STAG: Smart Tools and Applications in Graphics (2022)
D. Cabiddu, T. Schneider, and G. Cherchi (Editors)

Topological Initialization of Injective Integer Grid Maps

Marco Livesu
CNR IMATI

Abstract

Integer Grid Maps (IGM) 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. Computing an IGM is notoriously a challenging task, because it requires to solve a numerical problem with mixed discrete and continuous variables which is known to be NP-Hard. As a result, state of the art methods rely on heuristics that may occasionally fail to produce a valid quadrilateral mesh. Existing pipelines incorporate a final sanitization step which attempts to fix such defects, but no guaranteees can be given in this regard. In this paper we propose a simple topological construction that allows to reduce the problem of computing an IGM to the one of mapping a topological disk to a convex domain. This is a much easier problem to deal with, because it does not endow integer translational and rotational constraints, permitting to obtain a parameterization that is guaranteed to incorporate all the correct integer transitions and to not contain degenerate or inverted elements. 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.

CCS Concepts
• ***Computing methodologies*** $\rightarrow$ ***Mesh models;***

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 [BLP*13]. Since the introduction of seminal works such as [BZK09; KNP07], 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 and also align to both principal curvatures and sharp features. Specifically, the class of mappings that allow to lay a quadrilateral grid onto a discrete surface are called Integer Grid Maps [BCE*13], or IGMs in short.

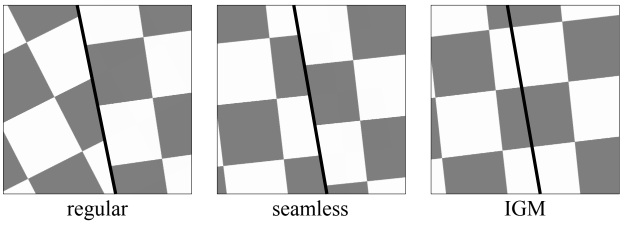

Figure 1: *Regular mappings impose no continuity across cuts. Seamless maps ensure that parametric isolines rotationally align up to multiples of 90°, but the grids at opposite sides of the cut may still expose translational misalignment. Integer Grid Maps ensure both rotational and translational alignment, so that contouring the integer isolines of the mapping yields a pure quad tessellation.*

The distinctive trait of an IGM is that it ensures continuity of the integer level sets of the underlying mapping 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 extremely challenging mixed-integer problem that is known to be NP-Hard [BZK10]. Nevertheless, modern mixed-integer solvers are still able to compute quality quadmeshes, but rely on heuristic approaches that do not offer guarantees of correctness and may unexpectedly fail to produce a valid result (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. A major source of inspiration comes from recent approaches for the robust computation of injective simplicial mappings, such as [RPPS17; SS15; JSP17; SYLF20; LYNF18].

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 map injectivity is preserved throughout the process. The first step is typically obtained with Tutte or similar alternatives [Tut63; SJZP19]. 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.

The ultimate goal of this line of research is 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 the 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 map 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 should contain few singular vertices which are well connected to one another to form a coarse quad layout [Cam17].

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).

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 [BLP*13].

Integer Grid Maps. Pioneering works in the field [BZK09; KNP07; RLL*06; BCE*13; TACD06] have established surface mappings as a prominent method for the generation of quadrilateral meshes. Integer Grid Mapings compute a mesh by solving a complex mixed-integer problem that is known to be NP-Hard [BZK10]. 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 [BZK09] 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. As stiffening alone was not enough, in [BCE*13] (Sec. 3.1) a set of linear anti-flip constraints that assign triangle vertices to three disjoint semi-infinite convex sub-spaces is used. However, these constraints are just a linearization of a more general non linear functional, and 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 [EBCK13], 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. The topological approach described in this paper cannot compete with these methods in terms of topological and geometric quality of the output quadmesh, but it is guaranteed to always produce an injective IGM from which a correct quadmesh can be trivially extracted. To this end, it can be seen as a powerful initialization for a robust IGM pipeline. For completeness, it should be noted that if a injective seamless mapping is known, global quantization [CBK15] can provide integer transitions guaranteeing a correct result. Nevertheless, we believe that exploring alternative robust methods that take care of all the necessary IGM requirements altogether is still an interesting research topic.

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 [CZ17] or as an initialization step for the computation of an IGM through quantization [CBK15; LCBK19]. Literature in the field mostly deals with the generation of parametric spaces that conform to a prescribed set of cone singularities or holonomy signature [CSZZ19; CLW16; ZTZC20; Lev21; MZ12; MZ13; MPZ14; SZC*22; BCW17]. 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 [Tar22; TPS14]. 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 [Tar22] would equally provide valid solutions to our problem.

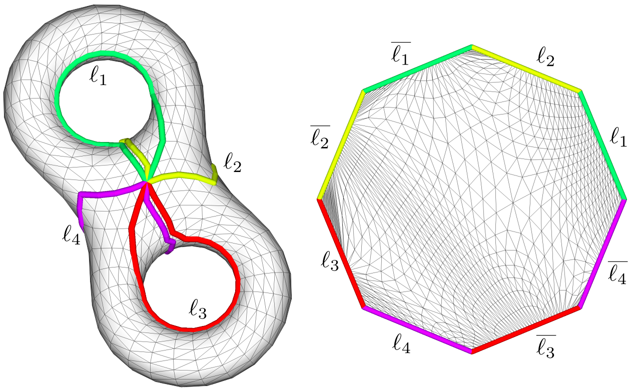

Figure 2: *The mapping Φ_{CPS} creates a one-to-one correspondence between a closed surface with genus g and a regular $4g-$gon. Cutting loops on the surface intersect at a mesh vertex and are fully disjoint elsewhere. The two images of each loop map to the edges of the polygon according to the gluing 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 [Tut63].*

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 [Ful97] 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 [EH04]. 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* [Bra21] (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 [Liv21]. Given a surface mesh M with genus g and a system of loops $L_M = \{\ell_1, \ell_2, \ldots, \ell_{2g}\}$, there exist multiple ways to map loop 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 [Ful97], Chapter 17b).

Canonical poligonal schema are a topological invariant. So far, this property has been exploited in computer graphics to compute cross-parameterizations between pairs of homotopic shapes [LBG*08; GJGQ05; WHL*08]. 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.

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. 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:

$$\begin{tikzcd} M \arrow[r, leftrightarrow, "\Phi_{CPS}"'] & CPS \arrow[r, leftrightarrow, "\Phi_{IGM}"'] & \text{regular grid} \end{tikzcd}$$

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 (i.e. minimal) polygonal 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 [Liv21]. Specifically, we first compute an initial cut graph using the *greedy homotopy basis* algorithm of Erickson and Whittlesey [EW05]. Since the loops produced by this method are not necessarily disjoint, we separate loops that collapsed to the same chains of mesh edges using the edge split strategy described in [Liv21], obtaining a system of fully disjoint loops $L_M = \{\ell_0, \ell_1, \ldots, \ell_{2g}\}$. We then cut the mesh along L_M, duplicating each loop and mapping the so generated topological disk to a regular $4g-$gon using the Tutte embedding [Tut63]. Due to the convexity of the polygonal schema, this mapping is guaranteed to be injective. A visual illustration of Φ_{CPS} is shown in Figure 2.

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

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

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

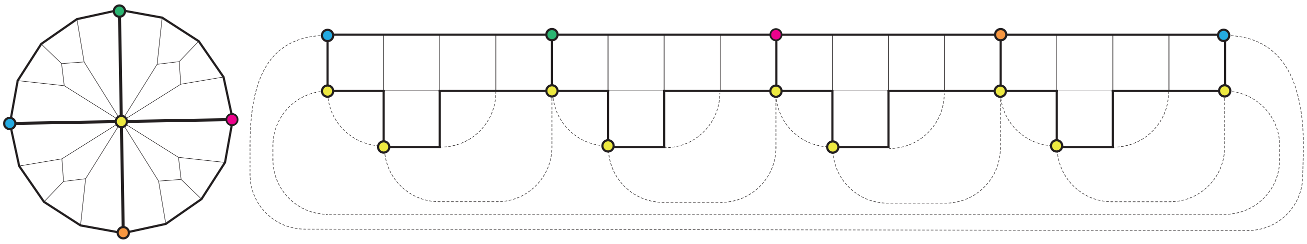

Figure 3: *Mapping Φ_{IGM} for a surface with genus 4, obtained by chaining four copies of the local template shown in Figure 4. Similarly, IGMs for surfaces with higher genus can be trivially obtained by inserting additional copies of the same template. Integer transitions are highlighted with dashed 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* $4g$.

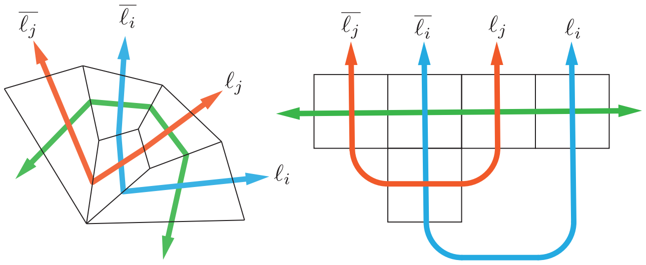

Figure 4: *Topological template used to generate the map Φ_{IGM}. Every sequence of four adjacent sides $\ell_i, \ell_j, \overline{\ell_i}, \overline{\ell_j}$ connects with the center of the polygon to detach a wedge of the polygonal schema. On top of this wedge is laid a quad layout (left) that defines 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.*

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 4. 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 3.

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 [Flo15]) 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 4 must be part of the mesh M. In our implementation we compute an arrangement between the polygonal schema and the quad template [CLSA20], splitting mesh edges to resolve intersections. Alternatively, one could apply some layout embedding algorithm (e.g. [BSK21]) to convert the quad template into chains of edges of M. However, considering the high valence of the vertex at the center of the 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 [AL15] 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

$$\frac{(4g-2)\pi}{4g} \tag{1}$$

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 5). 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 1, corner angles are $6\pi/8$, which means that

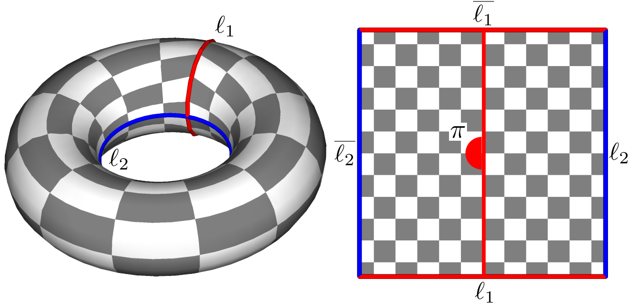

Figure 5: *For the case of surfaces with genus 1 the mapping Φ_{CPS} is already an Integer Grid Map. This is because the canonical polygonal schema is a square, hence the two images of each loop are aligned up to a rotation of π and map to opposite sides of the polygon, ensuring both rotational and translational alignment.*

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 6). 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 $g > 2$.

Interestingly, $g = 1,2$ are the only genera for which Φ_{CPS} becomes seamless. This can be proved by simply observing that, following Equation 1, 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 [Liv19], which implements the greedy homotopy basis algorithm [EW05], mesh refinement to obtain a valid system of loops [Liv21], construction and mapping to the canonical polygonal schema, inverse bilinear coordinates [Flo15], as well as the portions of [CLSA20] that are necessary to robustly detect intersections and split mesh edges to overlay the template in Figure 4 onto the Φ_{CPS} mapping.

In Figure 7 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

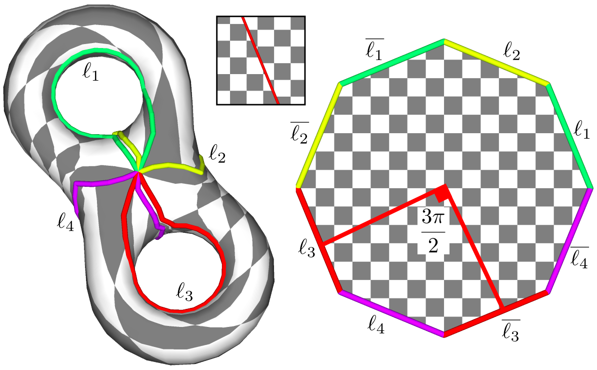

Figure 6: *For the case of surfaces with genus 2 the mapping Φ_{CPS} is seamless. In fact, the two images of the same loop are aligned up to a rotation of $3\pi/2$. Note that, differently from the case of genus 1, the mapping is not an IGM (see the translational misalignment in the closeup). Surfaces 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.*

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 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. Giving map validity for granted, in the next subsections we report the main quality shortcomings of our construction, which regard geometric distortion and mesh connectivity.

6.1. Distortion

Apart from trivial cases such as the torus in Figure 5, mappings to the canonical polygonal schema unavoidably suffer from severe geometric distortion. This is partly due to an inverse correlation between angles in the input surface and angles in the polygonal schema that depends solely on the genus g and that badly affects the mapping quality.

To understand this connection one should recall that the origin of the system of loops in the input surface 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 1 it is easy to show that corner angles in the polygonal schema tend to π for $g \to \infty$. 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 stretch because they are mapped to open corners of the polygonal schema. As can be noticed from the two rightmost mappings in Figure 7 bundles of cutting loops may travel parallel to each other for long

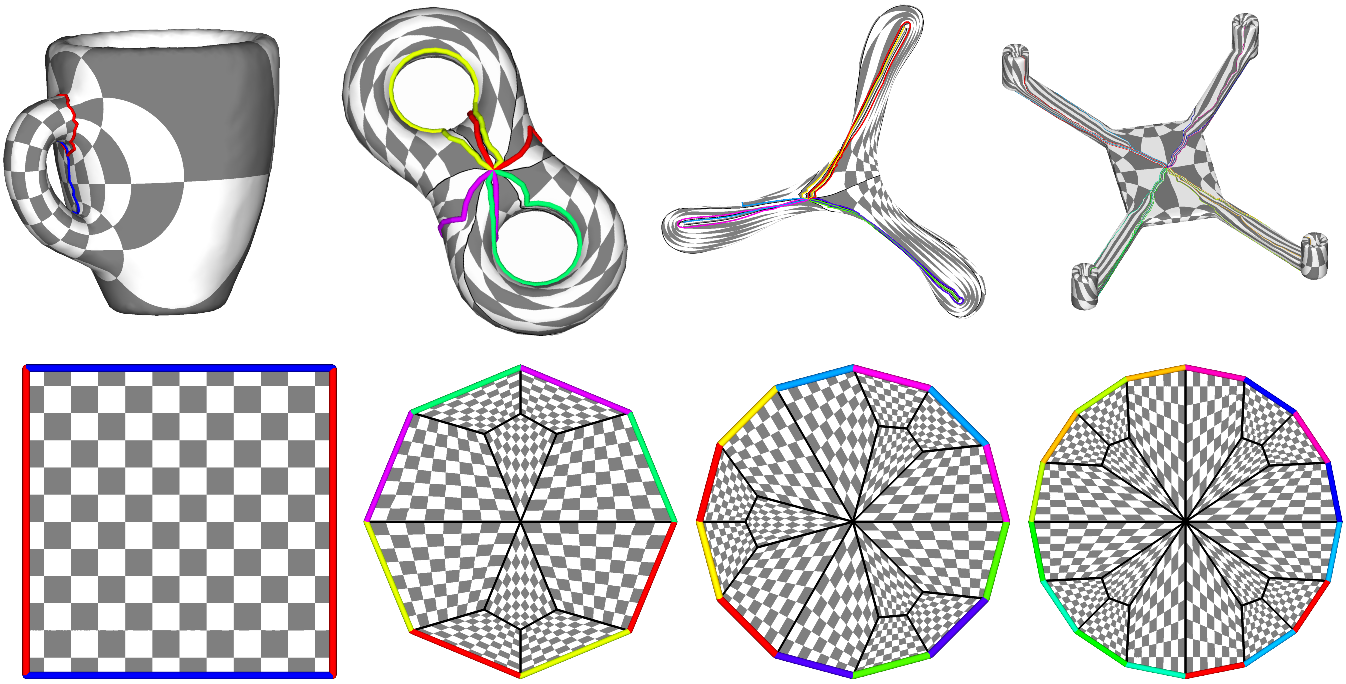

Figure 7: *Integer Grid Maps obtained with our method for surfaces with genus in between 1 and 4 (left to right). For each example, cutting loops are color mapped to the edges of the canonical polygonal schema. Thick black lines denote the topological construction used to overlay the integer grid mapping.*

distances in the input surface, generating a stretch that is not confined to the local neighborhood of the origin of the system of loops, but rather propagates to large areas of the mapping. Unfortunately, this distortion is intrinsically encoded in this topological construction and cannot be avoided.

6.2. Connectivity

As widely discussed in [Cam17; BLP*13] many applications require quadrilateral meshes to endow a good connectivity, meaning that the singular vertices of the mesh must have a controlled valence (i.e. number of incoming edges) and are connected to one another so as to decompose the surface into an atlas of few regular grids. The topological construction in Section 4 falls short on this requirement, because it tends to produce a singular structure that, despite coarse, contains badly shaped charts that are bounded by singular vertices with possibly very high valence.

Similarly to geometric distortion, the amount and valence of singular vertices fully depends on the mesh genus g. This relation can be understood by analyzing Figure 3, which shows all (and only) singular mesh vertices that are introduced by our topological construction. Specifically, we count three different types of singular vertices:

- *i)* all corners in the polygonal schema map to the same mesh vertex in the input surface, which corresponds to the origin of the system of loops. As a result, the valence of such vertex in the output quadrilateral mesh depends on the genus g, and corresponds to the number of vertices in the polygonal schema plus the incoming edges of the templated scheme, that is, $8g$;
- *ii)* the center of the polygonal schema is the only point shared by all the copies of the atomic template in Figure 4. Since such template is repeated g times and each template increases its valence by 3, the final valence of such a vertex in the output quadmesh is $3g$;
- *iii)* the remaining singular vertices control the edge flow inside the template shown in Figure 4, ensuring continuity across the cutting loops. They arise only at the interior of each template, therefore the genus g does not affect their valence, but only the number of their occurrences. Specifically, since each copy of the template requires 3 vertices with valence 3, the output mesh will contain exactly $3g$ valence 3 vertices overall.

Note that vertices with high valence are typically unwanted, e.g. because they put a tight bound on the quality of their incident elements (see Figure 2 in [LPC22]). In many practical cases, quadmeshes are expected to contain only vertices with valence 3,4,5. Vertices with different valence can be split, transforming them into multiple singular vertices of this restricted valence [PZKW11]. Editing the mesh connectivity in order perform such operation and obtain a good mesh topology is out of the scope of this article.

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.

As anticipated in Section 1, the ultimate goal of this research is to realize a provably robust pipeline for the computation of integer grid maps, of which the mapping algorithm described in Section 4 is intended to be the first 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.

Note that differently from similar robust pipelines for provably injective simplicial maps, for the case of an IGM the quality improvement regards not only geometric, but also topological aspects.

For the geometric aspects, barrier energies and line search with rollback operators used in prior art already proved effective [RPPS17; SS15; JSP17; SYLF20; LYNF18]. For the improvement of the mesh connectivity atomic topological operators such as the ones described in [TPC*10; FTD21; BLK11; PZKW11] will be explored. It is still unclear how these two ingredients could be combined. A tempting option 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 [TPP*11; ULP*15], although in that case topological changes were only temporary and had the only 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 [PCS*22]. 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 robust methods to map a surface in a convex domain do not extend to 3D (see Section 2 in [Liv20]), therefore guaranteeing the map injectivity would be extremely challenging.

References

[AL15] Aigerman, Noam and Lipman, Yaron. "Orbifold Tutte embeddings." *ACM Trans. Graph.* 34.6 (2015), 190–1 4.

[BCE*13] Bommes, David, Campen, Marcel, Ebke, Hans-Christian, et al. "Integer-grid maps for reliable quad meshing". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 1–12 1, 2.

[BCW17] Bright, Alon, Chien, Edward, and Weber, Ofir. "Harmonic global parametrization with rational holonomy". *ACM Transactions on Graphics (TOG)* 36.4 (2017), 1–15 2.

[BLK11] Bommes, David, Lempfer, Timm, and Kobbelt, Leif. "Global structure optimization of quadrilateral meshes". *Computer graphics forum*. Vol. 30. 2. Wiley Online Library. 2011, 375–384 7.

[BLP*13] Bommes, David, Lévy, Bruno, Pietroni, Nico, et al. "Quad-mesh generation and processing: A survey". *Computer Graphics Forum*. Vol. 32. 6. Wiley Online Library. 2013, 51–76 1, 2, 6.

[Bra21] Brahana, Henry Roy. "Systems of circuits on two-dimensional manifolds". *Annals of Mathematics* (1921), 144–168 3.

[BSK21] Born, Janis, Schmidt, Patrick, and Kobbelt, Leif. "Layout embedding via combinatorial optimization". *Computer Graphics Forum*. Vol. 40. 2. Wiley Online Library. 2021, 277–290 4.

[BZK09] Bommes, David, Zimmer, Henrik, and Kobbelt, Leif. "Mixed-integer quadrangulation". *ACM Transactions On Graphics (TOG)* 28.3 (2009), 1–10 1, 2.

[BZK10] Bommes, David, Zimmer, Henrik, and Kobbelt, Leif. "Practical mixed-integer optimization for geometry processing". *International Conference on Curves and Surfaces*. Springer. 2010, 193–206 1, 2.

[Cam17] Campen, Marcel. "Partitioning surfaces into quadrilateral patches: A survey". *Computer graphics forum*. Vol. 36. 8. Wiley Online Library. 2017, 567–588 2, 6.

[CBK15] Campen, Marcel, Bommes, David, and Kobbelt, Leif. "Quantized global parametrization". *Acm Transactions On Graphics (tog)* 34.6 (2015), 1–12 2.

[CLSA20] Cherchi, Gianmarco, Livesu, Marco, Scateni, Riccardo, and Attene, Marco. "Fast and Robust Mesh Arrangements using Floating-point Arithmetic". *ACM Transactions on Graphics (SIGGRAPH Asia 2020)* 39.6 (2020). doi: 10.1145/3414685.3417818 4, 5.

[CLW16] Chien, Edward, Levi, Zohar, and Weber, Ofir. "Bounded distortion parametrization in the space of metrics". *ACM Transactions on Graphics (TOG)* 35.6 (2016), 1–16 2.

[CSZZ19] Campen, Marcel, Shen, Hanxiao, Zhou, Jiaran, and Zorin, Denis. "Seamless parametrization with arbitrary cones for arbitrary genus". *ACM Transactions on Graphics (TOG)* 39.1 (2019), 1–19 2.

[CZ17] Campen, Marcel and Zorin, Denis. "Similarity maps and field-guided T-splines: a perfect couple". *ACM Transactions on Graphics (TOG)* 36.4 (2017), 1–16 2.

[EBCK13] Ebke, Hans-Christian, Bommes, David, Campen, Marcel, and Kobbelt, Leif. "QEx: Robust quad mesh extraction". *ACM Transactions on Graphics (TOG)* 32.6 (2013), 1–10 2.

[EH04] Erickson, Jeff and Har-Peled, Sariel. "Optimally cutting a surface into a disk". *Discrete & Computational Geometry* 31.1 (2004), 37–59 3.

[EW05] Erickson, Jeff and Whittlesey, Kim. "Greedy optimal homotopy and homology generators". *SODA*. Vol. 5. 2005, 1038–1046 3, 5.

[Flo15] Floater, Michael S. "Generalized barycentric coordinates and applications". *Acta Numerica* 24 (2015), 161–214 4, 5.

[FTD21] Feng, Leman, Tong, Yiying, and Desbrun, Mathieu. "Q-zip: singularity editing primitive for quad meshes". *ACM Transactions on Graphics (TOG)* 40.6 (2021), 1–13 7.

[Ful97] Fulton, William. *Algebraic topology: a first course*. Vol. 153. Springer Science & Business Media, 1997 3.

[GJGQ05] Garner, C, Jin, Miao, Gu, Xianfeng, and Qin, Hong. "Topology-driven surface mappings with robust feature alignment". *VIS 05. IEEE Visualization, 2005.* IEEE. 2005, 543–550 3.

[JSP17] Jiang, Zhongshi, Schaefer, Scott, and Panozzo, Daniele. "Simplicial complex augmentation framework for bijective maps". *ACM Transactions on Graphics* 36.6 (2017) 2, 7.

[KNP07] Kälberer, Felix, Nieser, Matthias, and Polthier, Konrad. "Quadcover-surface parameterization using branched coverings". *Computer graphics forum*. Vol. 26. 3. Wiley Online Library. 2007, 375–384 1, 2.

[LBG*08] Li, Xin, Bao, Yunfan, Guo, Xiaohu, et al. "Globally optimal surface mapping for surfaces with arbitrary topology". *IEEE Transactions on Visualization and Computer Graphics* 14.4 (2008), 805–819 3.

[LCBK19] Lyon, Max, Campen, Marcel, Bommes, David, and Kobbelt, Leif. "Parametrization quantization with free boundaries for trimmed quad meshing". *ACM Transactions on Graphics (TOG)* 38.4 (2019), 1–14 2.

[Lev21] Levi, Zohar. "Direct seamless parametrization". *ACM Transactions on Graphics (TOG)* 40.1 (2021), 1–14 2.

[Liv19] Livesu, Marco. "cinolib: a generic programming header only C++ library for processing polygonal and polyhedral meshes". *Transactions on Computational Science XXXIV*. Springer, 2019, 64–76 5.

[Liv20] Livesu, Marco. "A Mesh Generation Perspective on Robust Mappings". *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association, 2020. isbn: 978-3-03868-124-3. doi: 10.2312/stag.20201234 7.

[Liv21] Livesu, Marco. "Scalable Mesh Refinement for Canonical Polygonal Schemas of Extremely High Genus Shapes". *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 27.1 (2021), 254–260. doi: 10.1109/TVCG.2020.3010736 3, 5.

[LPC22] Livesu, Marco, Pitzalis, Luca, and Cherchi, Gianmarco. "Optimal Dual Schemes for Adaptive Grid Based Hexmeshing". *ACM Transactions on Graphics* 41.2 (2022). doi: 10.1145/3494456 6.

[LYNF18] Liu, Ligang, Ye, Chunyang, Ni, Ruiqi, and Fu, Xiao-Ming. "Progressive parameterizations." *ACM Trans. Graph.* 37.4 (2018), 41–1 2, 7.

[MPZ14] Myles, Ashish, Pietroni, Nico, and Zorin, Denis. "Robust field-aligned global parametrization". *ACM Transactions on Graphics (TOG)* 33.4 (2014), 1–14 2.

[MZ12] Myles, Ashish and Zorin, Denis. "Global parametrization by incremental flattening". *ACM Transactions on Graphics (TOG)* 31.4 (2012), 1–11 2.

[MZ13] Myles, Ashish and Zorin, Denis. "Controlled-distortion constrained global parametrization". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 1–14 2.

[PCS*22] Pietroni, Nico, Campen, Marcel, Sheffer, Alla, et al. "Hex-Mesh Generation and Processing: a Survey". *ACM Transactions on Graphics (TOG)* (2022) 7.

[PZKW11] Peng, Chi-Han, Zhang, Eugene, Kobayashi, Yoshihiro, and Wonka, Peter. "Connectivity editing for quadrilateral meshes". *Proceedings of the 2011 SIGGRAPH Asia conference*. 2011, 1–12 6, 7.

[RLL*06] Ray, Nicolas, Li, Wan Chiu, Lévy, Bruno, et al. "Periodic global parameterization". *ACM Transactions on Graphics (TOG)* 25.4 (2006), 1460–1485 2.

[RPPS17] Rabinovich, Michael, Poranne, Roi, Panozzo, Daniele, and Sorkine-Hornung, Olga. "Scalable locally injective mappings". *ACM Transactions on Graphics (TOG)* 36.4 (2017), 1 2, 7.

[SJZP19] Shen, Hanxiao, Jiang, Zhongshi, Zorin, Denis, and Panozzo, Daniele. "Progressive embedding". *ACM Transactions on Graphics* 38.4 (2019) 2.

[SS15] Smith, Jason and Schaefer, Scott. "Bijective parameterization with free boundaries". *ACM Transactions on Graphics (TOG)* 34.4 (2015), 1–9 2, 7.

[SYLF20] Su, Jian-Ping, Ye, Chunyang, Liu, Ligang, and Fu, Xiao-Ming. "Efficient bijective parameterizations". *ACM Transactions on Graphics (TOG)* 39.4 (2020), 111–1 2, 7.

[SZC*22] Shen, Hanxiao, Zhu, Leyi, Capouellez, Ryan, et al. "Which cross fields can be quadrangulated? global parameterization from prescribed holonomy signatures". *ACM Transactions on Graphics (TOG)* 41.4 (2022), 1–12 2.

[TACD06] Tong, Yiying, Alliez, Pierre, Cohen-Steiner, David, and Desbrun, Mathieu. "Designing quadrangulations with discrete harmonic forms". *Eurographics symposium on geometry processing*. 2006 2.

[Tar22] Tarini, Marco. "Closed-form quadrangulation of n-sided patches". *Computers & Graphics* 107 (2022), 60–65 2.

[TPC*10] Tarini, Marco, Pietroni, Nico, Cignoni, Paolo, et al. "Practical quad mesh simplification". *Computer Graphics Forum*. Vol. 29. 2. Wiley Online Library. 2010, 407–418 7.

[TPP*11] Tarini, Marco, Puppo, Enrico, Panozzo, Daniele, et al. "Simple quad domains for field aligned mesh parametrization". *Proceedings of the 2011 SIGGRAPH Asia Conference*. 2011, 1–12 7.

[TPS14] Takayama, Kenshi, Panozzo, Daniele, and Sorkine-Hornung, Olga. "Pattern-Based Quadrangulation for N-Sided Patches". *Computer Graphics Forum*. Vol. 33. 5. Wiley Online Library. 2014, 177–184 2.

[Tut63] Tutte, William Thomas. "How to draw a graph". *Proceedings of the London Mathematical Society* 3.1 (1963), 743–767 2, 3.

[ULP*15] Usai, Francesco, Livesu, Marco, Puppo, Enrico, et al. "Extraction of the quad layout of a triangle mesh guided by its curve skeleton". *ACM Transactions on Graphics (TOG)* 35.1 (2015), 1–13 7.

[WHL*08] Wang, Hongyu, He, Ying, Li, Xin, et al. "Polycube splines". *Computer-Aided Design* 40.6 (2008), 721–733 3.

[ZTZC20] Zhou, Jiaran, Tu, Changhe, Zorin, Denis, and Campen, Marcel. "Combinatorial construction of seamless parameter domains". *Computer Graphics Forum*. Vol. 39. 2. Wiley Online Library. 2020, 179–190 2.