In recent years a focus has been on so-called *seamless* parametrizations, where the transition functions across the cuts are rigid transformations with a rotation about some multiple of 90° . Of particular interest, e.g. for quadrilateral meshing, paneling, or texturing, are those instances where in addition the translational part of these transitions is integral (or more generally: quantized). We show that finding not even the optimal, but just an arbitrary valid quantization (one that does not imply parametric degeneracies), is a complex combinatorial problem. We present a novel method that allows us to solve it, i.e. to find valid as well as good quality quantizations. It is based on an original approach to quickly construct solutions to linear Diophantine equation systems, exploiting the specific geometric nature of the parametrization problem. We thereby largely outperform the state-of-the-art, sometimes by several orders of magnitude.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling;

Keywords: quad meshing, interval assignment, T-mesh, rounding

Links: DL PDF

1 Introduction

The parametrization of surfaces is one of the basic tools used in countless Computer Graphics and Geometry Processing applica-

tions [Floater and Hormann 2005]. Of particular interest are *global* parametrizations that cover the entire surface. Unless the surface is of disc topology, such global parametrizations over \mathbb{R}^2 inevitably need to consist of *charts* (or a single chart), with non-identity transitions across the *cuts* [Sorkine et al. 2002; Gu et al. 2002; Lévy et al. 2002; Dong et al. 2006], because the surface is not globally homeomorphic to a subset of the plane.

Seamless Parametrization

In recent years a focus has been on parametrizations where the transition functions across the cuts are not arbitrary but of a very restricted class: rigid transformations with a rotation angle of some multiple of 90° – sometimes termed *seamless* parametrizations [Myles and Zorin 2012; Purnomo et al. 2004] (cf. Figure 2 middle). Such transitions imply that the *singularities* of the parametrization are of a very specific type: the Gaussian curvature of the induced metric is a multiple of $\frac{\pi}{2}$ at every singularity. For most application scenarios of such seamless parametrizations it is necessary that in addition the translational part of these transitions is integral (or, almost equivalently, the singularities have integer parameters) (cf. Figure 2 right). More generally we can speak of *quantized* translations or parameters, which are integer up to some global factor. Such parametrizations are sometimes called *integer-grid maps* [Bommes et al. 2013b]. They are important for quadrilateral meshing [Kälberer et al. 2007; Bommes et al. 2009; Pietroni et al. 2011; Li et al. 2011; Bommes et al. 2013b; Campen and Kobbelt 2014b; Panizzo et al. 2014; Ebke et al. 2014] and paneling [Liu et al. 2011; Pietroni et al. 2015], as well as high-quality texturing [Ray et al. 2010].

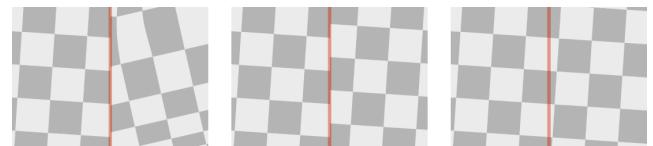


Figure 2: Behavior of various classes of parametrizations across cuts (red). Left: arbitrary transitions. Middle: so-called “seamless” parametrization with rigid transitions with rotational continuity across the cut. Right: quantized, integral parametrization.

(c) 2015 Marcel Campen, David Bommes, Leif Kobbelt.

This is the authors' version of the work. It is posted here for your personal use. Not for redistribution. The definitive version was published in Proceedings of the SIGGRAPH Asia 2015 Conference, ACM Transactions on Graphics, vol. 34, 6. <http://dx.doi.org/10.1145/2816795.2818140>.

*e-mail:(campen|bommes|kobbelt)@cs.rwth-aachen.de

While a number of powerful methods for the generation of *continuous* seamless parametrizations has been presented during the last decade, most notably by Myles et al. [2012; 2013; 2014] and as a component of quadrilateral meshing techniques [Kälberer et al. 2007; Bommes et al. 2009], the issue of quantization has not yet been addressed to a satisfactory level.

Integral Parametrization

At first sight quantization may seem like a trivial problem: just compute a seamless parametrization, round (or quantize) its continuous translations, and re-optimize the parametrization with the translation variables constrained to these rounded values. Unfortunately, the translations are not mutually independent, and they are coupled with the singularity parameters. Hence, simply rounding to the closest integers can lead to infeasibility of the constrained problem as well as to degeneracies in the resulting parametrization due to singularities forced to collapse in parameter space (cf. Figure 3). While the first problem can be solved by determining a set of independent variables, the latter problem is much more severe.

In previous work, three different strategies have been followed for parametrization quantization:

- Direct Rounding [Kälberer et al. 2007]
- Iterative Rounding [Bommes et al. 2009]
- Branch-and-Bound Rounding [Bommes et al. 2013b]

Direct Rounding rounds independent variables of a continuous solution to integers and then re-solves the parametrization optimization system with these variables fixed. Note that in this case the parametrization problem needs to be solved twice.

Iterative Rounding rounds the variables one-by-one, each time re-solving the system with an increasing number of variables fixed. Note that here the parametrization problem needs to be solved $n+1$ times, where n is the number of integer variables. This typically leads to lower distortion in the final result [Bommes et al. 2013a]. Both methods take no measures to prevent singularities from collapsing. Hence, especially when the quantization is performed to a coarse scale, there is a significant risk that the final parametrization will be degenerate – or infeasible, if anti-degeneracy constraints [Lipman 2012; Bommes et al. 2013b] are employed. In other words: these quantization methods are non-robust.

Branch-and-Bound Rounding is more powerful. It does not determine just one quantization but explores virtually all possible discrete variable assignments. Note that here the worst case asymptotic complexity is exponential in the number of discrete variables. By checking the individual resulting parametrizations for degeneracies it is possible to determine a valid quantization, as well as (in theory) ultimately the optimal one. The employed general purpose solver is unaware of the geometric nature of the problem and

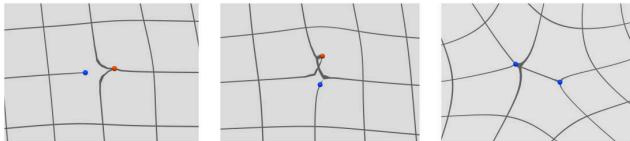


Figure 3: *Naïve rounding to integral transitions does easily force singularities to collapse in parameter space. No regular, locally injective parametrization can then be obtained, because degenerate or inverted elements are implied. This, for instance, leads to broken isolines (black), non-four-sided isoline-polygons, etc., in the vicinity of singularities on the surface (red and blue).*

the implications of its internal optimization steps (cf. Section 3.4). To achieve practical performance thus an extreme-decimation technique and simplification heuristics [Bommes et al. 2013b] were proposed to reduce complexity for a significant speed-up, at the expense of weakening the notion of optimality. Still, for problem instances with some hundred singularities it can take hours or days until a valid solution is found.

1.1 Contribution

We present a novel technique to generate valid quantizations, thus integral global parametrizations. In contrast to Direct Rounding and Iterative Rounding, it is guaranteed to yield *valid* results, i.e. it addresses the inherent robustness issue of these approaches. Compared to Branch-and-Bound Rounding, results of similar quality are obtained in a fraction of the time required, i.e. the performance issue of this approach is addressed. In complex instances with thousands of singularities, our technique yields high quality results within minutes, while Branch-and-Bound Rounding typically does not find any valid solution even after 24h of allotted run time.

1.2 Outline

In Sections 3 and 4 we introduce the formal background and motivate our approach. In Section 5 then the fundamental data structures are introduced, before the core of the actual quantization algorithm follows in Section 6. In Section 7 we describe how the final integral parametrization is obtained, evaluate and compare results in Section 8, and discuss extensions and broader applicability in Sections 9 and 10.

2 Related Work

Besides the global parametrization methods already discussed in the previous section, which are *unstructured* in the sense that arbitrary cut graphs are used, there are a number of *structured* approaches, which rely on specific classes of cut graphs, in particular chart layouts with triangular [Khodakovsky et al. 2003; Pietroni et al. 2010; Guskov et al. 2000] or quadrilateral [Eck and Hoppe 1996; Purnomo et al. 2004; Dong et al. 2006; Zhang et al. 2010; Tarini et al. 2011; Campen and Kobbelt 2014b] regions. Methods which use cuts without restrictions on the transitions (e.g. [Sorkine et al. 2002; Lévy et al. 2002]) are less related.

The quad layout based variants particularly lend themselves to (continuous) seamless parametrization. Even the quantization problem is quite easily solved in these cases (the reasons are elucidated in Section 10.1), however, at the cost of first having to construct the quad layout. Unfortunately, doing this in such a way that the layout allows for a high-quality parametrization is a very intricate problem in itself [Campen et al. 2012; Campen and Kobbelt 2014a], apparently not less complicated than the problem we address here. In fact, a recent proposal for quad layout generation was to derive the layout from a coarsely quantized global parametrization [Bommes et al. 2013b].

Non-conforming layouts (i.e. chart layouts with T-junctions) have also been used for global parametrization purposes. The regions can be quadrilateral [Bommes et al. 2008; Myles et al. 2010] or of more general shape [Tong et al. 2006]. We discuss the problem of quantizing such T-mesh based parametrizations in Section 10.1.

A way to parameterize surfaces of arbitrary topology without the need for explicit cuts is to use so-called *periodic parametrizations* [Ray et al. 2006]. These do not directly yield a conforming grid structure on the surface.

3 Formal Setup

3.1 Global Parametrization

We consider piecewise linear surface parametrizations. These can be represented by (u, v) -coordinates (*parameters*) at each vertex of a triangle mesh. The mesh is virtually cut along a subset of its edges, the so-called *cut-edges*. Hence multiple copies of the vertices along these cuts exist and carry their own (u, v) -coordinates. Per cut-edge e a (directed) transition function τ_e exists, which relates the (u, v) coordinates of such pairs of vertex copies:

$$\tau_e(u_i, v_i) = (u_j, v_j). \quad (1)$$

Definition 1. We say the parametrization is **seamless** if for all τ

$$\tau(u, v) = R_{90^\circ}^r(u, v) + (s, t), \quad r \in \{0, 1, 2, 3\}, (s, t) \in \mathbb{R}^2.$$

The so-called *matchings* r can be considered not to be part of the actual parametrization problem. They are derived a priori, e.g. from cross fields [Kälberer et al. 2007; Bommes et al. 2009] or cone metrics [Myles and Zorin 2012]. Vertices at which the parametrization needs to be singular (in particular vertices around which the matchings do not sum to a multiple of 4) are called *singularities*. Let their set be S and their number be n from now on.

Definition 2. We say the parametrization is **integral** if for all τ

$$\tau(u, v) = R_{90^\circ}^r(u, v) + (s, t), \quad r \in \{0, 1, 2, 3\}, (s, t) \in \mathbb{Z}^2$$

and $(u_i, v_i) \in \mathbb{Z}^2$ for all $i \in S$.

Note that both conditions, $(s, t) \in \mathbb{Z}^2$ and $(u_i, v_i) \in \mathbb{Z}^2$, are necessary; they are strongly related but not equivalent. In particular, $(s, t) \in \mathbb{Z}^2$ implies $(u_i, v_i) \in \frac{1}{2}\mathbb{Z}^2$ for most singularities $i \in S$ [Ebke et al. 2013; Kälberer et al. 2007] but it does not imply actual integrality. Vice versa, singularity parameter integrality implies $(s, t) \in \mathbb{Z}^2$ on genus $g = 0$ surfaces, but in general $4g$ continuous degrees of translation remain free.

The variables of the parametrization problem are the (u, v) -coordinates (per vertex) and the (s, t) -translations (on cuts). Note that this set of variables is not completely independent; for instance, the parameters of copies of vertices at cut edges are related by multiple transitions, and singularity parameters are related to translations by fixpoint formulas [Ebke et al. 2013], implying certain interdependencies. In the following we denote by z those variables (of either kind) which need to be integral for an integral parametrization.

Based on this an energy functional to be minimized is formulated:

$$E(\{(u, v)\}, \{(s, t)\}) \rightarrow \min, \quad \text{subject to (1).} \quad (2)$$

Our technique is oblivious to the concrete nature of this functional, so, for instance, a field-guided parametrization energy [Kälberer et al. 2007; Bommes et al. 2009], a least-squares conformal parametrization energy [Lévy et al. 2002], or an as-rigid-as-possible parametrization energy [Myles and Zorin 2012] could be used. We use the field-guided version [Bommes et al. 2009] for our demonstrations in this paper, as it is very flexible and quite prevalent in our targeted areas [Kälberer et al. 2007; Bommes et al. 2009; Pietroni et al. 2011; Li et al. 2011; Bommes et al. 2013b; Ebke et al. 2014; Campen and Kobbelt 2014b; Panozzo et al. 2014; Campen and Kobbelt 2014a; Jiang et al. 2015].

Definition 3. A parametrization is **regular** if all triangles have a positive area in parameter space.

This leads to the further constrained optimization problem

$$E(\{(u, v)\}, \{(s, t)\}) \rightarrow \min, \quad \text{s.t. (1) and regularity.} \quad (3)$$

Under reasonable topological assumptions on the provided matchings¹, the existence of seamless as well as integral solutions to (3) (although not necessarily at the given input mesh resolution/discretization) is a corollary of the quadrilateral cell decomposition existence proof by [Jucovič and Trenkler 1973]. In practice the discretization as well as the non-convexity of the regularity constraints can be an issue; we discuss this aspect in Section 7.1.

3.2 Quantization

Definition 4. A **quantization** σ is the assignment $\{z_i = c_i, \dots\}$, $c_i \in \mathbb{Z}$, of integers to some translations and singularity parameters, i.e. certain variables of (3). A quantization is called **consistent** if no contradictions with respect to interdependencies of the assigned variables are implied, i.e. if (1) is respected. If all integer degrees of freedom are fixed, the quantization is called **complete**, otherwise **partial**. A quantization σ^* is called **completion** of σ if $\sigma \cup \sigma^*$ is a complete quantization.

With slight abuse of terminology we shall also call a parametrization which respects σ a quantization. It will always be clear from the context whether an assignment or a parametrization is meant.

As a more general form of quantization one could require $c_i \in s\mathbb{Z}$ for some global constant $s \in \mathbb{R}$. As this corresponds to a mere global rescaling of the parametrization, we, w.l.o.g., restrict to integral values in the following.

Fundamental Problem: Not every consistent quantization σ allows for a regular parametrization, i.e. $\{(3) \text{ s.t. } \sigma\}$ can be infeasible. The quantizations obtained by Direct Rounding or Iterative Rounding can exhibit exactly this problem, especially in the case of coarse quantization scales. Our method addresses the problem of finding quantizations for which $\{(3) \text{ s.t. } \sigma\}$ is feasible.

Definition 5. A (complete or partial) **quantization** σ is **valid** iff $\{(3) \text{ subject to } \sigma\}$ is feasible, i.e. if there exists a regular parametrization that respects σ . It is called **validly completable** if a completion σ^* exists such that $\sigma \cup \sigma^*$ is valid.

3.3 Feasibility

Probably the most common reason for infeasibility in practice is the parametric coincidence of singularities. If a quantization makes two singularities coincide parametrically, this can force the mesh region inbetween to either degenerate or partially invert, i.e. some triangles necessarily assume a zero or negative area in parameter space (cf. Figure 3), intuitively because there is just no room for these triangles between the two singularities in parameter space. As such violation of (3) is unacceptable or at least undesirable in all known applications, such quantizations must be avoided.

Note that parametric coincidence of singularities i and j does not mean $(u_i, v_i) = (u_j, v_j)$, because in the given setting transitions need to be taken into account. So instead this means $(u_i, v_i) = \tau_p(u_j, v_j)$, for some path p on the surface connecting i and j . τ_p is the transition along p , i.e. the concatenation of all transitions of cut-edges crossed by the path. As τ_p does not depend on the actual path p but just on its path homotopy class c (with respect to the surface punctured at the singularities) [Myles et al. 2010], this is equivalent to $(u_i, v_i) = \tau_c(u_j, v_j)$ for some path homotopy class c .

¹The Poincaré-Hopf theorem must be fulfilled. This is guaranteed by all known singularity (thus matching) determination algorithms. Furthermore, index ≥ 1 singularities and a particular special case must be excluded: a single pair of singularities on a genus 1 surface [Jucovič and Trenkler 1973].

However, the absence of such coincidences is neither a necessary, nor known to be a sufficient condition for feasibility. It is not necessary because parametrization regularity only requires local, not global injectivity of the parametrization. The chart(s) of a regular seamless parametrization thus form *overlapping* polygons in parameter space [Weber and Zorin 2014].

Neglecting the possible insufficiency, one could think of specifying constraints $(u_i, v_i) \neq \tau_c(u_j, v_j)$ for all i, j, c and providing them to a generic solver – but these constraints are very complex in nature. They are disjunctions of strict inequality constraints, $(u_i < s_j) \vee (u_i > s_j) \vee (v_i < t_j) \vee (v_i > t_j)$, where $(s_j, t_j) = \tau_c(u_j, v_j)$, and thus are non-convex. This and the fact that in general there are infinitely many path homotopy classes c precludes this approach.

The *singularity spacer* strategy employed by Bommes et al. [2013b] can be interpreted as a strong simplification using two heuristics: one to select only a small finite subset of these constraints, and one to choose one of the four disjuncts, thus making the constraints convex. Due to the first heuristic it does not prevent all invalid quantizations from arising during optimization, and the second excludes a subset of the valid quantizations.

3.4 Our Approach

We follow a novel approach to effectively and efficiently circumvent these problems. While previous approaches (Iterative Rounding, Branch-and-Bound Rounding) are conceptually based on incrementally building chains of *partial* quantizations

$$\emptyset \subset \sigma_1 \subset \sigma_2 \subset \sigma_3 \subset \cdots \subset \sigma_k, \quad \sigma_k \text{ complete,}$$

our method is based on building a series of transformations between entirely *complete* quantizations

$$\sigma_0 \rightarrow \sigma_1 \rightarrow \sigma_2 \rightarrow \sigma_3 \rightarrow \cdots \rightarrow \sigma_k, \quad \sigma_k \text{ valid.}$$

This change of strategy has significant implications (and ultimately is the key to achieving efficiency and robustness): for an arbitrary partial quantization we are not aware of any polynomial time algorithm to decide whether it is validly completable. Previous approaches are thus essentially “in blind flight” during the chain building – in the worst case only in the very end (in case of a branching strategy: at a leaf of the branch tree) invalidity can be recognized.

By operating within the class of complete quantizations only, we get rid of this complication: only the validity of complete quantizations, not the valid-completeness of partial quantizations is of concern. We will see that, by performing a change of variables, one can decide this validity of a class of complete quantizations in empirically linear time. This ultimately allows us to transform one (valid) complete quantization into another in quasilinear time, making building the series of transformations very efficient.

Method Overview

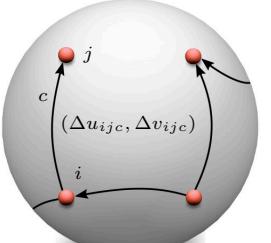
1. Solve (3) to obtain seamless parametrization P .
2. Construct valid complete quantization σ guided by P :
 - (a) Extract motorcycle graph T of P .
 - (b) Evolve the quantization σ using T ’s Hilbert basis.
3. Solve $\{(3) \text{ s.t. } \sigma\}$ to obtain quantized parametrization.

Steps 1 and 3 are based on previous work and described in Section 7. The actual quantization is performed in step 2. Section 4 motivates the choice of the so-called motorcycle graph as underlying structure, Sections 5 deals with step 2(a), i.e. its construction as well as its properties, and Section 6 describes the central step 2(b).

4 Differential Parameters

Imagine an arbitrary embedded directed graph on the surface, having the singularities as nodes, connecting all of them. Each edge of the graph can be identified with a tuple (i, j, c) , representing its start node i , its end node j , and the path homotopy class c it belongs to. We can assign to each edge a vector $(\Delta u_{ijc}, \Delta v_{ijc})$, whose function is specifying the parametric difference between i and j . Fixing $(u_0, v_0) = 0$ and using the definition

$$(u_j, v_j) = \tau_c(u_i, v_i) + (\Delta u_{ijc}, \Delta v_{ijc})$$



these *differential parameters* $(\Delta u_{ijc}, \Delta v_{ijc})$ obviously imply all singularity parameters (assuming there are no contradictions). If the graph includes (undirected) cycles which generate the surface’s fundamental group, also all translations are implied, i.e. the additional $4g$ degrees of freedom (cf. Section 3.1) are fixed. If the differential parameters are integral, so are all the singularity parameters and translations. Therefore, we can quantize the differential parameters instead. Note that depending on the choice of graph the number of edges can be between $\max(0, n-1) + 2g$ (a spanning tree of the n singularities and $2g$ homology basis loops) and infinite (because of infinitely many homotopy classes c).

The motivation to use these differential parameters is the following: with the right choice of graph structure we enable the design of an efficient test for quantization validity. The key is choosing a graph that forms a cell decomposition of the surface, i.e. its edges do not intersect (only at the nodes) and each cell is homeomorphic to a disc. The vectors $(\Delta u_{ijc}, \Delta v_{ijc})$ of the edges bounding a cell then form a polygon in parameter space (after transformation into a common chart, if necessary).

To get a first idea of the beneficial implications of this, assume that all these vectors are non-zero and all these polygons are simple and have positive signed area. Then all singularities are separated and one can conclude that the quantization is valid: each surface cell can regularly be mapped to its polygon, and the union of these maps provides a witness parametrization (not necessarily linear per triangle) which demonstrates feasibility.

The situation gets particularly simple when we construct the graph such that the $(\Delta u_{ijc}, \Delta v_{ijc})$, by construction, form axis-aligned rectangles instead of arbitrary polygons: they are always convex, thus simple, and their orientation can be controlled easily through the signs of their height and width. To enable such cells we allow the graph to contain additional nodes besides the singularities. Note that in this case a quantization can be valid even if some differential parameters are zero and polygons degenerate – as long as the singularities are separated by *some* positive polygons.

5 T-Mesh

The particular type of embedded graph we consider is the edge graph of a so-called *T-mesh*, an arbitrary non-conforming partition of the surface into four-sided cells, whose nodes include (but are not restricted to) the singularities.

As we want to restrict to axis-aligned rectangles in parameter space, there is no need to assign a differential tuple $(\Delta u_e, \Delta v_e)$ to each edge e ; one differential parameter, denoted x_e , suffices. It can intuitively be understood as the *parametric length* of e . In order to restrict to non-negative signed area, we orient all edges such that the constraints $x_e \geq 0$ imply this. It will later become clear that

a restriction to $x_e > 0$ (while it would simplify the matter) would often exclude the quantizations of highest quality.

As x (the set of all differential parameters x_e) implies z (the integer variables of the integral parametrization problem), an assignment of integer values to x can be called quantization as well.

Due to the non-negativity constraints on x , the assigned directions of the T-mesh edges imply a partial ordering of the graph nodes in parameter space a priori. To not let this impair the quality of the quantizations that can be obtained, the T-mesh should be constructed in such a way that for each edge the directional choice is natural and unlikely to preclude any good quantization.

5.1 Motorcycle Graph

To this end we construct the T-mesh graph directly based on the continuous parametrization that is to be quantized. We solve (3) (cf. Section 7) to obtain a continuous, seamless parametrization, and extract this parametrization’s *motorcycle graph* T (cf. Figure 4). Its construction has been described for the discrete case (quad mesh) by Eppstein et al. [2008], and it can analogously be constructed by tracing parametrization isolines (*separatrices*) instead of quad mesh edges, starting from the singularities of the parametrization. We use sanitization and tracing algorithms as described by Ebke et al. [2013] to perform the tracing in a numerically robust manner. If there are no singularities, one arbitrary, regular starting point is picked. T is known to be a cell decomposition of the surface [Eppstein et al. 2008].

Each edge of this graph is in line with either the parametric u - or v -direction of the optimal continuous parametrization., i.e. we can orient it in positive parametric direction in a natural manner.

It is important to realize that we only make use of T ’s topological structure – its geometric embedding (as shown in Figure 4) is of no concern. In particular will the final quantized parametrization (cf. Section 7) not be forced to align to the edges of T on the surface in any way. We also note that, while T is not of minimal size (cf. Section 4), the number of its nodes and edges still is only $O(n)$ [Eppstein et al. 2008].

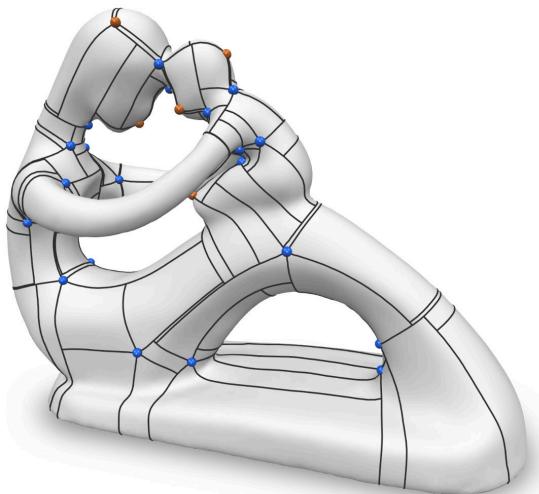
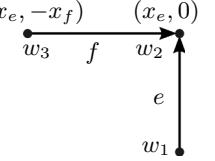


Figure 4: Motorcycle graph T -mesh obtained from a continuous seamless parametrization. The black graph edges emanate from the parametrization’s singularities (red and blue) and end in a T-junction (or, rarely, in a cross-junction or at another singularity).

5.2 Checking Validity on a T-Mesh

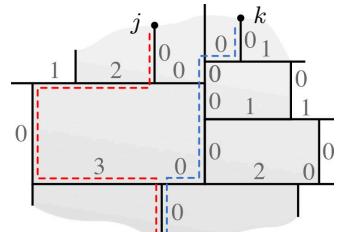
Assume a complete assignment of x -values to the edges of T is given. To decide whether this quantization is valid, we exploit the fact that T forms a cell decomposition and each cell has non-negative dimensions. Conceptually, for a singularity i we consider the set $P^0(i)$ of cells of T which have zero distance (in terms of the cell dimensions specified by x) to i . Notice that only a singularity j incident to a cell from $P^0(i)$ can be unseparated from i , violating validity. Due to non-negative cell dimensions, $P^0(i)$ is a connected region and all incident edges can thus be conquered locally, using a graph search starting from i .

To get an idea of this process, consider a node w_1 of T and an incident edge e , its other incident node being w_2 . Assume (w.l.o.g.) that e is an edge aligned to the u -direction at w_1 . Then w_2 has parametric difference $(\pm x_e, 0)$ (in w_1 ’s local system) from w_1 along e . The sign depends on whether we move with (+) or against (−) e ’s direction. Now consider following another edge f incident to w_2 , ending at w_3 . Then w_3 has difference $(\pm x_e \pm x_f, 0)$ or $(\pm x_e, \pm x_f)$ from w_1 , depending on whether f is a straight extension of e , or a left or right turn is made from e to f . (Note that at a singular node “straight” is not well-defined, but this case will not be of interest in the following).



Because the motorcycle graph forms a subdivision of the parameter space into axis-aligned convex cells, there exists a weakly- u -monotone or weakly- v -monotone path between every pair of nodes [Dumitrescu et al. 2013]. Weakly- u/v -monotone means that the u/v -coordinate of the difference never decreases (or never increases) as we follow the path from node to node. For our case (being only interested in P^0) this implies that it suffices to consider paths along which one of the two coordinates of the accumulated parametric difference always stays zero.

We exploit this when exploring the paths of P^0 using a breadth-first graph search: paths which started, e.g., with moving along a u -edge *with* its direction are never extended by walking along u -edges *against* their direction. Due to the monotony property, we can stop at nodes where both coordinates become non-zero. Furthermore, if one of the coordinates is non-zero at a node, it is not necessary to move along edges which bring it even further away from zero, unless there is no other edge to choose. In this way the search is restricted to the edges in the P^0 region. The inset illustrates the P^0 -region (grey) of singularity i and two exemplary paths (red and blue) potentially followed, reaching singularities j and k with difference $(0, 0)$.



We perform this search from each singularity. If in any case a singularity is reached with zero parametric distance, we conclude that the quantization is invalid. Otherwise, it is valid.

It is not easy to derive a tight asymptotic complexity bound for this algorithm. $O(n^2)$ is easy to show, because the graph search is in class $O(n)$ and executed n times. In the case of all x_e being positive, the complexity is $O(n)$. In the case of all x_e being zero it is $O(n)$ as well (the first search already provides the answer). Empirical observations indicate an overall linear behavior also for in-between cases.

5.3 Differential Parameter System

As T is not a minimal graph, there are certain interdependencies between the associated differential parameters, i.e. certain consistency conditions must be met. For each cell of T we have the consistency conditions that the parametric lengths of the edges on opposite sides S_0 and S_2 as well as S_1 and S_3 sum up to the same value, i.e.

$$\sum_{e \in S_0} x_e = \sum_{e' \in S_2} x_{e'}, \quad \sum_{e \in S_1} x_e = \sum_{e' \in S_3} x_{e'}. \quad (4)$$

These equations form an underdetermined, homogeneous linear Diophantine equation system \mathcal{E} (two rows per cell, one column per edge of T) of the general form

$$Ax = 0, \quad x_i \in \mathbb{Z} \geq 0. \quad (5)$$

Any solution to this equation system is a consistent complete quantization (a consistent assignment of integer values to all discrete variables, our differential parameters x). In general, finding the nontrivial solutions, even without consideration of conditions regarding validity and the optimization objective, is an NP-complete problem [Tomás and Filgueiras 1997]. Fortunately our system has certain specific properties (its coefficients are from $\{-1, 0, 1\}$ only, each column contains exactly two non-zeroes – or one, if boundaries are present, cf. Section 9) which we will exploit in the following to overcome this. The geometric T-mesh-based interpretation of the system will prove to be very helpful to develop an intuitive understanding of the involved concepts and algorithms.

Obviously, the trivial quantization $x = 0$ is a solution to this system, but we are clearly not interested in this one (besides being implausible, it is invalid). One can show that every T-mesh permits a solution $x \neq 0$ (cf. Appendix A.1). Furthermore one can show that every T-mesh whose edges lie on separatrices of a seamless parametrization (like our motorcycle graph T , by construction) permits a solution where every $x_i > 0$ (cf. Appendix A.2). The existence of one such solution of course implies the existence of infinitely many solutions (e.g. integer multiples of this solution). Of all solutions we are interested in the ones which are valid and allow for quantized parametrizations which are good with respect to the objective \mathbf{E} . We will thus let the unquantized minimizer of \mathbf{E} guide the solution finding process described in the following section.

6 Quantization

We are interested in a consistent quantization which in addition is valid. Being consistent means obeying (5). The space of solutions to (5) is a monoid; each solution (i.e. each consistent complete quantization) x can be written as a linear combination $x = a_1 v_1 + \dots + a_m v_m$, with non-negative integer coefficients $a_i \in \mathbb{Z} \geq 0$, of a (unique) set of *generating vectors* (also called extremal rays or Hilbert basis) $\{v_0, \dots, v_m\}$ [Rockafellar 1970]. Because in general the monoid is not free, the number m of generating vectors can be exponential in the dimension of the system [Tomás and Filgueiras 1997].

Even though our equation systems do have a restricted structure, experiments revealed that the number of generating vectors often is so large that they could not even be kept in main memory. Correspondingly, their computation [Nožička et al. 1974; Schilling et al. 2000] is very time consuming for anything but the simplest models.

6.1 Algorithm Overview

To avoid these problems, we follow the strategy of constructing individual generating vectors v_i on demand and adjusting their coefficients a_i in the linear combination, transforming one consistent complete quantization into another.

This raises a number of questions: which quantization to start with, which v_i to construct next (and how), how to adjust a_i then? And how to do this such that the final quantization is guaranteed to be valid, and close to the unquantized minimizer of \mathbf{E} ?

We start with the trivial solution, i.e. $a_i = 0, \forall i$, implying a consistent complete quantization σ_0 – which, however, is invalid and of bad quality. Considering that validity is a hard constraint and parametrization quality a soft goal, we then follow a two-stage approach giving validity the appropriate priority:

- **Stage I:** Iteratively transform the invalid complete σ_0 into a valid complete quantization σ_j .
- **Stage II:** Iteratively transform the valid complete σ_j into a valid complete quantization σ_k with better quality.

In both stages we confine all transformations to the space of consistent complete quantizations. Furthermore, in the second stage we restrict all transformations to the space of valid quantizations, thereby guaranteeing a valid result.

Note that in order to stay within the space of consistent quantizations, we cannot simply alter the value of one variable to transform one quantization into another one: the smallest possible differences between two consistent quantizations are the generating vectors!

We proceed in a greedy, edge-centric manner: we select the T-mesh edge e whose current value is (in a yet to be defined sense) worst and construct a generating vector with $v_i[e] \neq 0$, such that a change of the coefficient a_i by 1, corresponding to addition or subtraction of v_i to/from the current quantization, affects $x[e]$.

6.2 Constructing Generating Vectors

To be as flexible as possible, we would like to make small steps, i.e. changes as small as possible. This means we would like to construct a generating vector v with $v[e] = 1$, changing $x[e]$ in the smallest possible increment. Changing $x[e]$ requires changes to other variables to keep consistency; technically, v needs to lie in A 's nullspace. To achieve this, we project v (with $v[e] = 1$, and 0 in all other components) into A 's nullspace, with the constraint $v[e] \neq 0$ (we will later see that $v[e] = 1$ is a too strong constraint; $v[e] = 2$ must also be permitted for certain circumstances).

We incrementally perform this projection as follows. There are two rows, r_i, r_j , in A with $r_i[e] = \pm 1$ and $r_j[e] = \pm 1$, for all other rows $r[e] = 0$. This means the entries i and j of Av with $v[e] = 1$ (and 0 otherwise) do not vanish. This can be remedied: let's consider row r_i with (w.l.o.g.) $r_i[e] = 1$; there is at least one other edge f with $r_i[f] = -1$. If we additionally increase $v[f]$ by 1, the i -th entry of Av vanishes again. But there is one other row k with $r_k[f] = \pm 1$, so now $Av[k] \neq 0$. This process can be repeated. If at some point we reach row j , we have constructed a vector v with $Av = 0$. If the cycle we made through the matrix is simple

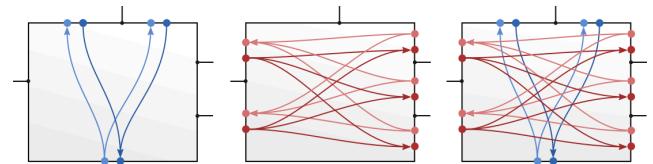


Figure 6: Directed graph D within a single cell of T-mesh T with several T-junctions. On the left only the vertical arcs are shown for clarity, in the middle the horizontal arcs, and on the right the complete set of arcs. Generating vectors of T 's constraint system (5) correspond to elementary circuits of this graph D .

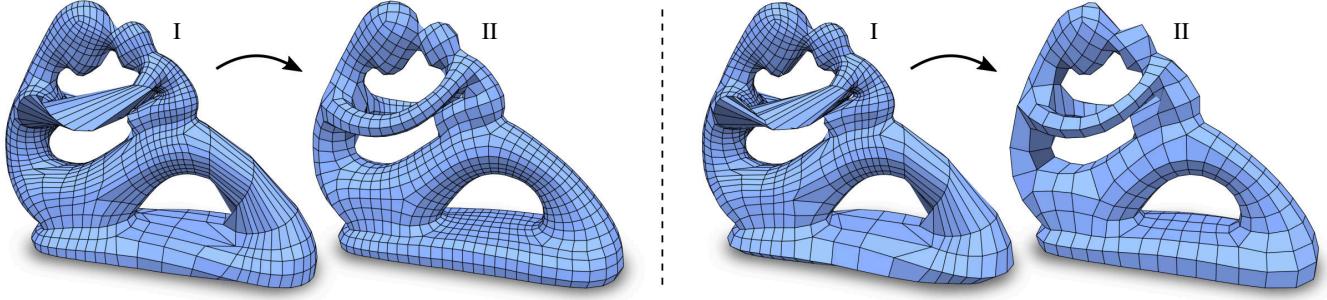


Figure 5: Quantizations obtained after stage I and stage II for two different quantization scales (i.e. target edge lengths), depicted by quad meshes extracted from the quantized parametrization. Note that the result of stage I is guaranteed to be valid, thus implies a regular integral parametrization. While valid, it is partially too coarse, partially too fine, and generally quite “twisted”. This is due to all x_e being strictly positive after stage I. This is well remedied in the final result of stage II.

(i.e. cannot be decomposed into multiple cycles) v is a generating vector, otherwise it is the sum of multiple generating vectors.

For illustration, it is useful to consider a geometric interpretation of the generating vectors. We can build a directed graph D based on the T-mesh T which models the above iteration through matrix A . Its construction for one cell of T is shown in Figure 6. Following an arc in this graph corresponds to moving to an entry of opposite sign in the same row, and moving across a node to a next arc corresponds to moving to the one other row containing a non-zero at a specific position. Therefore there is a direct correspondence between the generating vectors of (5) and the *elementary circuits* (cyclic simple directed arc paths) of D [Johnson 1975]. To be precise, there are two oppositely oriented circuits corresponding to one generating vector. This is due to the existence of two D -nodes per T -edge with oppositely oriented incident arcs – which is necessary because in the above algorithm a row of A can likewise be used in two different “directions” (from 1 to -1 or from -1 to 1).

Note that the process is non-deterministic: in each row of A there can be multiple entries of opposite sign and one can be chosen arbitrarily. In the graph D this corresponds to the choice of one of the outgoing arcs of a node. This raises a question: is there always at least one choice that leads back to row j , terminating the procedure? Or put in terms of D : is there a circuit through each node? This can be answered positively. It is a direct consequence of the positive solution existence proof (cf. A.2).

Note that due to the non-determinism, the number of elementary circuits through a T -edge (D -node) e can be exponential in n . We use dynamic programming to select and construct a *good* one of these in quasilinear time. To this end we assign weights to the edges of T (the nodes of D) and find a cycle with minimum total weight. Note that this amounts to a simple Dijkstra-type algorithm.

Weights The weights w should reflect how much we would like an edge of T to keep its current x -value. This depends on the edge’s *ideal* length x_{ideal} . This length we read from the initial continuous parametrization (the minimizer of \mathbf{E}) computed for the motorcycle graph construction. We generally use positive weights so as to favor small generating vectors v which perform only little overall change. We use a three-tier weighting scheme, distinguishing between edges which are currently shorter than their ideal value, edges which are about ideal, and edges which are currently larger than their ideal value. Furthermore we use a different set of weights depending on whether we intend to add (w_+) or subtract (w_-) the vector v . In the case of addition, let $d_+(e) = x_{\text{ideal}}(e) - x(e)$, in the case of subtraction $d_-(e) = x(e) - x_{\text{ideal}}(e)$. Then define:

$$w_{\pm}(e) = \begin{cases} 1/(d_{\pm}(e) + 1) & \text{if } 1 \leq d_{\pm}(e) \\ \eta/(d_{\pm}(e) + 1) & \text{if } 0 \leq d_{\pm}(e) < 1 \\ \eta^2(1 - d_{\pm}(e)) & \text{if } d_{\pm}(e) < 0 \end{cases}$$

where η is the number of edges of T , i.e. we give strict preference to edges which even after the addition of v remain shorter than ideal (case 1, $w \in (0, \frac{1}{2}]$) over edges which become longer than ideal (case 2, $w \in (\frac{1}{2}, 1]\eta$) over edges which already are longer than ideal (case 3, $w \in (1, \infty)\eta^2$). We override $w_-(e) = \infty$ if $x(e) = 0$. The choice of η accomplishes that edges of a higher tier are only considered if no cycle in the lower tier edges exists.

Result The result is a vector v whose entries are from $\{0, 1, 2\}$ – in terms of D : a circuit which crosses each edge of T zero, one, or two times. An entry of 2 can arise when both (oppositely directed) nodes on one T -edge are used by the circuit. Assume a node would have been used multiple times; then the circuit would not be simple, i.e. be decomposable into multiple circuits – with a lower weight each, so the circuit is not weight-optimal, i.e. not constructed by our algorithm. Hence, entries > 2 do not occur (and even 2 is rare).

6.3 Greedy Edge Selection

To successfully perform stage I of our algorithm, we can proceed as follows: as long as there is an edge e with $x(e) = 0$, construct and add a generating vector v with $v[e] = 1$ (or 2). This trivially leads to a valid quantization, as all differential parameters x are positive upon termination. We process edges in order of increasing weight, i.e. in each step we choose $e = \operatorname{argmin}_{e|x(e)=0} w_+(e)$. Note that one could stop the first stage as soon as the quantization is valid, i.e. not all x need to become positive. The additionally required validity check, however, makes this option slightly slower on average.

For stage II we determine the edges $e_+ = \operatorname{argmin}_e w_+(e)$ and $e_- = \operatorname{argmin}_e w_-(e)$. If $w_+(e_+) < w_-(e_-)$ we determine a generating vector v_+ with $v_+[e_+] \neq 0$, otherwise v_- with $v_-[e_-] \neq 0$. We then tentatively add/subtract this vector and evaluate whether this operation (A) preserves validity and (B) improves the overall situation. (A) is decided using the algorithm described in Section 5.2. (B) is measured as the sum of squared relative differences between actual value and ideal value over all edges of T : $\sum_e (x(e)/x_{\text{ideal}}(e) - 1)^2$. If the assessment is positive, the addition/subtraction is performed. Otherwise we try the edge with the second smallest weight, and so forth, until an improvement can be made. Afterwards, again the edge with the smallest weight is determined and this process is iterated. We terminate when no edge’s generating vector can provide any improvement anymore. Figure 5 illustrates the outcome of stage I and stage II on an example model.

7 Parametrization

For the initial seamless parametrization (3) we make use of a common field-guided objective [Bommes et al. 2009]. We add regularity constraints [Bommes et al. 2013b], thus a solver supporting (linear) inequality constraints is required. We use IPOPT for our examples. Note that in our setting there are no integer variables, i.e. the rather complex mixed integer techniques presented in these papers are not needed – only the basic objective and the constraints.

Once a valid complete quantization σ is determined, for the final parametrization we strive to optimize (3) subject to σ . We do this by adding linear equality constraints to the above optimization problem, imposing the quantization:

For a path p in T between two vertices i and j , the differential parameters can be collected to obtain the desired distance (x, y) between i and j . A constraint $(u_i, v_i) = \tau(u_j, v_j) + (x, y)$ is then formulated, where τ is the accumulated transition (with variable translation) collected along the path (from all mesh edges crossed by p). In order to set up a minimal, i.e. linearly independent, set of constraints, we compute a spanning tree of the singularities in T and a system of loops of T [Colin de Verdière and Lazarus 2005]. This yields a set of paths for which the above equality constraints are set up. This gives a constraint system $C(\{(u, v)\}, \{(s, t)\}) = b$, or short: $Cz = b$ (collecting all u, v, s, t variables in the vector z).

Solving (3) subject to these constraints then yields the regular quantized parametrization – allowing, for instance, for the extraction of a consistent quad mesh (cf. Figure 5, right) [Ebke et al. 2013]. Optionally, postprocessing steps, e.g. for the geometric improvement of the irregular vertices' positions, either based on the final parametrization [Campen and Kobbelt 2014b] or on the extracted quad mesh [Zhang et al. 2009], could be applied afterwards.

7.1 Robustness

The quantizations obtained by our approach appropriately separate singularities in parameter space by construction. While in general this property might not be sufficient for validity, specifically for those quantizations we construct based on the motorcycle graph, feasibility (possibly subject to input mesh refinement) is easily shown constructively [Myles et al. 2014].

In practice, the triangle mesh discretization and the convexification of parametrization regularity constraints [Bommes et al. 2013b; Lipman 2012] however, can still sometimes lead to infeasibility. These constraints are conservatively made convex based on a reference pose, and exclude not only irregular parametrizations, but any parametrization deviating too much from the reference.

Improving upon [Bommes et al. 2013b], where an a priori fixed regularity constraint linearization is employed, we found it very helpful to perform a *dynamic re-linearization* for the final parametrization. To this end, initially we do not impose $Cz = b$ as a hard constraint, but add the energy term $\lambda(Cz - b)^2$ (with a large value λ) to the minimization objective, thereby not affecting feasibility.

Conceptually, this does not fix the singularities to specific (potentially infeasible) parametric positions, but pulls them there as close as possible while obeying the (linearized) regularity constraints. If for the solution z' we find $\|Cz' - b\|_\infty > \varepsilon$, we re-linearize the constraints as described by Bommes et al. [2013b] (based on the new reference pose provided by z'), and re-solve. This is iterated until $\|Cz' - b\|_\infty \leq \varepsilon$ (which is not strictly guaranteed to occur), and then a final solve with constraints $Cz = b$ is performed. With $\lambda = 10^6$ and $\varepsilon = 10^{-4}$ this succeeded even for highly distorted results like the intermediate states of stage I shown in Figure 5.

7.1.1 Guarantees

While the above re-linearization strategy proved to perform well in our experiments, we note that there is no guarantee that this dynamic re-linearization always leads to a feasible parametrization problem – in theory, the initial, unquantized parametrization problem could already be infeasible.

Very recently a method for continuous seamless global parametrization with success guarantee was presented [Myles et al. 2014]. As our method is oblivious to and independent of the source of the initial parametrization, this method can of course be used to obtain the initial parametrization for the T-mesh construction in Section 5.1.

To obtain strict guarantees for the final parametrization step, the topological partition simplification described by Myles et al. [2014] can be employed to remove zero-cells from the T-mesh resulting from our algorithm in Section 6. Afterwards each cell can individually be parameterized regularly, in the simplest case using Tutte's barycentric mapping [Tutte 1963], instead of following Section 7.

8 Results

We applied our quantization strategy (quantized global parametrization, QGP) to continuous seamless parametrizations computed using part of the pipeline described by Bommes et al. [2009] (cross field construction followed by continuous parametrization, but no integer rounding) with tri-sector based regularity constraints [Bommes et al. 2013b]. In order to evaluate its performance, in terms of run time and result quality, we compare to the one previous quantization approach which is robust, i.e. IGM [Bommes et al. 2013b].

IGM makes use of a generic branch-and-bound based mixed integer solver like CPLEX or Gurobi (we use the latter). Such solvers allow for the specification of a time limit for optimization, after which the best solution found so far (if any) is returned. Making use of this possibility in the context at hand is mandatory, because waiting for the optimum to be found and confirmed is often infeasible (hours, days, ...). Note that such time limits can be imposed on our method, too, because in stage II we always have a valid quantization to be returned at any time.

In Figure 7 we demonstrate what quality can be expected from both algorithms after what run time. We only consider the quantization phase for comparison here, because initial steps like continuous parametrization, and final steps like constrained parametrization, mesh extraction, mesh optimization, etc., do not differ between both methods (and depend on mesh complexity rather than singularity count). In the case of IGM it can take quite long until a first valid solution is found; we indicate the time at which a solution is found using a “[” in this figure. Times at which an algorithm finally terminated are indicated by “]”.

Generally, valid results are obtained very quickly with QGP, even for complex models with a large number n of singularities. The result quality then quickly improves with increasing run time. Obtaining similar results with IGM requires allotting a much longer run time. For complex examples with many integer degrees of freedom even after several hours no result, which we could compare to, can be delivered by IGM.

With this increased performance, the time needed for initial and final parametrization now dominates the overall run time, putting a bound on the maximum performance improvement that can be achieved by our method when considering the big picture. It thus seems desirable that future research be focused on finding more efficient approaches to these constrained parametrization problems.

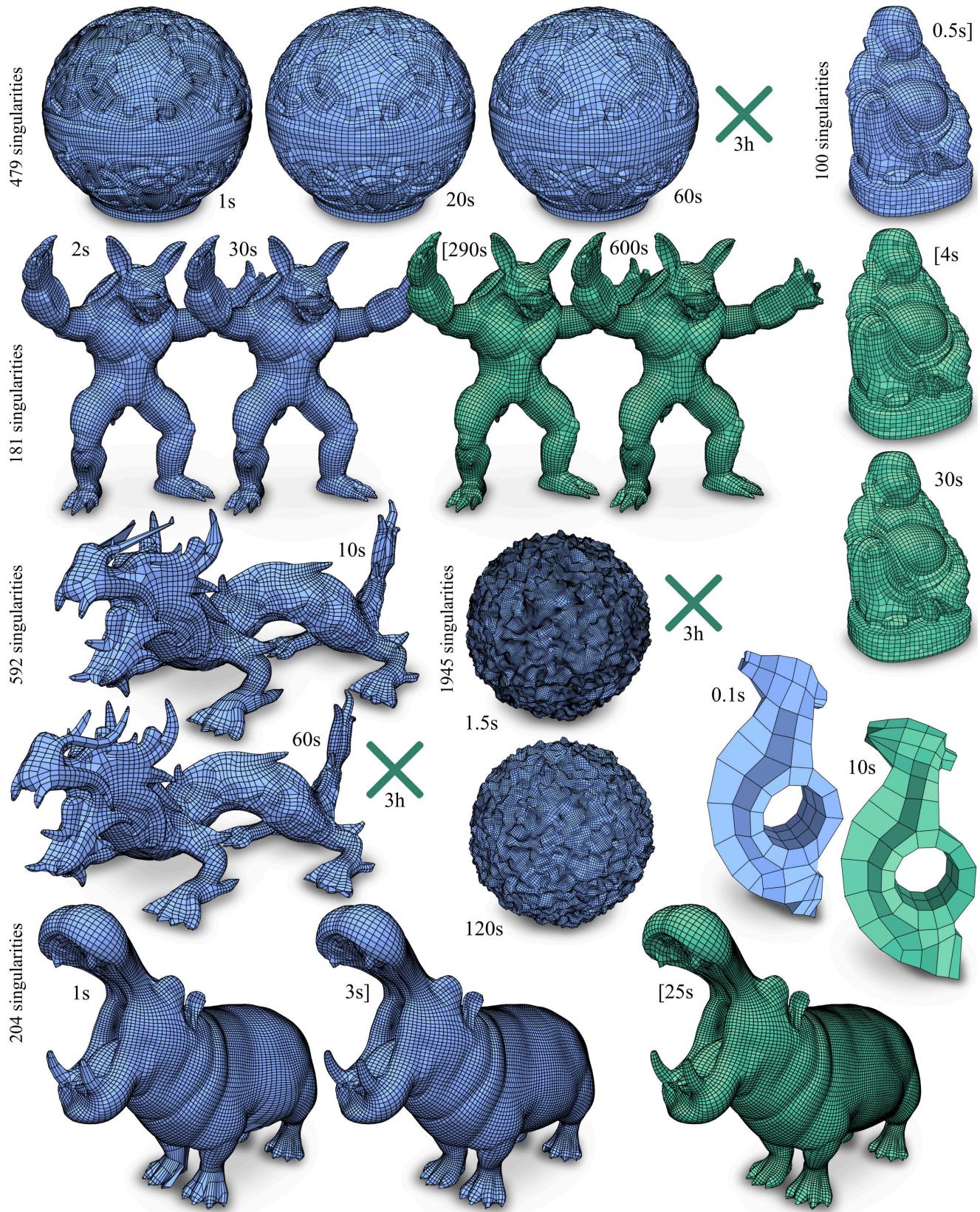


Figure 7: Quad meshes extracted from integral parametrizations quantized by our QGP (blue) and IGM (green) [Bommes et al. 2013b]. Run times are specified next to the models. “[” indicates a time where the first valid result is found. “]” indicates a time where the algorithm terminates, delivering the (greedy) optimum. A green cross means IGM could not produce any result even after 3 hours of run time.

9 Generalization

So far we considered smooth surfaces without boundary. In the following we extend the described algorithms to handle surfaces with boundary, and piecewise smooth surfaces with feature curves – both cases of practical relevance. Most concepts extend quite easily, but some details must be taken care of.

9.1 Feature Curves

If the surface contains prominent feature curves, it is usually expected from a quantized parametrization to align integer isolines to these curves. To this end one can add constraints to the initial and final parametrization problem, forcing all the vertices along a curve to have a common u - or v -parameter (subject to transitions) [Bommes et al. 2009]. We need to make sure that the quantization does not contradict these constraints by forcing two singularities on a common feature curve to have different u - and v -parameters. To this end one lets the motorcycles first trace out the feature curves. If there is a feature curve not containing any singularity, virtual (valence 4) singularities are added on the end points to emit motorcycles. Only after this the remaining motorcycles are traced. In this way all feature curves are present in T as edges, avoiding any contradictions.

Note that the fact that singularities (or virtual singularities) lie on the feature curve already implies that its corresponding isoline will be integral. What, however, needs to be taken care of additionally, is that a feature edge and a singularity not on the feature edge must not collapse in parameter space – analogous to two singularities that must not have zero distance. This is easily integrated into the validity check (cf. Section 5.2): invalidity is not only reported when another singularity node is reached with zero distance, but also when any point on a feature edge is reached with zero distance. Figure 8 shows an example model with numerous feature curves handled in this way.

9.2 Boundaries

When the surface has boundaries, they can be taken into account as feature curves as described above. In this way parametrization and motorcycle graph are aligned with the boundary. Then, the only modification that needs to be made to our quantization algorithm is due to the following difference: in addition to generating vectors which correspond to cycles in the graph D , there will be generating vectors which correspond to paths in D which run from a node without any incoming arc to a node without any outgoing arc (i.e. boundary nodes, on T -edges along the boundary, which are adjacent to only one cell). These can easily be included in our algorithm described in Section 6.2 by performing a *bidirectional* Dijkstra-type algorithm (one search with, one against the arc directions) until both fronts hit each other (forming a cycle) or hit boundary nodes (forming a path). Figure 8 shows example quad meshes of models with boundary generated in this manner.

In the (practically less relevant) problem setting without alignment to the boundaries, additional precaution would have to be taken: while on a surface without boundary, or with aligned boundary, the motorcycle graph is guaranteed to be connected (and, because each cell is of disc topology, the cell decomposition that the graph describes even homotopy equivalent to the underlying surface [Eppstein et al. 2008]), this does not generally hold when unaligned boundaries are present. Additional isoline segments may then be added as edges to T so as to remedy this. We leave to future work determining the simplest or most efficient way of selecting these additional edges for use cases that require unaligned parametrizations.

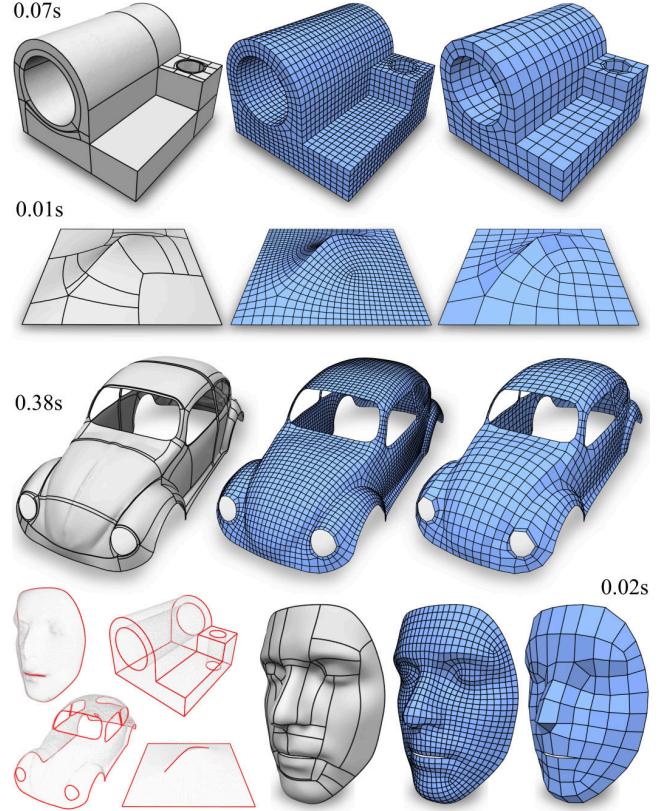


Figure 8: Quad meshes from integral parametrizations with alignment to feature curves and boundaries (red). By construction the depicted motorcycle graphs (black) include all feature and boundary curves. Notice how, when aiming for coarser quantizations, the quads remain small where necessary to prevent interior singularities and features or boundaries from collapsing parametrically.

10 Discussion & Future Work

As laid out in Section 5, the use of T-mesh based differential parameters is the key to our algorithm’s efficiency. This choice of oriented graph structure, just like the oriented spacer constraints of Bommes et al. [2013b], excludes certain quantizations – intuitively those where close singularities change their relative order in parameter space compared to the non-quantized parametrization. Note that, while the spacer constraints already predetermine in which direction (u or v) close singularities are separated in parameter space, in our setting this is still open, providing additional flexibility.

We use a greedy strategy in order to yield a polynomial time approximation algorithm for an NP-hard problem. This choice was made because the exponential worst case complexity of the robust previous approach [Bommes et al. 2013b] can be a significant obstacle in practical applications. As demonstrated in Section 8 the performance of the algorithm in terms of run time and result quality provides clear practical advantages. Nevertheless, it could be of interest to investigate ways to explore the space of generating vectors and quantizations in a non-greedy fashion in future work, perhaps even to provably optimal results.

In a somewhat different problem setting, actually not making sure that singularities are separated in parameter space, but merging them on the surface when the quantization forces them to collapse in parameter space, can be a strategy of interest. A number of non-trivial issues would have to be addressed for this. For instance, certain singularities cannot technically be merged, because they would

result in a singularity of an index too low or too high to be representable in a linear parametrization or to be valid for meshing purposes. Furthermore, depending on the application no or not every change to the singularity layout may be admissible.

10.1 Interval Assignment

The subproblem of assigning non-negative integers to the edges of a T-mesh in a consistent manner is related to the so-called structured interval assignment problem, a classical problem in mesh generation [Tam and Armstrong 1993; Mitchell 2000]. There is a fundamental difference, though: in this interval assignment problem positive intervals are required, while we need non-negative integers (subject to validity constraints), and this possibility of zero-edges is essential for the result quality (e.g. in Figure 5 left, in the final result after stage II 99 of the 432 edges of the underlying T-mesh are zero-edges, while the intermediate result of stage I has no zero-edges). Approaches to positive interval assignment described in the literature are thus not applicable to our problem.

Applying our quantization algorithm from Section 6 to the interval assignment problem, however, is possible with a simple change: forbid the subtraction of a generating vector if it would decrease the parametric length of any edge below 1 (and setting $w_-(e) = \infty$ if $x(e) = 1$). We make no claim about the relative performance for this use case here though, as it is out of the scope of this paper. The adapted algorithm could, however, be a relatively lightweight robust alternative to non-robust algorithms (not guaranteeing positivity) that are sometimes applied in the Computer Graphics field in this context [Tong et al. 2006; Bommes et al. 2008].

Finally, we note that a quad layout is a special case of a T-mesh (no T-junctions). In this case, any non-determinism vanishes from the generating vector construction. Each edge is affected by only one generating vector, i.e. these vectors (corresponding to the quad layout's dual cycles) are orthogonal. Hence, coefficients are effectively chosen individually per cycle of quads, no interdependencies need to be considered, and processing order does not matter.

11 Conclusion

We analyzed the complications involved in transitioning from seamless to integral global parametrizations, which are essential, for instance, in the field of mesh generation. This process of quantization comes with a number of combinatorial and topological issues which need to be addressed for robustness reasons. We showed that by formulating the problem in alternative degrees of freedom based on a convex decomposition of the parameter space, it is possible to derive a relatively simple algorithm which performs the quantization efficiently. It is based on navigating in the solution space of a system of consistency conditions while strictly avoiding configurations leading to infeasibility of the quantized parametrization problem. The quantizations generated are thus guaranteed to induce correct integral parametrizations, and we demonstrated the advantages over previous approaches.

Acknowledgments

We thank Max Lyon and Moritz Ibing for their help with the implementation and evaluation. This research was funded by the European Research Council (ERC Advanced Grant "ACROSS", grant agreement 340884) and the German Research Foundation (DFG grant GSC 111 (Aachen Institute for Advanced Study in Computational Engineering Science) and Gottfried-Wilhelm-Leibniz Programm). Several of the models are courtesy of the AIM@SHAPE Repository and the Stanford 3D Scanning Repository.

References

- BOMMES, D., VOSSEMER, T., AND KOBBELT, L. 2008. Quadrangular parameterization for reverse engineering. *Mathematical Methods for Curves and Surfaces*, 55–69.
- BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. In *Proc. SIGGRAPH 2009*, 1–10.
- BOMMES, D., LÉVY, B., PIETRONI, N., PUPPO, E., SILVA, C., TARINI, M., AND ZORIN, D. 2013. Quad-mesh generation and processing: A survey. *Computer Graphics Forum* 32, 6, 51–76.
- BOMMES, D., CAMPEN, M., EBKE, H.-C., ALLIEZ, P., AND KOBBELT, L. 2013. Integer-grid maps for reliable quad meshing. In *Proc. SIGGRAPH 2013*, 98:1–98:12.
- CAMPEN, M., AND KOBBELT, L. 2014. Dual strip weaving: Interactive design of quad layouts using elastica strips. In *Proc. SIGGRAPH Asia 2014*, 183:1–183:10.
- CAMPEN, M., AND KOBBELT, L. 2014. Quad layout embedding via aligned parameterization. *Computer Graphics Forum* 33, 8, 69–81.
- CAMPEN, M., BOMMES, D., AND KOBBELT, L. 2012. Dual loops meshing: quality quad layouts on manifolds. In *Proc. SIGGRAPH 2012*, 110:1–110:11.
- COLIN DE VERDIÈRE, E., AND LAZARUS, F. 2005. Optimal system of loops on an orientable surface. *Discrete & Computational Geometry* 33, 3, 507–534.
- DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. C. 2006. Spectral surface quadrangulation. In *Proc. SIGGRAPH 2006*, 1057–1066.
- DUMITRESCU, A., ROTE, G., AND TTH, C. 2013. Monotone paths in planar convex subdivisions and polytopes. In *Discrete Geometry and Optimization*, vol. 69 of *Fields Institute Communications*. Springer, 79–104.
- EBKE, H.-C., BOMMES, D., CAMPEN, M., AND KOBBELT, L. 2013. Qex: Robust quad mesh extraction. In *Proc. SIGGRAPH Asia 2013*, 168:1–168:10.
- EBKE, H.-C., CAMPEN, M., BOMMES, D., AND KOBBELT, L. 2014. Level-of-detail quad meshing. In *Proc. SIGGRAPH Asia 2014*, 184:1–184:11.
- ECK, M., AND HOPPE, H. 1996. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *Proc. SIGGRAPH 96*, 325–334.
- EPPSTEIN, D., GOODRICH, M. T., KIM, E., AND TAMSTORF, R. 2008. Motorcycle Graphs: Canonical Quad Mesh Partitioning. *Computer Graphics Forum* 27, 5, 1477–1486.
- FLOATER, M. S., AND HORMANN, K. 2005. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*. Springer, 157–186.
- GU, X., GORTLER, S. J., AND HOPPE, H. 2002. Geometry images. In *Proc. SIGGRAPH '02*, 355–361.
- GUSKOV, I., VIDIMCE, K., SWELDENS, W., AND SCHRÖDER, P. 2000. Normal meshes. In *Proc. SIGGRAPH 2000*, 95–102.
- JIANG, T., FANG, X., HUANG, J., BAO, H., TONG, Y., AND DESBRUN, M. 2015. Frame field generation through metric customization. *ACM Trans. Graph.* 34, 4.
- JOHNSON, D. B. 1975. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.* 4, 1, 77–84.

- JUCOVIČ, E., AND TRENKLER, M. 1973. A theorem on the structure of cell-decompositions of orientable 2-manifolds. *Matematika* 20, 63–82.
- KÄLBERER, F., NIESER, M., AND POLTHIER, K. 2007. Quad-cover - surface parameterization using branched coverings. *Computer Graphics Forum* 26, 3, 375–384.
- KHODAKOVSKY, A., LITKE, N., AND SCHRÖDER, P. 2003. Globally smooth parameterizations with low distortion. *ACM Trans. Graph.* 22, 3, 350–357.
- LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.* 21, 3, 362–371.
- LI, E., LÉVY, B., ZHANG, X., CHE, W., DONG, W., AND PAUL, J.-C. 2011. Meshless quadrangulation by global parameterization. *Computers & Graphics*, 992–1000.
- LIPMAN, Y. 2012. Bounded distortion mapping spaces for triangular meshes. In *Proc. SIGGRAPH 2012*, 108:1–108:13.
- LIU, Y., XU, W., WANG, J., ZHU, L., GUO, B., CHEN, F., AND WANG, G. 2011. General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.* 30, 6, 140:1–140:10.
- MITCHELL, S. A. 2000. High fidelity interval assignment. *Int. J. Comput. Geometry Appl.* 10, 4, 399–415.
- MYLES, A., AND ZORIN, D. 2012. Global parametrization by incremental flattening. In *Proc. SIGGRAPH 2012*, 109:1–109:11.
- MYLES, A., AND ZORIN, D. 2013. Controlled-distortion constrained global parametrization. *ACM Trans. Graph.* 32, 4, 105:1–105:14.
- MYLES, A., PIETRONI, N., KOVACS, D., AND ZORIN, D. 2010. Feature-aligned T-meshes. In *Proc. SIGGRAPH 2010*, 117:1–117:11.
- MYLES, A., PIETRONI, N., AND ZORIN, D. 2014. Robust field-aligned global parametrization. In *Proc. SIGGRAPH 2014*, 135:1–135:14.
- NOŽIČKA, F., GUDDAT, J., HOLLATZ, H., AND BANK, B. 1974. *Theorie der linearen Parametrischen Optimierung*. Akademie-Verlag, Berlin.
- PANOZZO, D., PUPPO, E., TARINI, M., AND SORKINE-HORNUNG, O. 2014. Frame fields: Anisotropic and non-orthogonal cross fields. *ACM Trans. Graph.* 33, 4.
- PIETRONI, N., TARINI, M., AND CIGNONI, P. 2010. Almost isometric mesh parameterization through abstract domains. *IEEE Trans. Vis. Comput. Graph.* 16, 4, 621–635.
- PIETRONI, N., TARINI, M., SORKINE, O., AND ZORIN, D. 2011. Global parametrization of range image sets. *ACM Trans. Graph.* 30, 6, 149:1–149:10.
- PIETRONI, N., TONELLI, D., PUPPO, E., FROLI, M., SCOPIGNO, R., AND CIGNONI, P. 2015. Statics aware grid shells. *Computer Graphics Forum* 34, 2, 627–641.
- PURNOMO, B., COHEN, J. D., AND KUMAR, S. 2004. Seamless texture atlases. In *Proc. Symp. Geom. Proc.*, 65–74.
- RAY, N., LI, W. C., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph.* 25, 1460–1485.
- RAY, N., NIVOLIERS, V., LEFEBVRE, S., AND LÉVY, B. 2010. Invisible seams. *Comput. Graph. Forum* 29, 4, 1489–1496.
- ROCKAFELLAR, R. T. 1970. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J.
- SCHILLING, C. H., LETSCHER, D., AND PALSSON, B. O. 2000. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *J. Theor. Biol.* 203, 3, 229–248.
- SORKINE, O., COHEN-OR, D., GOLDENTHAL, R., AND LISCHINSKI, D. 2002. Bounded-distortion piecewise mesh parameterization. In *IEEE Visualization*, 355–362.
- TAM, T. K. H., AND ARMSTRONG, C. G. 1993. Finite element mesh control by integer programming. *International Journal for Numerical Methods in Engineering* 36, 15, 2581–2605.
- TARINI, M., PUPPO, E., PANOZZO, D., PIETRONI, N., AND CIGNONI, P. 2011. Simple quad domains for field aligned mesh parametrization. *Proc. SIGGRAPH Asia 2011*, 142:1–142:12.
- TOMÁS, A. P., AND FILGUEIRAS, M. 1997. An algorithm for solving systems of linear diophantine equations in naturals. In *Progress in Artificial Intelligence*, vol. 1323 of *Lecture Notes in Computer Science*. 73–84.
- TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *Proc. SGP '06*, 201–210.
- TUTTE, W. T. 1963. How to draw a graph. *Proc Lond Math Soc* 13, 743–767.
- WEBER, O., AND ZORIN, D. 2014. Locally injective parametrization with arbitrary fixed boundaries. *ACM Trans. Graph.* 33, 4, 75:1–75:12.
- ZHANG, Y., BAJAJ, C., AND XU, G. 2009. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. *Comm. Num. Meth. in Eng.* 25, 1, 1–18.
- ZHANG, M., HUANG, J., LIU, X., AND BAO, H. 2010. A wave-based anisotropic quadrangulation method. In *Proc. SIGGRAPH 2010*, 118:1–118:8.

A Appendix

A.1 Non-Trivial Solution

As every node of graph D (cf. Section 6.2) has an outgoing arc, D contains an infinite path. As the set of arcs is finite it must be cyclic, i.e. there is at least one circuit in D , thus a non-trivial generating vector, providing a non-trivial solution to (5). \square

A.2 Positive Solution

Let L be the set of the parametric lengths $l_i > 0$ of all graph edges, measured in the underlying parametrization. Assume the parametrization is represented by rational numbers, i.e. $(u, v) \in \mathbb{Q} \times \mathbb{Q}$ (which holds trivially in any numerical implementation). Then every separatrix has a rational parametric position, and, as all edges are isolines (i.e. axis-aligned in parameter space) and start and end at separatrices, l_i is simply the difference of rational numbers. Therefore the lowest common denominator q of L exists and scaling the parametrization by q results in a set of positive integral edge lengths, providing a positive solution to (5). \square

Note that this specific solution is mainly of theoretical interest – the edge lengths can be very large integers.