Computers & Graphics (2023)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Towards a Robust and Portable Pipeline for Quad Meshing: Topological Initialization of Injective Integer Grid Maps

Marco Livesu[a,*]

[a] *CNR IMATI, Genoa, Italy*

ARTICLE INFO

Article history:
Received March 17, 2023

Keywords: surface mapping, mesh generation, quadmeshing

ABSTRACT

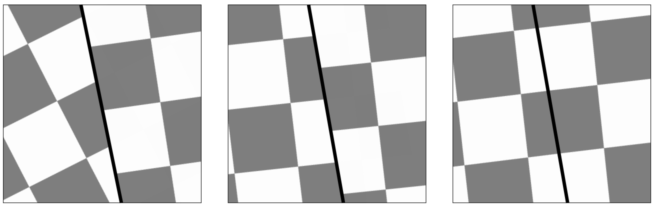
Integer Grid Maps (IGMs) are a class of mappings characterized by integer isolines that align up to unit translations and rotations of multiples of 90 degrees. They are widely used in the context of remeshing, to lay a quadrilateral grid onto the mapped surface. The presence of both discrete and continuous degrees of freedom makes the computation of IGMs extremely challenging. In particular, solving for all degrees of freedom altogether leads to a mixed-integer problem that is known to be NP-Hard. Such a problem can only be solved heuristically, occasionally failing to produce a valid quadrilateral mesh. In this paper we propose a simple topological construction that allows to reduce the problem of computing a valid IGM to the one of mapping a topological disk to a convex domain. This is a much easier problem to deal with, because it completely removes the integer constraints, permitting to obtain a provably injective parameterization that is guaranteed to incorporate all the correct integer transitions with a simple linear solve. Not only the proposed algorithm is easy to implement, but it is also independent from costly numerical solvers that are unavoidable in existing quadmeshing pipelines, preventing their exploitation in open source or low-budget projects. Despite provably correct, the so generated maps contain a considerable amount of geometric distortion and a poor quad connectivity, making this technique more suitable for a robust initialization rather than for the computation of an application-ready IGM. In the article we present the details of our construction, also analyzing its geometric and topological properties.

1. Introduction

The generation of quadrilateral meshes is an important task in geometry processing and is widely exploited in many applicative fields, spanning from animation to Computer-Aided Design, architecture and many others [1]. Since the introduction of seminal works such as [2, 3] methods that employ a parameterization to generate a quadrilateral mesh have established themselves as a standard the facto and are able to produce high quality tessellations that endow a sparse singular structure, also aligning to both principal curvatures and sharp features. The class of mappings that allow to lay a quadrilateral grid onto a discrete surface are called Integer Grid Maps (IGM) [4].

The distinctive trait of an IGM is that it ensures continuity of the integer level sets of the underlying map across cuts, obtained by explicitly imposing translational and rotational alignment constraints by multiples of 90 degrees (Figure 1, right). When such alignment constraints are observed and singularities arise at integer locations, tracing the integer level sets of the mapping yields a pure quadrilateral mesh. The presence of explicit constraints makes the computation of the IGM an

*Corresponding author:
e-mail: marco.livesu@gmail.com (Marco Livesu)



regular seamless IGM

Fig. 1. Left: regular mappings impose no continuity across cuts. Middle: seamless maps ensure that parametric isolines rotationally align up to integer multiples of 90°, but the grids at opposite sides of the cut may still expose translational misalignment. Right: Integer Grid Maps ensure both rotational and translational alignment, so that contouring the integer isolines of the mapping yields a pure quad tessellation that is completely oblivious of the underlying mapping cuts.

extremely challenging mixed-integer problem that is known to be NP-Hard [5], resulting in brittle pipelines that do not offer guarantees of correctness and may unexpectedly fail to produce a valid result. Solving for rotational and translational alignment separately allows to guarantee the validity of the IGM [6], but this approach still relies on expensive commercial solvers for which an equivalent free counterpart does not really exist, preventing open source or low-budget projects to implement a fully robust quadmeshing pipeline (Section 2).

This work aims to provide a first ingredient towards the realization of a provably robust pipeline for the generation of quadrilateral meshes through the computation of IGMs. In addition to validity requirements, we take into consideration also the simplicity of the approach and its external dependencies. Our ultimate goal is to design a quadmeshing pipeline that does not depend on costly numerical solvers, so that it can be easily installed within any commercial or open source tool.

A major source of inspiration comes from recent approaches for the robust computation of injective simplicial mappings, such as [7, 8, 9, 10, 11]. At a high level, these robust methods all follow a similar pipeline composed of two steps: (i) they generate an initial feasible solution that is guaranteed to be injective, albeit arbitrarily distorted; (ii) they carefully improve such initial solution reducing geometric distortion, making sure that injectivity is preserved throughout the process. The first step is obtained with Tutte or similar alternatives [12, 13]. The second step is typically implemented using barrier energies that grow to infinite when a triangle becomes nearly degenerate, ensuring that all mesh elements preserve their correct orientation, that was set to be globally coherent in the initialization step.

We wish to reproduce a similar pipeline for the generation of Integer Grid Maps, hence of quadrilateral meshes. In particular, in this paper we focus on the first step of the pipeline, introducing a robust topological method for the initialization of a valid IGM. Note that creating an IGM poses additional challenges that are not handled by existing methods for the computation of simplicial maps. In terms of validity, in addition to the injectivity requirement one must also ensure that map singularities arise at integer locations and that integer isolines are continuous across cuts. In terms of quality, in addition to the desire to minimize geometric distortion one may also want to produce a good topology, which typically means that the quadmesh contains few singular vertices of low valence, connected to one another so as to form a coarse quad layout [14, 1].

The topological initialization proposed in this paper takes care of all the validity requirements, ensuring that the mapping is injective and that the integer parametric isolines yield a pure quadrilateral mesh for shapes of any genus (Section 4). The fulfillment of the quality requirements is addressed at the second step of the pipeline, which is left to future works (Section 7).

This work extends the conference article [15], which was previously presented at STAG 2022, obtaining the Best Paper Award. This new version extends the previous article in various aspects, namely:

1. better clarifying the positioning of the proposed methodology w.r.t. the state of the art in the field (Sections 1 and 2);
2. showing that in some cases the originally proposed topological construction was not applicable, due to the gluing scheme not being in normal form (Figure 5));
3. providing a modification of the original algorithm that fixes this issue, ensuring applicability to any possible input manifold (Section 4.1);
4. providing a comprehensive analysis of the output mesh quality (Section 6);
5. incorporating additional considerations on extensions and future works (Section 7).

2. Related works

Quadrilateral meshing is a vast topic with a long list of techniques and applications. In the remainder of the section we will focus on the methods that are most relevant to our work. For a broader perspective, we point the reader to a comprehensive survey, such as [1].

Integer Grid Maps. Pioneering works in the field [2, 3, 16, 4, 17] have established surface mappings as a prominent method for the generation of quadrilateral meshes. Integer Grid Map methods compute a mesh by solving a complex mixed-integer problem that is known to be NP-Hard [5]. In practice, various heuristics are employed to make the problem tractable, exposing the output IGM to various imperfections. Attempts to remedy such defects are also heuristic. As an example, in the original MIQ [2] article the authors employ a greedy rounding strategy for the integer constraints and the so called *stiffening* (Sec. 5.4) to remove flipped elements, which consists in iteratively increasing the energy functional to locally penalize distortion. In a subsequent work [4], greedy rounding was substituted with a dedicated branch-and-bound solver and stiffening replaced by a set of linear anti-flip constraints that assign triangle vertices to three disjoint semi-infinite convex sub-spaces (Sec. 3.1 in [4]). However, the branch-and-bound solver is still heuristic and is known to take even days of computation for complex models or to occasionally fail to find a valid solution [6]. Furthermore, the anti-flip contraints for the preservation of injectivity are just a linearization of the full (non linear) functional, hence they restrict the orientation of mapped triangles, possibly leading to unfeasible configuration spaces where all constraints cannot be globally satisfied altogether. Ebke and colleagues introduced QEx [18], a powerful tool that is guaranteed to extract a valid quad mesh if the underlying parameterization is locally injective, possibly not compliant with the singularities in the mapping. State of the art solvers do not guarantee that the mapping is injective, nonetheless QEx is often still able to extract a valid mesh, although no guarantees can be given in this regard.

If an injective seamless mapping is known, global quantization [6] can provide integer transitions that are guaranteed to fulfill the integer continuity requirements of an IGM. Furthermore, recent research has shown that robust high quality feature preserving quad meshes can also be computed by completely avoiding the construction of a map [19]. Both these methods internally solve global integer problems with a commercial tool (Gurobi) that performs remarkably better than open source counterparts, hence it cannot be practically replaced.

The topological approach described in this paper cannot compete with any of the aforementioned methods in terms of topological and geometric quality of the output, but it only requires a linear solve, it does not depend on any commercial solver and it is guaranteed to always produce an injective IGM from which a provably correct quadmesh can be trivially extracted. To this end, it can be seen as a powerful initialization for a robust IGM pipeline that can be included in any toolkit, both commercial and open source.

Seamless maps. Global seamless mappings can be seen as a continuous relaxation of IGMs, because they only ensure that parametric isolines align up to rotations of multiples of 90, but do not guarantee the continuity of integer isolines across cuts (Figure 1, middle). They can be used to fit tensor product higher order surfaces [20] or as an initialization step for the computation of an IGM through quantization [6, 21]. Literature in the field mostly deals with the generation of parametric spaces that conform to a prescribed set of cone singularities or holonomy signature [22, 23, 24, 25, 26, 27, 28, 29, 30]. Methods for seamless mapping that support manifolds of arbitrary genus partially overlap with this work, especially recent methods that use a combinatorial structure to robustly initialize the parameter domain, which have been a major source of inspiration for this work. Since an IGM is also a seamless map (but not vice-versa), our method can also be used to initialize such a mapping. However, the most modern methods are equally robust (i.e. they guarantee injectivity) and also contain less distortion and match prescribed cone singularities, therefore there are no practical reasons to prefer our tool in this setting.

Polygon Quadrangulation. Our work is also loosely connected with methods for the quadrangulation of simple polygons. Known closed form methods for polygon quadrangulation input a prescribed number of partitions for each polygon side and output a quadrilateral meshing of the interior that conforms to it [31, 32]. With proper tuning, the integer continuity constraints of IGMs can be translated into boundary conditions for these methods, obtaining a coarse quad tessellation that substitutes the templated coarse quadrilateral mesh that is laid over a simplicial mapping (Section 4.2). In our implementation we favored a templated solution because it scales flawlessly to surfaces of any genus and always provides a symmetric mesh topology, but recent methods such as [31] would equally provide valid solutions to our problem.

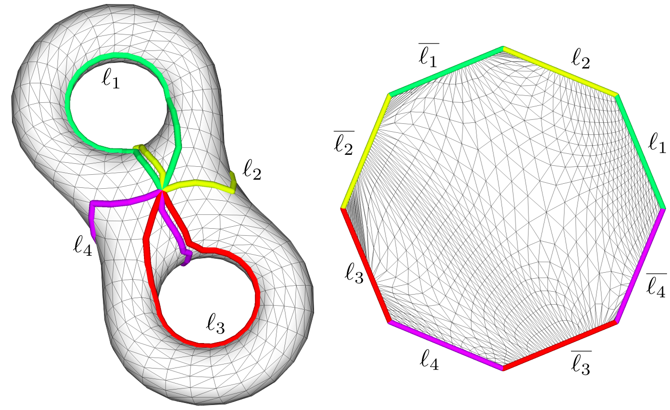


Fig. 2. The mapping Φ_{CPS} creates a one-to-one correspondence between a closed surface with genus g and a regular $4g$−gon. Cutting loops on the surface intersect at a mesh vertex and are fully disjoint elsewhere. The two images of each loop map to the edges of the polygon according to the gluing scheme $\ell_1, \ell_2, \overline{\ell_1}, \overline{\ell_2}, \ell_3, \ell_4, \overline{\ell_3}, \overline{\ell_4}$ (normal form). The mesh interior is mapped using Tutte [12].

3. Topological Background

In this section we briefly introduce a few notions from algebraic topology that are relevant for this work. Readers interested in more details about this topic can refer to [33] or similar books.

Polygonal Schema. Any closed surface mesh M can be cut open to form a topological disk and then flattened to the plane, obtaining a mapping. The union of cutting arcs and their meeting nodes forms the so called *cut graph*, which maps to a topological n−gon called the *polygonal schema* of M [34]. The genus of M sets a lower bound on the complexity of the polygonal schema. Specifically, it can be proven that for a surface with genus g there exists no polygonal schema with less than $4g$ sides, which is the minimal existing schema and is called *canonical* [35] (Figure 2).

Gluing Scheme. For surfaces with non trivial genus $g > 0$, each side of the polygonal schema takes the form of a loop that cuts open one of the shape handles along one direction (tangential or perpendicular). Specifically, if the schema is canonical the cut graph designs a system of $2g$ loops that all emanate from the same origin and are fully disjoint elsewhere. Note how the fact that cutting loops are fully disjoint ensures that the polygonal schema is minimal. In fact, if two loops were partially coincident, the polygonal schema would contain two additional vertices (two copies of the merging point between the two loops) and two additional sides. Efficient algorithms to compute a fully disjoint system of loops are discussed in [36]. Given a surface mesh M with genus g and a system of loops $L_M = \left\{\ell_1, \ell_2, \ldots, \ell_{2g}\right\}$, there exist multiple ways to map loop

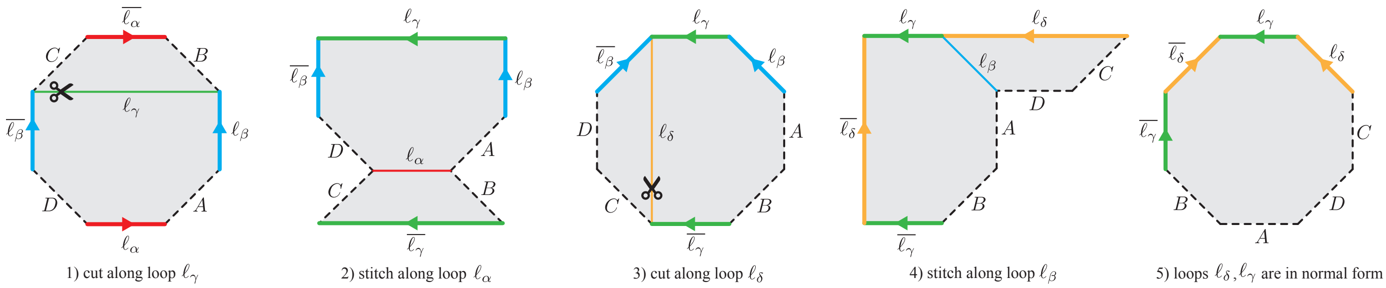


Fig. 3. Given a polygonal schema and an ordered sequence of non consecutive sides $\ell_\alpha, \ell_\beta, \overline{\ell_\alpha}, \overline{\ell_\beta}$ (here separated by four generic chains of sides A, B, C, D), it is always possible to substitute the string $\ell_\alpha, \ell_\beta, \overline{\ell_\alpha}, \overline{\ell_\beta}$ with four consecutive sides $\ell_\gamma, \ell_\delta, \overline{\ell_\gamma}, \overline{\ell_\delta}$ by applying a sequence of cut and stitch operations. Using the same construction for all similar patterns yields a polygonal schema in normal form.

images to the sides of the canonical schema. In particular, denoting with ℓ_i and $\overline{\ell_i}$ two images of the same loop $\ell_i \in L_M$, the gluing scheme

$$\ell_1, \ell_2, \overline{\ell_1}, \overline{\ell_2}, \ldots, \ell_{2g-1}, \ell_{2g}, \overline{\ell_{2g-1}}, \overline{\ell_{2g}}$$

is called the *normal form* (see [33], Chapter 17b).

Canonical polygonal schema are a topological invariant. So far this property has been exploited in computer graphics to compute cross-parameterizations between pairs of homotopic shapes [37, 38, 39]. In the next section we will show how a canonical polygonal schema with gluing scheme in normal form can be used to trivially design an Integer Grid Map.

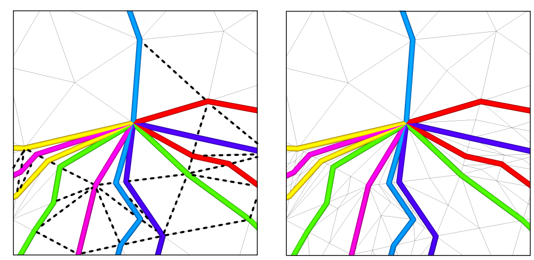


Fig. 4. Left: edges that directly connect vertices belonging to adjacent loops (thick dashed lines) create barriers that prevent the tracing of additional loops that travel towards the origin. Right: splitting all such edges always guarantees the existence of a valid path.

4. Method

Our method takes in input a closed mesh M with genus $g > 1$ and returns a valid quadrangulation of it. The algorithm is based on the composition of two mappings. In the first step we compute Φ_{CPS}, a mapping to the Canonical Polygonal Schema of M with gluing scheme in normal form. In the second step we compute Φ_{IGM}, a mapping from the polygonal schema to a regular grid, obtaining an Integer Grid Map. Both mappings are guaranteed to be injective. Overall, the relationship between the various embeddings and the mappings connecting them is as follows:

$$M \underset{\Phi_{CPS}}{\longleftrightarrow} CPS \underset{\Phi_{IGM}}{\longleftrightarrow} \text{regular grid}$$

In the remainder of the section we provide details about the construction of both Φ_{CPS} and Φ_{IGM}.

4.1. Computation of Φ_{CPS}

An injective mapping between a closed surface and a flat polygon can only be computed if the mesh is cut open to form a topological disk. As mentioned in Section 3, mappings to a canonical (or minimal) schema require the cut graph to be a system of $2g$ loops that meet at a single point and are fully disjoint elsewhere. We compute such a system as indicated in [36]. Specifically, we first initialize a cut graph using the *greedy homotopy basis* algorithm of Erickson and Whittlesey [40]. The loops produced by this method are not necessarily disjoint. We therefore separate portions of loops that travel along the same chains of mesh edges using the edge split strategy discussed in [36] (Sec. 4.1), obtaining a system of fully disjoint loops $L_M = \left\{\ell_0, \ell_1, \ldots, \ell_{2g}\right\}$. If loops in L_M are arranged in normal form, we cut the mesh along each loop and map the surface to a regular $4g-$gon using the Tutte embedding [12]. A visual illustration of Φ_{CPS} is shown in Figure 2. Due to the convexity of the polygonal schema, this mapping is guaranteed to be injective. In case L_M does not yield a gluing scheme in normal form, prior to cutting and mapping we proceed as indicated in the next paragraph.

Gluing Scheme. The homotopy basis algorithm [40] does not guarantee that loops are ordered so as to generate a polygonal schema in normal form. Luckily, given a polygonal schema with any gluing scheme, it is always possible to modify the ordering of its sides to enforce the normal form. A sequence of cut and stitch operations that produce the desired result were originally introduced by Fulton in [33] (Section 17b, page 238) and are shown in Figure 3. Despite mathematically correct this approach is overly complex from a geometry processing perspective, because it operates on the flattened schema (hence requires computing multiple mappings), yet because since we operate on an explicit mesh representation a considerable amount of mesh surgery is necessary to cut and weld together the various pieces of the polygon.
Here we propose an alternative solution that exploits the duality between edges in the polygonal schema and loops in the homotopy basis, obtaining an equivalent algorithm that operates directly on the input manifold without requiring any cutting or mapping. The main idea is that the cut and stitch operations

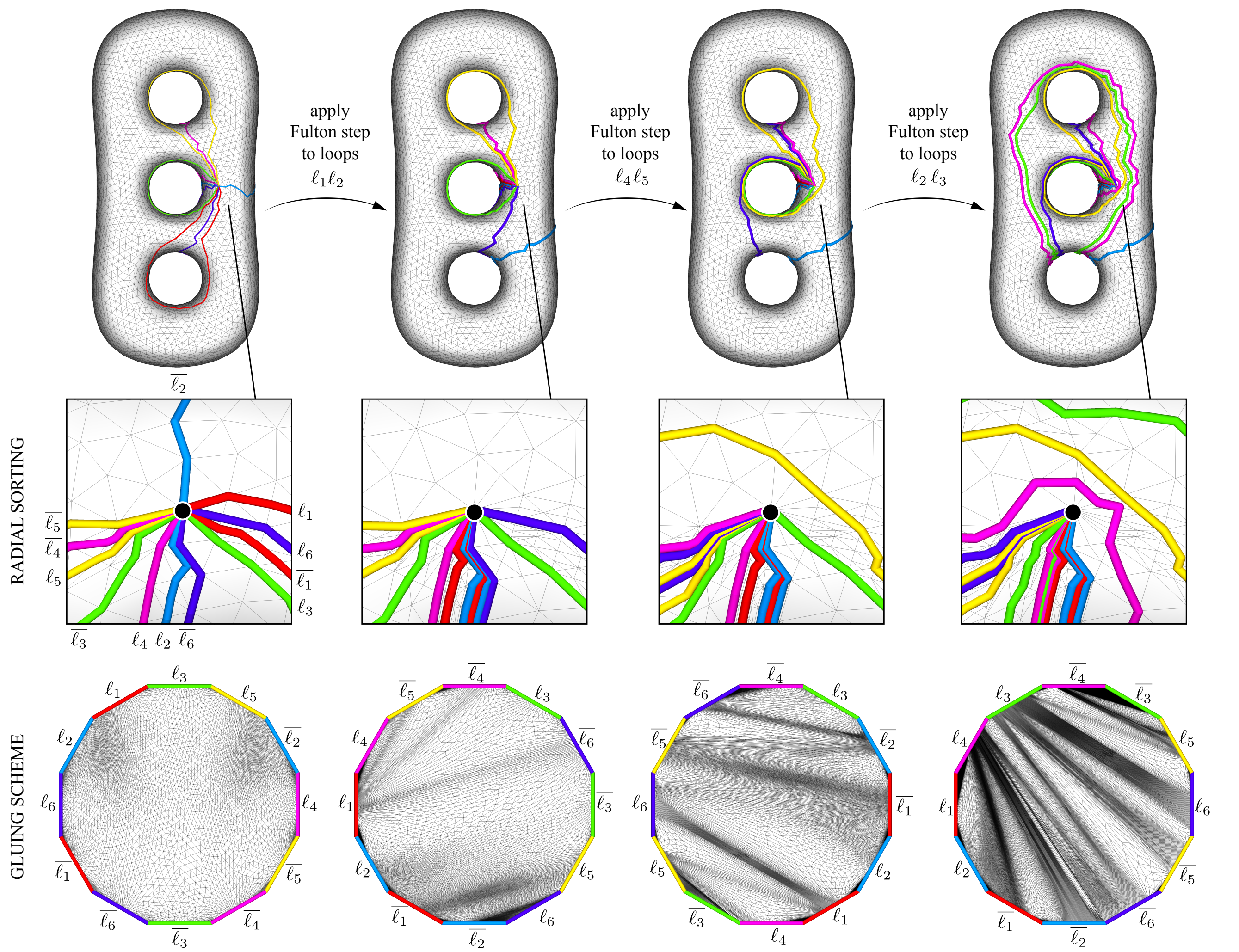


Fig. 5. Given a polygonal schema not in normal form (left column) we obtain a normal form (right column) by iteratively updating pairs of loops in the homotopy basis. Each update operation is equivalent to the Fulton scheme shown in Figure 3, but for efficiency it is executed directly on the input manifold. Closeups in the middle line show the radial loop sorting around the origin of the homotopy basis. The polygonal schemata shown in the bottom line are used to illustrate changes in the gluing scheme but their computation is not necessary to apply the algorithm.

shown in Figure 3 become much more efficient and easier to implement if executed in the input mesh. Specifically:

- cutting the polygonal schema along a curve is equivalent to tracing a new loop on the mesh, which can be done with a simple Dijkstra traversal constrained to visit only vertices that are not already participating in another loop. This avoids duplicating vertices and updating triangles in the data structure. Note that vertex constraints may prevent the existence of a solution. Splitting edges that directly connect vertices belonging to adjacent loops ensures that a solution can always be found (Figure 4);

- welding the polygonal schema along two copies of the same loop is equivalent to simply discarding such loop, unmarking the edges that participate in it. This avoids merging pairs of existing vertices and updating triangles in the data structure.

As can be noticed in the first column of Figure 5, the radial sorting of the basis loops around the origin of the system does not match with the gluing scheme in the polygonal schema. Therefore, to apply the Fulton's algorithm directly on the input mesh it is necessary to devise, from the radial sorting, the gluing scheme that will arise after cutting and flattening. The function f that puts these two orderings in correspondence is defined as

$$f := \begin{cases} gs(0) = rs(0) \\ gs(i+1) = \underset{rs}{\mathrm{next}}\left(\overline{gs(i)}\right) \end{cases} \tag{1}$$

where rs denotes the radial sorting, gs the gluing scheme, and the next operator denotes the next loop in rs after the one passed as argument. Interestingly, when the gluing scheme is in normal form the radial sorting and the gluing scheme coincide (right column in Figure 5). This correspondence is indeed bijective, in fact the inverse function f^{-1} that maps the gluing scheme into the radial sorting is also equal to f, meaning that these functions are closed w.r.t. the normal form.

Figure 5 shows the complete sequence of operations for a manifold with genus 3. Note that the loops generated by iteratively

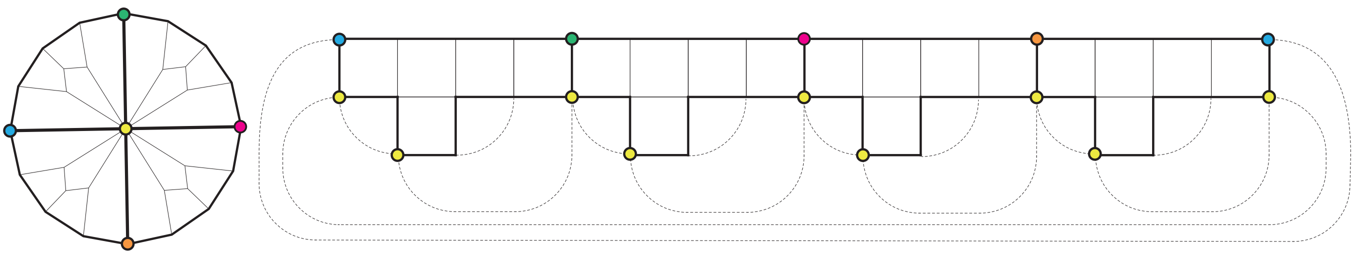


Fig. 6. Mapping Φ_{IGM} for a surface with genus 4, obtained by chaining four copies of the local template shown in Figure 8. Similarly, IGMs for surfaces with higher genus can be trivially obtained by inserting additional copies of the same template. Integer transitions are highlighted with dashed lines. Note that the valence of the vertex at the center of the schema (yellow circle) grows linearly with the genus g, and is $3g$. Also note that all the boundary vertices in the polygonal schema are images of the same mesh vertex, which corresponds to the origin of the system of loops. Its valence in the output quadmesh is $4g$.

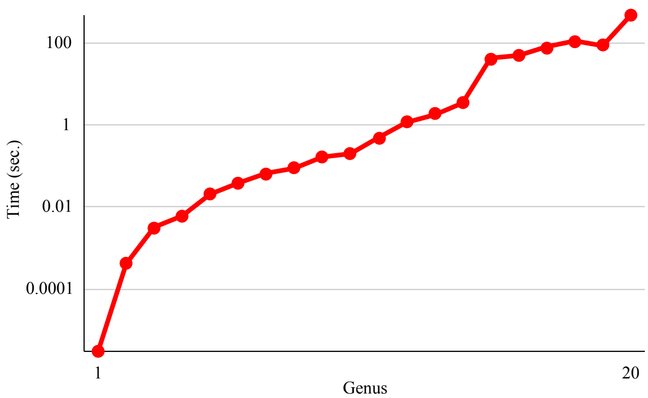


Fig. 7. Time necessary to impose a gluing scheme in normal form for a sequence of cubes with growing number of handles (from 1 to 20). Vertical scale is logarithmic.

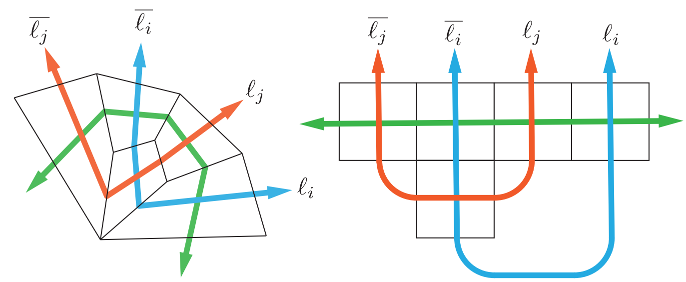


Fig. 8. Topological template used to generate the map Φ_{IGM}. Every sequence of four adjacent sides $\ell_i, \ell_j, \overline{\ell_i}, \overline{\ell_j}$ connects with the center of the polygon to detach a wedge of the polygonal schema. On top of this wedge is laid a quad layout (left) that defines a valid integer grid map (right). Colored arrows show the edge flows of the resulting quadrilateral mesh. The global mapping Φ_{IGM} can be obtained by composing multiple occurrences of this local mapping.

applying the Fulton's algorithm tend to be overly complex, also inducing a considerable amount of distortion in the mapping (bottom right corner). In terms of performances, fixing the gluing scheme on meshes with growing genus seems to have an exponential impact on running times (Figure 7), which are mostly dominated by the mesh refinement necessary to ensure that new loops can always be traced. Nevertheless, the proposed algorithm is guaranteed to always produce a normal form and the mapping is guaranteed injective.

4.2. Computation of Φ_{IGM}

For this phase we heavily exploit the fact that the gluing scheme associated to Φ_{CPS} is in normal form. In fact, this form ensures that the two copies of each loop are always close to each other in the polygonal schema, and that there is only one other loop (i.e., CPS edge) in between them. As a result, the gluing scheme can be decomposed into g fully disjoint subgroups

$$\underbrace{\ell_1, \ell_2, \overline{\ell_1}, \overline{\ell_2}}_{\text{group 1}},\ \underbrace{\ell_3, \ell_4, \overline{\ell_3}, \overline{\ell_4}}_{\text{group 2}},\ \dots,\ \underbrace{\ell_{2g-1}, \ell_{2g}, \overline{\ell_{2g-1}}, \overline{\ell_{2g}}}_{\text{group } g}$$

Note that both images of each cutting loop appear only in one group. This means that every single group isolates a component of the mapping that is completely disjoint from the others. Since the continuity conditions imposed by integer grid maps apply only across boundaries, the global problem of computing an IGM can be split into a sequence of smaller problems that are much easier to deal with.

Every single group $\ell_i, \ell_j, \overline{\ell_i}, \overline{\ell_j}$ defines a wedge of the polygonal schema that is bounded by these four edges on the outside, and by two segments connecting their endpoints with the polygon centroid. An IGM for such a wedge can be computed by overlaying the quadrilateral topological scheme depicted in the left part of Figure 8. The right part of the same figure shows how this translates into an actual grid map with integer transitions. Computing a valid global mapping for the whole surface is as easy as chaining multiple copies of this template in order to reach the wanted genus. An example for the case $g = 4$ is shown in Figure 6.

The actual mapping is computed by detecting – for each vertex in $\Phi_{CPS}(M)$ – the quad that contains it, and then using quadrilateral inverse bilinear coordinates (Section 3 in [41]) to express its position as a function of the quad corners. Note that, for this mapping to be correct, the vertices and edges of the topological layout shown in Figure 8 must be part of the mesh M. In our implementation we compute an arrangement between the polygonal schema and the quad template [42], splitting mesh edges to resolve intersections. Alternatively, one could apply some layout embedding algorithm (e.g. [43]) to convert the quad template into chains of edges of M. However, considering the high valence of the vertex at the center of the

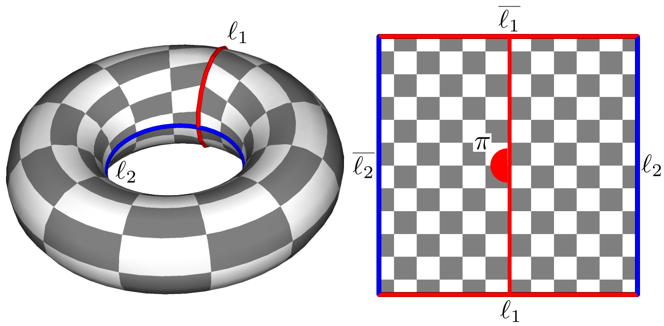


Fig. 9. For the case of surfaces with genus 1 the mapping Φ_{CPS} is already an Integer Grid Map. This is because the canonical polygonal schema is a square, hence the two images of each loop are aligned up to a rotation of π and map to opposite sides of the polygon, ensuring both rotational and translational alignment.

polygonal schema, it is unlikely that a valid embedding could be computed without any refinement, especially for high genus shapes.

5. Special cases

The mapping algorithm described so far works for any $g \in [2, \infty)$. We discuss here the special cases of $g = 0, 1, 2$, which endow similar, but partly different, properties.

Genus 0. For genus zero surfaces there exists no polygonal schema, hence the mapping Φ_{CPS} cannot be constructed as described in Section 4.1. A valid IGM for this class of shapes can still be robustly generated if the mesh is mapped to a tileable parametric space. In [44] Aigerman and Lipman showed that symmetric patterns can be transferred onto a surface through mappings to orbifold embeddings. Specifically, if the orbifold has four cone landmarks, each of angle π, the embedding is also a quadrangulation (see Figure 1 in their article).

Genus 1. The sum of inner angles of a simple polygon with n sides is $(n-2)\pi$. Thus, every inner angle of a regular polygonal schema is

$$\frac{(4g-2)\pi}{4g} \tag{2}$$

which for the case of $g = 1$ becomes $\pi/2$. Indeed, the canonical polygonal schema of topological tori is a simple square with gluing scheme $\ell_1, \ell_2, \overline{\ell_1}, \overline{\ell_2}$. Therefore, the two images of the same cutting loop are aligned up to a rotation of π, which means that the mapping Φ_{CPS} is seamless (in the sense of Figure 1, middle). In addition to this, the two images of each loop map to opposite sides of the square, meaning that there is also a translational alignment, hence Φ_{CPS} is already a valid IGM (Figure 9). In this case, the mapping Φ_{IGM} can be simply set to the identity.

Genus 2. Also for surfaces having genus 2 Φ_{CPS} is a seamless map. In fact, the gluing scheme is $\ell_1, \ell_2, \overline{\ell_1}, \overline{\ell_2}, \ell_3, \ell_4, \overline{\ell_3}, \overline{\ell_4}$ and, according to Equation 2, corner angles are $6\pi/8$, which means that the two images of the same cutting loop are aligned up to a rotation of $3\pi/2$. However, differently from the case $g = 1$, there is no translational alignment, hence the mapping is not a valid IGM (Figure 10). For this reason, the case $g = 2$ does not require special handling, and the mapping can be computed as for any other surface with genus $g > 2$.

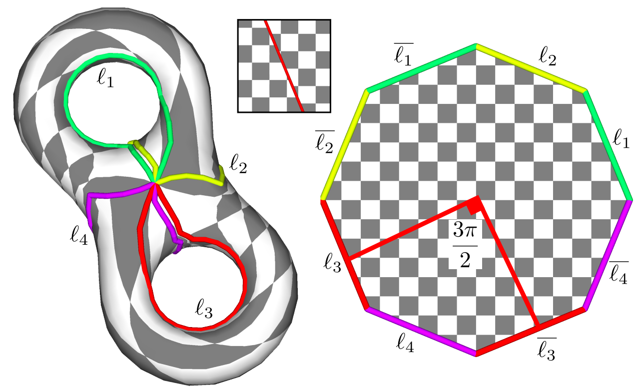


Fig. 10. For the case of surfaces with genus 2 the mapping Φ_{CPS} is seamless. In fact, the two images of the same loop are aligned up to a rotation of $3\pi/2$. Note that, differently from the case of genus 1, the mapping is not an IGM (see the translational misalignment in the closeup). Surfaces of genera 1 and 2 are the only ones that yield a seamless map. Mappings of surfaces of higher genus are neither seamless nor IGMs.

Interestingly, $g = 1, 2$ are the only genera for which Φ_{CPS} becomes seamless. This can be proved by simply observing that, following Equation 2, the rotation angle between two images of the same loop can be written as

$$2\frac{(4g-2)\pi}{4g} = 2\pi - \frac{\pi}{g}\,.$$

The term 2π can be omitted due to the periodicity of rotations. The term π/g becomes smaller than $\pi/2$ for $g > 2$, and goes to 0 only for $g \to \infty$. Thus, for any finite value of $g > 2$ there cannot exist a rotational alignment by an integer multiple of $\pi/2$. □

6. Discussion

We implemented a C++ software prototype to construct the mappings Φ_{CPS} and Φ_{IGM}. Our reference implementation is freely available at `https://github.com/mlivesu/topological_IGM`. Nevertheless, reproducing our method from scratch requires little effort, mostly because many of the necessary ingredients are already available in existing geometry processing toolkits. Specifically, we based our code on Cinolib [45], which implements the greedy homotopy basis algorithm [40], mesh refinement to obtain a valid system of loops [36], construction and mapping to the canonical polygonal schema, the manifold Fulton's algorithm to impose the normal form, inverse bilinear coordinates [41], as well as the portions of [42] that are necessary to robustly detect intersections and split mesh edges to overlay the template in Figure 8 onto the Φ_{CPS} mapping.

In Figure 11 we show some IGMs for surfaces of growing genus. Note that our algorithm puts no limits on the geometric or topological complexity of the input surfaces, and is capable of scaling to shapes with any genus, always providing strict theoretical guarantees of correctness. Namely, all mappings are guaranteed to not contain degenerate or inverted elements, and

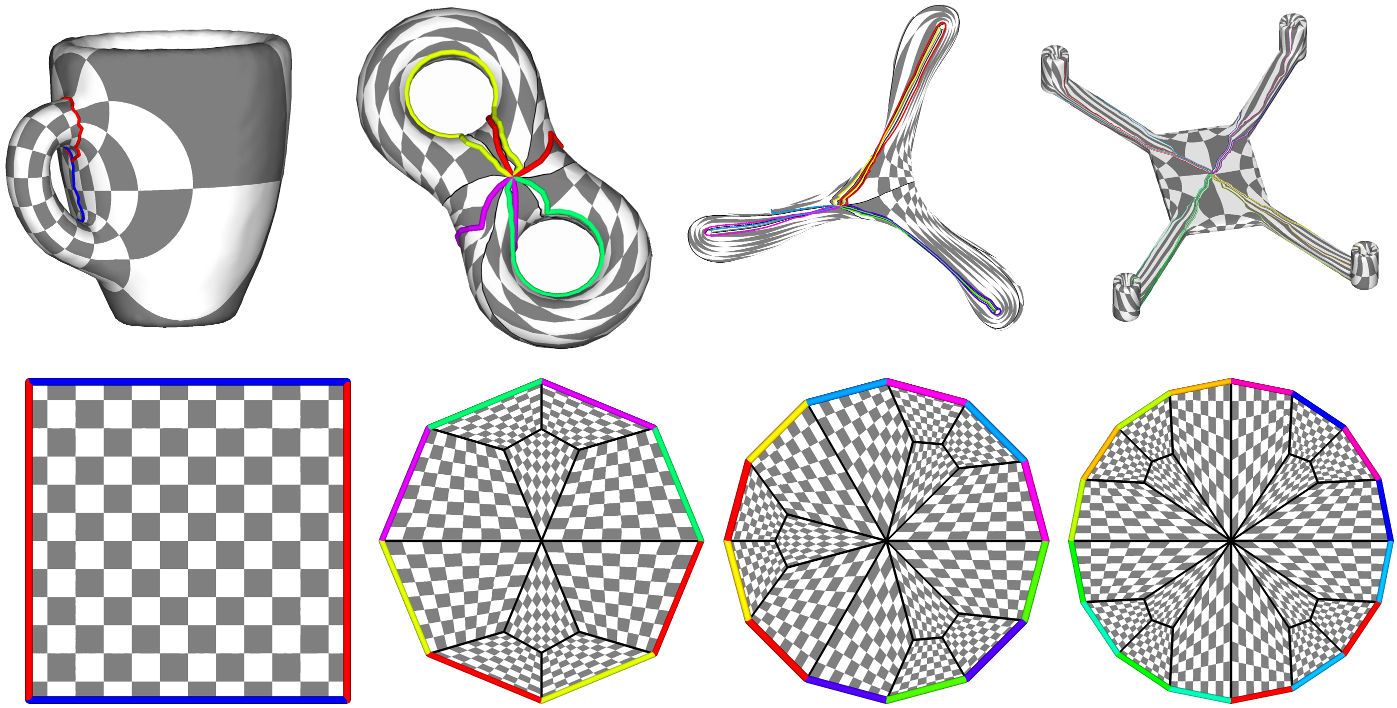
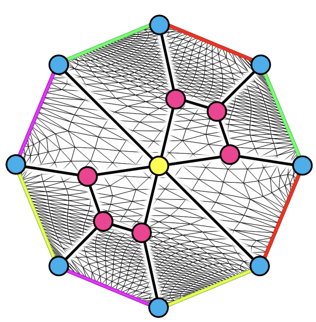


Fig. 11. Integer Grid Maps obtained with our method for surfaces with genus in between 1 and 4 (left to right). For each example, cutting loops are color mapped to the edges of the canonical polygonal schema. Thick black lines denote the topological construction used to overlay the integer grid mapping.



Vertex	Valence	Occurrences
●	$3g$	1
●	$8g$	1
●	3	$3g$

Fig. 12. List of singular vertices in the output quad mesh that are introduced by our topological construction. With g we denote the input mesh genus (in this case $g = 2$). Any additional output vertex would be constructed by refining the thick quads, hence will be regular.

the integer isolines of Φ_{IGM} design a topologically correct quadrangulation of the input surface.

We emphasize once again that the method we propose is not meant to produce an application ready IGM, bur rather to initialize a provably correct IGM that is expected to undergo some robust quality improvement step, both topologically and geometrically. Giving map validity for granted, in the next subsections we report on the main properties of our construction.

6.1. Connectivity

The quadmeshes generated with out approach contain three types of irregular vertices, the occurrence and valence of which depends on the mesh genus g. Overall, there are $3g+2$ irregular vertices, shown in Figure 12 and described here below

- **CPS center (yellow):** the center of the CPS maps to an output vertex in the quadmesh that has valence $3g$. This is because each transition scheme (Figure 8) contributes with 3 incoming edges and, for a mesh with genus g, exactly g transition schemes are neeeded;

- **CPS corners (blue):** the corners of the CPS represent the origin of the homotopy basis and all map to the same vertex in the output mesh. Considering that the schema for a manifold with genus g is a $4g-$gon and that the transition schemes introduce one additional incoming edge to each CPS corner, the overall valence of such a vertex is $8g$;

- **others (pink):** in addition to the vertices above, each transition scheme necessitates three irregular vertices to drive the edge flow and connect the two copies of each loop (Figure 8). Differently form the two types before, in this case the vertex valence is fixed to 3 and the mesh genus only affects the overall number of such irregular vertices, which is $3g$.

Vertices with high valence are typically unwanted, hence the so generated connectivity is not of high quality. This is because these vertices put a tight bound on the geometric quality of their incident elements (see e.g., Figure 2 in [47]). In many practical cases quadmeshes are expected to contain only irregular vertices of valence 3 and 5, possibly connected to one another so as to decompose the surface into a coarse atlas of regular grids [14, 1]. Topological operators to enhance the valence and connectivity of a given quadmesh exist in the literature [48, 49], but performing such operation to obtain a better mesh topology is out of the scope of this article.

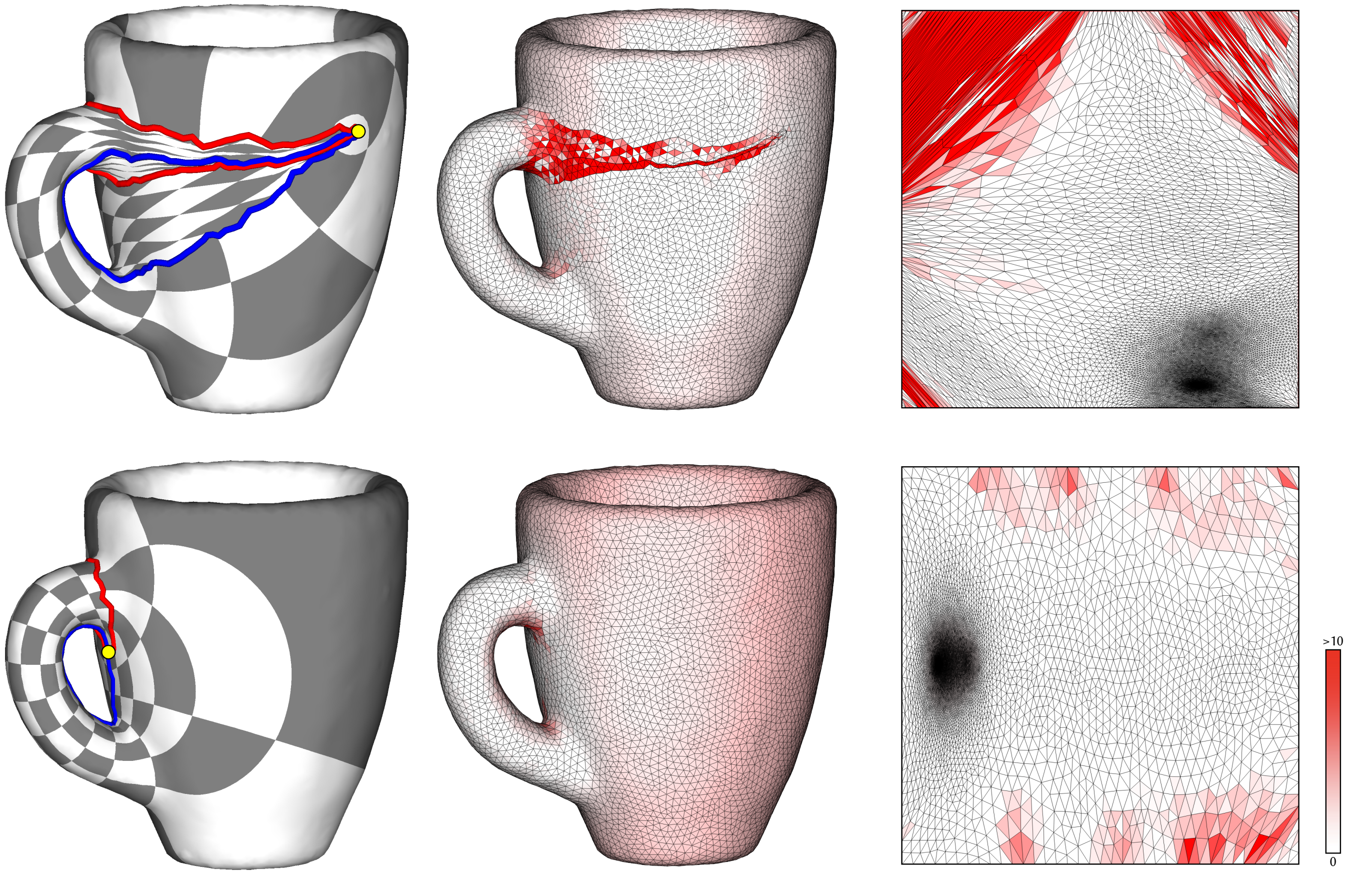


Fig. 13. Distribution of geometric distortion heavily depends on the positioning of the origin of the homotopy basis (left column, yellow circles). As a rule of thumb, it is always convenient to position the origin nearby the shape's handles (bottom line), thus avoiding the creation of long and narrow bundles of loops that connect the origin to each handle, accumulating unnecessary distortion nearby the corners of the polygonal schema (top right). Distortion plots are based on the ARAP energy [46].

6.2. Distortion

Apart from trivial cases such as the torus in Figure 9, mappings to the canonical polygonal schema often suffer from severe geometric distortion. This depends mostly on two reasons, that we analyze in the following paragraphs.

Impact of the homotopy basis. Mapping to a polygonal schema requires cutting each handle along two topologically orthogonal directions (tangentially and transversally). In practice, this means that positioning the origin of the system of loops closed to the mesh handles is always a good idea, because it reduces the length of the cuts hence the possibility for loops to travel in tight bundles towards the origin, creating narrow passages that will inevitably stretch in the polygonal schema. A pictorial illustration of this effect is shown in Figure 13, where a poor positioning of the origin of the homotopy basis clearly accumulates unnecessary distortion around the corners of the flattening. Similar considerations could also be applied to the whole geometry of the loops. The greedy homotopy basis algorithm is designed to create loops with minimal length, but geodesics tend to concentrate around areas of minimal curvature (Sec. 2.2 in [50]) thus promoting the generation of narrow passages also distant from the origin of the basis. Systems of loops that yield a mapping with lower geometric distortion could likely be created by incorporating a different metric into the greedy homotopy basis algorithm [40], namely substituting the geodesic distance with a distortion aware metric or with a repulsive energy that pushes loops away from each other [51]. Such a modification is technically feasible and was recently explored in [52] to align the homotopy basis with the cut locus of a distance field.

Impact of mesh genus. The number of handles negatively impacts geometric distortion, especially nearby the origin of the system of loops. To understand this connection one should recall that the origin of the system is a mesh vertex which has $2g$ loops (i.e. $4g$ incoming cuts). Cutting along all loops splits the neighborhood of such vertex into $4g$ wedges that map to the corners of the polygonal schema. Starting from Equation 2 it is easy to show that corner angles in the polygonal schema tend to π for $g \rightarrow \infty$. Conversely, wedge angles in the input surface tend to 0 for growing values of g, because the solid angle at the origin of the system is divided by a progressively bigger number of wedges. As a result, the more g grows the more narrow wedges will map to open corners of the polygonal schema, introducing inevitable stretching.

7. Conclusions and Future Works

We have presented a novel topological construction to generate provably correct integer grid maps for surfaces of any genus. The proposed method widely exploits tools from algebraic topology, and is based on the composition between a mapping to the canonical polygonal schema and a quadrilateral template that allows to obtain the necessary integer continuity across cutting seams. Differently from existing robust methods, our topological construction does not rely on commercial solvers and can be readily deployed on both commercial and open source frameworks.

As anticipated in Section 1 the ultimate goal of this research is to realize an application ready pipeline for the computation of integer grid maps, of which the algorithm described in Section 4 is intended to be only the initialization step. Therefore not surprisingly, the analysis of the geometric and topological properties of our results in Section 6 revealed that the mappings are currently overly distorted and endow a poor mesh connectivity. The improvement of both aspects is the goal of the second step of the pipeline and will be the principal subject for future works.

For the geometric distortion, barrier energies and line search with rollback operators recently introduced for robust injective mapping already proved effective [7, 8, 9, 10, 11]. For the improvement of the mesh connectivity atomic topological operators such as the ones described in [53, 49, 54, 48] will be explored. Specifically, the vertex split operator allows to reduce vertex valence by one, whereas the quad collapse operator allows to increase the valence by one (Figure 2 in [48]). It is still unclear how these ingredients could be combined. A tempting solution would be to interleave them, optimizing for geometry and topology alternatively. Similar approaches were already used in the past, e.g. for the case of abstract domains [55, 56], although in that case topological changes were only temporary and had the function to ensure that all mesh vertices were free to move at least once in order to reduce geometric distortion.

Finally, the extension of similar techniques to the third dimension (i.e. to generate hexahedral meshes) is an interesting avenue for future works. In fact, volumetric integer grid maps has gained attention in the hexmesh community, and their reliable computation is an important open challenge [57]. Unfortunately, the canonical polygonal schema has no direct counterpart in the realm of 3-manifolds, and it remains unclear whether similar topological constructions could be exploited to realize an initial mapping on top of which a templated hex transition could be installed. Besides the difficulties in generating a suitable parametric space, it is also worth reminding that the methods we use to map a surface in a convex domain do not extend to 3D. The question whether the Tutte embedding could be extended to tetrahedral meshes has been open for a long time. Very recent findings show that despite this seems to be possible for a restricted class of topologies, the problem admits no solution for practically relevant meshes [58]. The difficulties of extending alternative mapping approaches to the volume setting are also discussed in [59] (Section 2). We therefore expect that guaranteeing the map injectivity on volumes would be extremely challenging.

Acknowledgments

Thanks are due to Giovanni Placini and Benedetto Manca for insightful discussions on polygonal schemata and for helping me porting the Fulton's algorithm to the manifold setting.

References

[1] Bommes, D, Lévy, B, Pietroni, N, Puppo, E, Silva, C, Tarini, M, et al. Quad-mesh generation and processing: A survey. In: Computer Graphics Forum; vol. 32. 2013,.

[2] Bommes, D, Zimmer, H, Kobbelt, L. Mixed-integer quadrangulation. ACM Transactions on Graphics 2009;28(3).

[3] Kälberer, F, Nieser, M, Polthier, K. Quadcover-surface parameterization using branched coverings. In: Computer Graphics Forum; vol. 26. 2007,.

[4] Bommes, D, Campen, M, Ebke, HC, Alliez, P, Kobbelt, L. Integer-grid maps for reliable quad meshing. ACM Transactions on Graphics 2013;32(4).

[5] Bommes, D, Zimmer, H, Kobbelt, L. Practical mixed-integer optimization for geometry processing. In: International Conference on Curves and Surfaces. Springer; 2010, p. 193–206.

[6] Campen, M, Bommes, D, Kobbelt, L. Quantized global parametrization. ACM Transactions on Graphics 2015;34(6).

[7] Rabinovich, M, Poranne, R, Panozzo, D, Sorkine-Hornung, O. Scalable locally injective mappings. ACM Transactions on Graphics 2017;36(4).

[8] Smith, J, Schaefer, S. Bijective parameterization with free boundaries. ACM Transactions on Graphics 2015;34(4).

[9] Jiang, Z, Schaefer, S, Panozzo, D. Simplicial complex augmentation framework for bijective maps. ACM Transactions on Graphics 2017;36(6).

[10] Su, JP, Ye, C, Liu, L, Fu, XM. Efficient bijective parameterizations. ACM Transactions on Graphics 2020;39(4).

[11] Liu, L, Ye, C, Ni, R, Fu, XM. Progressive parameterizations. ACM Transactions on Graphics 2018;37(4).

[12] Tutte, WT. How to draw a graph. Proceedings of the London Mathematical Society 1963;3(1).

[13] Shen, H, Jiang, Z, Zorin, D, Panozzo, D. Progressive embedding. ACM Transactions on Graphics 2019;38(4).

[14] Campen, M. Partitioning surfaces into quadrilateral patches: A survey. In: Computer Graphics Forum; vol. 36. 2017,.

[15] Livesu, M. Topological Initialization of Injective Integer Grid Maps. In: Smart Tools and Applications in Graphics (STAG). 2022,.

[16] Ray, N, Li, WC, Lévy, B, Sheffer, A, Alliez, P. Periodic global parameterization. ACM Transactions on Graphics 2006;25(4).

[17] Tong, Y, Alliez, P, Cohen-Steiner, D, Desbrun, M. Designing quadrangulations with discrete harmonic forms. In: Eurographics Symposium on Geometry Processing. 2006,.

[18] Ebke, HC, Bommes, D, Campen, M, Kobbelt, L. Qex: Robust quad mesh extraction. ACM Transactions on Graphics 2013;32(6).

[19] Pietroni, N, Nuvoli, S, Alderighi, T, Cignoni, P, Tarini, M. Reliable feature-line driven quad-remeshing. In: ACM SIGGRAPH; vol. 40. 2021,.

[20] Campen, M, Zorin, D. Similarity maps and field-guided t-splines: a perfect couple. ACM Transactions on Graphics 2017;36(4).

[21] Lyon, M, Campen, M, Bommes, D, Kobbelt, L. Parametrization quantization with free boundaries for trimmed quad meshing. ACM Transactions on Graphics 2019;38(4).

[22] Campen, M, Shen, H, Zhou, J, Zorin, D. Seamless parametrization with arbitrary cones for arbitrary genus. ACM Transactions on Graphics 2019;39(1).

[23] Chien, E, Levi, Z, Weber, O. Bounded distortion parametrization in the space of metrics. ACM Transactions on Graphics 2016;35(6).

[24] Zhou, J, Tu, C, Zorin, D, Campen, M. Combinatorial construction of seamless parameter domains. In: Computer Graphics Forum; vol. 39. 2020,.

[25] Levi, Z. Direct seamless parametrization. ACM Transactions on Graphics 2021;40(1).

[26] Myles, A, Zorin, D. Global parametrization by incremental flattening. ACM Transactions on Graphics 2012;31(4).

[27] Myles, A, Zorin, D. Controlled-distortion constrained global parametrization. ACM Transactions on Graphics 2013;32(4).

[28] Myles, A, Pietroni, N, Zorin, D. Robust field-aligned global parametrization. ACM Transactions on Graphics 2014;33(4).
[29] Shen, H, Zhu, L, Capouellez, R, Panozzo, D, Campen, M, Zorin, D. Which cross fields can be quadrangulated? global parameterization from prescribed holonomy signatures. ACM Transactions on Graphics 2022;41(4).
[30] Bright, A, Chien, E, Weber, O. Harmonic global parametrization with rational holonomy. ACM Transactions on Graphics 2017;36(4).
[31] Tarini, M. Closed-form quadrangulation of n-sided patches. Computers & Graphics 2022;107.
[32] Takayama, K, Panozzo, D, Sorkine-Hornung, O. Pattern-based quadrangulation for n-sided patches. In: Computer Graphics Forum; vol. 33. 2014,.
[33] Fulton, W. Algebraic topology: a first course; vol. 153. Springer Science & Business Media; 1997.
[34] Erickson, J, Har-Peled, S. Optimally cutting a surface into a disk. Discrete & Computational Geometry 2004;31(1).
[35] Brahana, HR. Systems of circuits on two-dimensional manifolds. Annals of Mathematics 1921;:144–168.
[36] Livesu, M. Scalable mesh refinement for canonical polygonal schemas of extremely high genus shapes. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2021;27(1).
[37] Li, X, Bao, Y, Guo, X, Jin, M, Gu, X, Qin, H. Globally optimal surface mapping for surfaces with arbitrary topology. IEEE Transactions on Visualization and Computer Graphics 2008;14(4).
[38] Garner, C, Jin, M, Gu, X, Qin, H. Topology-driven surface mappings with robust feature alignment. In: VIS 05. IEEE Visualization, 2005. 2005,.
[39] Wang, H, He, Y, Li, X, Gu, X, Qin, H. Polycube splines. Computer-Aided Design 2008;40(6).
[40] Erickson, J, Whittlesey, K. Greedy optimal homotopy and homology generators. In: SODA; vol. 5. 2005,.
[41] Floater, MS. Generalized barycentric coordinates and applications. Acta Numerica 2015;24.
[42] Cherchi, G, Livesu, M, Scateni, R, Attene, M. Fast and robust mesh arrangements using floating-point arithmetic. ACM Transactions on Graphics (SIGGRAPH Asia 2020) 2020;39(6).
[43] Born, J, Schmidt, P, Kobbelt, L. Layout embedding via combinatorial optimization. In: Computer Graphics Forum; vol. 40. 2021,.
[44] Aigerman, N, Lipman, Y. Orbifold tutte embeddings. ACM Transactions on Graphics 2015;34(6).
[45] Livesu, M. cinolib: a generic programming header only c++ library for processing polygonal and polyhedral meshes. In: Transactions on Computational Science XXXIV. Springer; 2019, p. 64–76.
[46] Liu, L, Zhang, L, Xu, Y, Gotsman, C, Gortler, SJ. A local/global approach to mesh parameterization. In: Computer Graphics Forum; vol. 27. 2008,.
[47] Livesu, M, Pitzalis, L, Cherchi, G. Optimal dual schemes for adaptive grid based hexmeshing. ACM Transactions on Graphics 2022;41(2).
[48] Peng, CH, Zhang, E, Kobayashi, Y, Wonka, P. Connectivity editing for quadrilateral meshes. In: Proceedings of the 2011 SIGGRAPH Asia conference. 2011,.
[49] Feng, L, Tong, Y, Desbrun, M. Q-zip: singularity editing primitive for quad meshes. ACM Transactions on Graphics 2021;40(6).
[50] Crane, K, Livesu, M, Puppo, E, Qin, Y. A survey of algorithms for geodesic paths and distances. arXiv preprint arXiv:200710430 2020;.
[51] Yu, C, Schumacher, H, Crane, K. Repulsive curves. ACM Transactions on Graphics 2021;40(2).
[52] Mancinelli, C, Livesu, M, Puppo, E. Practical computation of the cut locus on discrete surfaces. Computer Graphics Forum 2021;40(5).
[53] Tarini, M, Pietroni, N, Cignoni, P, Panozzo, D, Puppo, E. Practical quad mesh simplification. In: Computer Graphics Forum; vol. 29. 2010,.
[54] Bommes, D, Lempfer, T, Kobbelt, L. Global structure optimization of quadrilateral meshes. In: Computer Graphics Forum; vol. 30. 2011,.
[55] Tarini, M, Puppo, E, Panozzo, D, Pietroni, N, Cignoni, P. Simple quad domains for field aligned mesh parametrization. In: Proceedings of the 2011 SIGGRAPH Asia Conference. 2011,.
[56] 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 2015;35(1).
[57] Pietroni, N, Campen, M, Sheffer, A, Cherchi, G, Bommes, D, Gao, X, et al. Hex-mesh generation and processing: a survey. ACM transactions on graphics 2022;42(2).
[58] Alexa, M. Tutte embeddings of tetrahedral meshes. arXiv preprint arXiv:221200452 2022;.
[59] Livesu, M. A Mesh Generation Perspective on Robust Mappings. In: Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference. 2020,.