

1 Reflectional Symmetry Field Processing for Non-Orientable Surfaces 2

3 ANONYMOUS AUTHOR(S)
4 SUBMISSION ID: PAPERS_600S2
5



33 Fig. 1. A number of non-orientable surfaces generated using our system.
34

35 Many graphics applications, such as texture and geometry synthesis,
36 quad-
37 rangular and triangular remeshing, and pen-and-ink sketching of surfaces,
38 make use of directional field to guide the orientation of texture and geometry
39 patterns, the edges of the meshes, and hatches, respectively. The quality
40 of the underlying direction field directly impacts the quality of application
41 results. Existing field processing research has focused on rotational sym-
42 metry fields, with the assumption that the underlying surface is *orientable*.
43 However, many models are non-orientable due to topological noise from
44 the mesh generation process. In addition, non-orientable surfaces such as
45 Klein bottles can be interesting objects for texture synthesis and pen-and-ink
46 illustration. For non-orientable surfaces, the direction fields need to respect
47 reflection symmetries.

48 In this paper, we investigate the processing of N -way reflectional sym-
49 metry (N -ReSy) fields on surfaces, including the representation of ReSy fields
50 on surfaces, their topological analysis, design, and automatic feature-aware
51

52 Permission to make digital or hard copies of all or part of this work for personal or
53 classroom use is granted without fee provided that copies are not made or distributed
54 for profit or commercial advantage and that copies bear this notice and the full citation
55 on the first page. Copyrights for components of this work owned by others than ACM
56 must be honored. Abstracting with credit is permitted. To copy otherwise, or republish,
57 to post on servers or to redistribute to lists, requires prior specific permission and/or a
58 fee. Request permissions from permissions@acm.org.

59 © 2020 Association for Computing Machinery.
60 0730-0301/2020/5-ART \$15.00
61 <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

62 generation, as well as global parameterization from ReSy fields and their
63 applications in texture and geometry synthesis, remeshing, and pen-and-ink
64 illustration of surfaces. Our technique works for non-orientable surfaces as
65 well as orientable surfaces that have inconsistent normal (i.e. non-oriented
66 but orientable).

67 In addition to our ReSy field processing pipeline, we also provide tools
68 to design non-orientable surfaces with more complicated geometry than
69 the textbook examples such as the real projective surface and the Kline
70 bottle. Closed non-orientable surfaces necessarily have self-intersections.
71 We provide the ability to control where self-intersections can occur in a
72 surface, which is useful in creating animations to demonstrate fundamental
73 properties regarding non-orientable surfaces, such as attaching two cross-
74 caps to a surface is equivalent to attaching a cross-handle.

75 Creating global parameterization of surfaces with respect to direction
76 fields often implicitly makes use of the *branched covering spaces* (BCSs) of the
77 surfaces. In this paper, we adapt the visualization of BCSs from orientable
78 surfaces to non-orientable surfaces. In addition, inspired by the existence of
79 reflectional symmetry, we introduce a novel visualization of BCSs by using
80 only pairwise connection between the layers in the BCSs, thus avoiding the
81 need for multiple layers intersecting with a complicated fashion.

82 CCS Concepts: • Human-centered computing → Visualization; • Com-
83 puting methodologies → Shape modeling.

Additional Key Words and Phrases: Non-orientable surfaces, reflectional symmetries, reflectional symmetry field design, topology, branched covering spaces visualization

ACM Reference Format:

Anonymous Author(s). 2020. Reflectional Symmetry Field Processing for Non-Orientable Surfaces. *ACM Trans. Graph.* 1, 1 (May 2020), 13 pages. <https://doi.org/10.1145/nnnnnnnnnnnnnnn>

1 INTRODUCTION

Many graphics applications make use of a directional field. For example, in pen-and-ink sketching of surfaces [Hertzmann and Zorin 2000], the hatches usually follow principal curvature directions to convey the bending in the surface. Similarly, when quadrangulating a surface, the edges in the quad meshes are usually required to also follow principal curvature directions in the surface [Alliez et al. 2003]. In pattern synthesis on surfaces [Wei et al. 2009], texture and geometry patterns with patterns of certain symmetries are synthesized over the surface such that the patterns follow natural directions. In all of these applications, quality of the underlying directional field has a direct impact on the quality of the subsequent application results.

Most existing algorithms for these applications assume that the underlying surfaces are orientable [Vaxman et al. 2017], i.e., it is possible to consistently compute the outward normal on the surface. However, non-orientable objects, for which consistent normal orientation is impossible, can also be interesting objects for geometry processing application. For example, Hertzmann and Zorin demonstrate their pen-and-ink algorithm on the Klein bottle, a non-orientable object that is not only an important example in the study of topology but also the source of inspiration for many artworks []. Non-orientable objects have also been used in plain-weaving [Akleman et al. 2009]. Such objects have also found applications in tensor field topology. For example, In addition, Roy et al. [2018a] demonstrate that an important topological feature in 3D linear symmetric tensor fields, referred to as *neutral surfaces*, is homeomorphic to attaching a handle to (\mathbb{RP}^2) , thus non-orientable. In addition, non-orientable surfaces can arise due to topological noise in the modeling process [?]. While it is possible to fix the topological error, the process can be tedious. Being able to process non-orientable surfaces directly can overcome the aforementioned limitations.

However, non-orientable surfaces present some unique challenge. Due to the lack of a globally consistent surface orientation, rotationally symmetry (RoSy) fields, which have been used extensively for the above geometry processing applications, become inadequate as visual artifacts appear where orientation flips occur (Figure ??). A new type of directional field is needed to extend geometry processing applications from orientable surfaces to non-orientable surfaces.

In this paper, we introduce the notion of *reflectional symmetry* (ReSy) fields and demonstrate that ReSy fields are adequate for pen-and-ink, texture and geometry synthesis, and remeshing of non-orientable surfaces. We provide a representation for ReSy fields for non-orientable surfaces represented as triangular meshes. This representation enables efficient topological analysis for ReSy fields, such as extracting the singularities and separatrices in the field and

cut graphs along which the underlying non-orientable surface can be cut into a topological disk.

Furthermore, we adapt existing algorithms from RoSy field processing to ReSy fields, including ReSy field design, feature-aware ReSy field generation, and global parameterization from a ReSy field. We also provide tools to enable the editing of the connectivity in the cut graphs and the locations of the *seams* where normal flips occur. The latter allows us to generate animations for a number important mathematical properties for non-orientable surfaces, such as that attaching a handle to a non-orientable surface is equivalent to adding a cross-handle to the same surface.

Moreover, we provide algorithms to visualize the *branched covering spaces* (BCSs) of non-orientable surfaces [Armstrong 1979]. In particular, inspired by the reflectivity in ReSy fields, we provide a novel visualization method in which different layers in the BCS can be connected pairwise, as opposed to having multiple layers connected through another object called *docking station* [Roy et al. 2018b]. In the new visualization, i.e. pairwise layer connection, the connectivity among the layers can be better demonstrated than using docking stations (Figure ??).

In addition to the functionalities related to ReSy field processing, our system also enables the modeling of non-orientable surfaces. Non-orientable patches such as cross-caps and cross-handles can be added to or removed from an interface. An object can be knotted, while two objects can be linked. We also provide the ability to modify the geometry of the geometry such as the self-intersections in a non-orientable surface matches where the seams (normal flips) are.

The remainder of the paper is organized as follows. In Section 2 we review relevant past research. We provide a review of mathematical concepts and properties that are most relevant to this research in Section 3. We then describe our representation and processing capabilities of non-orientable surfaces in Section 5, our approach to the design, processing, and application results of reflectional symmetry fields in Section 4, and our novel construction of BCSs for non-orientable surfaces in Section 6. We provide analysis of the performance of our system in Section 7 before concluding in Section 8.

2 RELATED WORK

There has been much work on the analysis and processing of rotational symmetry fields. To review all of this work is beyond the scope of this paper. Instead, we refer to our reader to the survey by Vaxman et al. [2017]. Here, we will review work most relevant to our research.

Quadrangular remeshing of surfaces has been one of driving forces behind recent advances in geometry processing since its introduction into the graphics community [Alliez et al. 2003]. To avoid T-junctions in the mesh, Ray et al. [2006] build a global parameterization of the input mesh that admits quadrangular grid structures. The structure is guided by a four-way rotational symmetry field. Palacios and Zhang [2007] generalize the four-way rotational symmetries on surfaces to N -way rotational symmetries, which they refer to as N -RoSy's. They also provide topological analysis and design capabilities for N -RoSy fields on orientable surfaces. Kälberer et al. [2007] produce to our knowledge the first pure quad meshing

method by using the idea of lifting the guiding rotational symmetry field to a vector field in the branched cover of the input surface and performing Hodge-decomposition on the lifted vector field to make it curl-free. Bommes et al. [2009] introduce a different formulation for generating the global parameterization from the rotational symmetry field based on mixed-integer programming. Campen et al. [Campen et al. 2012] produce a partition of the surface using the notion of dual loops, which leads to controlled high-quality quadrangulation. Zhang et al. [2010] provide a formulation for quad remeshing using the idea of waves. Bommes et al. [2013] provide formulations to guarantee injectivity of the parameterization. Rabinovich et al. [2017] address the local injectivity issue using a novel energy function. The idea of rotational symmetric fields has also been applied to triangular remeshing of surfaces with control over the number of irregular vertices in the mesh [Nieser et al. 2012].

Besides remeshing, rotational symmetry fields have also found applications in other computer graphics applications, such as texture and geometry synthesis [Nieser et al. 2012; Wei et al. 2009], painterly rendering [?], pen-and-ink sketching of surfaces [Hertzmann and Zorin 2000], and generating plain-weaving patterns on surfaces [Akleman et al. 2009].

To our knowledge, there has been relatively little mention of reflectional symmetry fields, which are needed to extend the aforementioned geometry processing applications to non-orientable surfaces. This is the focus of our research, as we provide the analysis and processing of reflectional symmetry fields on orientable and non-orientable surfaces.

The topology of non-orientable surfaces is well-understood [Armstrong 1979]. In graphics, Hertzmann and Zorin [2000] demonstrate their pen-and-ink approach with a Klein bottle. Akleman et al. [2009] generate plain-weaving patterns on surfaces through edge twisting, which can lead to non-orientable surfaces. Note that their input surfaces are orientable surfaces. In general, most geometry processing application assume orientable surfaces as an input. However, non-orientable surfaces can result due to topological errors during the modeling process. While such models can be turned into orientable surfaces through topological surgery, we aim to allow geometry processing on these models without having to perform the surgeries. In addition, much art has been produced about non-orientable surfaces. We aim to extend geometry processing applications to non-orientable surfaces.

Branched covering spaces are an important mathematical concept that has found applications in surface parameterization [Campen et al. 2012; Kälberer et al. 2007; Nieser et al. 2012]. In addition, it has been used in tensor field visualization [Tricoche 2002]. However, branched covering spaces are not explicitly constructed in all this work. A number of researchers provide algorithms to construct branched covering spaces for the complex plane (topologically a sphere) [Bátorová et al. 2013; Nieser et al. 2010; Valíková and Chalmovianský 2015]. Roy et al. [2018b] construct branched covering spaces for orientable surfaces of arbitrary genus with a guiding rotational symmetry field defined on it. To reduce the amount of self-intersection, they move the layers in the branched covering spaces away from each other and use tubes to connect them. While this approach makes the global topology more clear, in more complicated cases the tubes from multiple layers must be connected to each

other in a twisted and self-intersecting fashion. In this paper, we introduce a novel visualization to remove the need for connecting multiple tubes, thus only have tubes that connect two layers.

3 MATH BACKGROUND

In this section, we review mathematical concepts and properties most relevant to our work.

Surface Orientability and Classification: A manifold surface is *orientable* if it is possible to consistently choose a “clockwise” orientation for all loops in the manifold [Armstrong 1979]. Otherwise, the surface is *non-orientable*. For surfaces in \mathbb{R}^3 , the orientability property is equivalent whether the surface has one or two sides. An orientable surface has two sides, such as a disk, a sphere, and a torus. A non-orientable surface has only one side, such as a Möbius band, the real projective space \mathbb{RP}^2 , and the Klein bottle. The orientability property is also related to whether a consistent choice of surface normal can be made for the whole surface. For orientable surfaces, it is possible to find such a consistent normal assignment, while for non-orientable surfaces it is not possible. Note that a surface is always *locally orientable*, i.e. given a point in the surface it is possible to find a neighborhood of the point inside which the surface is orientable.

The topology of a surface can be modified through a number of surgeries, such as cutting out a disk in the interior of the surface, which increases the number of boundaries in the surface by one. It is also possible to *remove* the boundaries in the surface through gluing. There are four commonly used operations: (1) adding a cap (the reverse of cutting out a disk), (2) adding a cross-cap, (3) adding a handle, and (4) adding a cross-handle. Note that these four operations can be realized through closing one or two boundaries of the surface through stitching.

We need a figure to show these four things

Adding a cap is equivalent to closing a boundary by stitching the left half to the right half in the same orientation. Adding a cross-cap is done in a similar fashion, except that the two halves of the boundary are stitched in opposing orientations. Adding a handle can be realized by stitching two boundaries together in a consistent fashion, while adding a cross-handle can be realized by also stitching two boundaries together but in a twisted fashion, with reversed orientation. The existence of a cross-cap or a cross-handle indicates the surface is non-orientable.

The *Euler characteristic* of a manifold surface is an intrinsic measure of a surface. For a surface S with only one connected component, its Euler characteristic $\chi(S) = 2 - b(S) - cp(S) - 2h(S) - 2ch(S)$ where $b(S)$, $cp(S)$, $h(S)$, and $ch(S)$ are the number of boundaries, the number of cross-caps, the number of handles, and the number of cross-handles in the surface, respectively. The surface is orientable if $cp(S) = ch(S) = 0$. Otherwise, it is non-orientable. For non-orientable surfaces, both a handle and a cross-handle can be replaced with two cross-caps without changing the topology.

Consequently, closed orientable surfaces can be classified based on the number of handles in the surface, and their Euler characteristics become $2 - 2h(S)$. In contrast, non-orientable surfaces can be classified based on the number cross-caps in the surfaces, and their Euler characteristics become $2 - cp(S)$. For example, the Klein

bottle (non-orientable) and the torus (orientable) both have a zero Euler characteristic. A sphere (orientable) and the real projective space have a Euler characteristic of two and one, respectively.

Rotational Symmetries on Surfaces: An N -way rotational symmetry (N -RoSy) [Palacios and Zhang 2007] is a collection of N vectors of the same length that are evenly-spaced angularly. Each vector in the collection is referred to as a *member vector*. Another way of considering an N -RoSy is to extend each member vector into a right-handed orthogonal basis, which we refer to as a *member basis* of the N -RoSy. There are N rotations, namely, $R_{\frac{2k\pi}{N}}$ ($0 \leq k < N$), that induce a cyclic permutation of the N member bases. These rotations form a group that is isomorphic to \mathbb{Z}_N , the *cyclic group* of order N .

Under an orthonormal basis, an N -RoSy can be represented by a vector (referred to the representation vector [Palacios and Zhang 2007]) whose length is the same as that of the member vectors of the N -RoSy and whose angular component is N times that of any of the member vectors (modulo 2π).

A *zero* N -RoSy satisfies that all member vectors have a length of zero. In this case, its representation vector is also zero.

An N -RoSy field on an orientable surface S is a continuous N -RoSy-valued function. A *singularity* in an N -RoSy field is a point in S where the N -RoSy field is zero. The *index* of the singularity is $\frac{1}{N}$ of the index of the same point as a singularity in the representation vector field. Given an N -RoSy field with only isolated singularities, the total index of all singularities in the field is equal to that of the Euler characteristic of the underlying surface.

A *streamline* in an N -RoSy field is a smooth curve in the field that is everywhere tangent to one of member vector directions in the field. A *separatrix* is a streamline that approaches a singularity. When N is an odd integer, a separatrix at a singularity can be further classified as either an *incoming separatrix* (approaching the singularity) or an *outgoing separatrix* (leaving the singularity).

Cyclic and Dihedral Groups: As mentioned earlier, the cyclic groups \mathbb{Z}_N can be used to model the rotations that maintain a regular N -gon. There are a total of N elements in \mathbb{Z}_N , including the identity.

Besides these rotations, there are N reflections in the plane that can also maintain the appearance of the regular N -gon. Together, the $2N$ element form the *dihedral group* of order N , denoted by \mathbb{D}_N . While the rotations also maintain the orientation of the N -gon, the reflections reverse it. The *order* of a reflection is two as performing the reflection twice results in the identity. The product of a rotation and a reflection in the dihedral group is a reflection. That is, inside the dihedral group, the product of a rotation and an order-two element also has order two.

The dihedral groups play a role for reflective symmetries similar to that of the cyclic groups for rotational symmetries.

Branched Covering Spaces: A branched covering space (BCS) extend the notion of *covering spaces* [Armstrong 1979] by allowing ramification points. More specifically, Let X and C be two topological spaces and $p : C \rightarrow X$ be continuous surjective map. C is said to be a **branched covering space** of X under p if there exists a nowhere dense set $\Delta \subset X$ such that $p|_{p^{-1}(X \setminus \Delta)} : p^{-1}(X \setminus \Delta) \rightarrow X \setminus \Delta$ is a covering mapping. The set $X \setminus \Delta$ is a **regular set** of the branched covering p , whereas Δ is the **singular set**. For example, \mathbb{R} covers

the set of non-negative numbers \mathbb{R}^+ under the map: $p : \mathbb{R} \rightarrow \mathbb{R}^+$ ($p(x) = |x|$). The singular set consists of the number 0 which has only one pre-image under p while other elements in \mathbb{R}^+ have two pre-images.

The branched covering mappings can be induced from an input N -RoSy field on some base surface S . Given an input surface S and an N -RoSy field defined on it, a branched covering space can be constructed that lifts the N -RoSy field to a vector field on the BCS in the following sense: (1) every regular point p in S corresponds to N points in the BCS, and (2) the collection of the vectors at points in the BCS that correspond to p is exactly the N -RoSy at p . The singularities in the N -RoSy field correspond to ramification points in the BCS, where the map between the base surface and the BCS has fewer-than- N pre-images.

One of the most important results is the *Riemann-Hurwitz formula* [Hartshorne 1977], which relates the topology of a surface to that of its branched cover.

4 REFLECTIONAL SYMMETRY FIELD PROCESSING

In this section, we describe our technique and system for the processing of reflectional symmetry fields on surfaces.

On a non-orientable surface, an N -RoSy field will not be continuous across the seams. More specifically, given an edge e in the mesh whose adjacent triangles have opposite vertex orderings, the RoSy's in the two triangles are no longer compatible as shown in Figure 2. To overcome this difficulty, we introduce the notion of reflection symmetries.

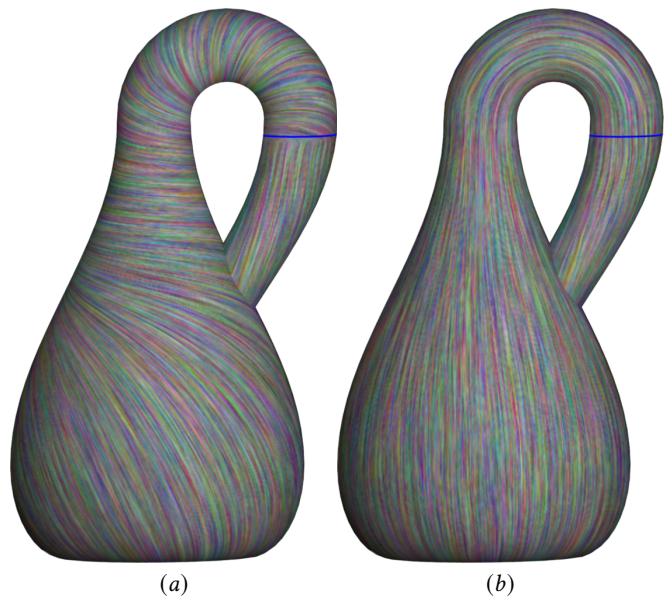


Fig. 2. This figure illustrates the incompatibility of an N -RoSy field across edges whose adjacent triangles have opposite *vertex orderings*. The 2-RoSy field designed on a Klein bottle is not continuous across the seam (blue curve) in (a). Using a 2-ReSy field in (b) overcomes this issue and the field is continuous even across seams.

457 An N -way reflection symmetry, which we abbreviate as N -ReSy,
 458 is a collection of $2N$ orthogonal bases: half of which are right-
 459 handed, and the other half left-handed. In fact, the right-handed
 460 member bases form an N -RoSy. There is a bijective map between the
 461 set of right-handed member bases and the set of left-handed member
 462 bases as follows: negating the Y -axis of a right-handed member basis
 463 results in a left-handed member basis, and vice versa. In fact, the
 464 left-handed member bases become right-handed orthogonal bases
 465 if the normal to the polygon is negated.

466 For the ease of subsequent processing, we represent a member
 467 frame of an N -ReSy as a vector representing its X -axis, similar to
 468 N -RoSy's, along with a choice of the vertex ordering.

469 An N -ReSy field is a continuous N -ReSy-valued function on the
 470 surface. Note that any non-orientable surface is *locally orientable*,
 471 i.e., given a point in the surface, there exists a neighborhood of the
 472 point so that it is possible to assign a consistent vertex ordering to
 473 all points in the neighborhood.

474 This means that if we choose a subsurface that is orientable, we
 475 can locally modify the normal to make it consistent in the subsurface,
 476 thus turning N -ReSy processing into N -RoSy processing.

477 For example, our N -ReSy fields are defined by assigning an N -
 478 ReSy at each vertex of the underlying triangular mesh. The N -ReSy
 479 values at any point on the edges or the triangles in the mesh are
 480 obtained by interpolating the N -ReSy values at the vertices of the
 481 edge or triangle. Since a triangle is locally orientable, when inter-
 482 polating the vertices' N -ReSy values, we can reassign the normal of
 483 some of the relevant vertices to make all the normals consist-
 484 ent. Consequently, N -ReSy interpolation is converted into N -RoSy
 485 interpolation.

486 A zero N -ReSy is one where its corresponding N -RoSy is zero. A
 487 singularity in an N -ReSy field is a point in the domain where the
 488 N -ReSy is zero. In our computational setup, a singularity occurs
 489 inside a triangle, which is locally orientable. Singularities can be
 490 extracted using similar algorithms from N -RoSy fields once the
 491 normal at the vertices of the triangle is made consistent.

492 In addition, the local N -ReSy pattern around the singularity re-
 493 sembles that around a singularity in an N -RoSy field. That is, a
 494 N -ReSy singularity can be measured by its singularity index, a mul-
 495 tiple of $\frac{1}{N}$ that is based on the winding number of the singularity.
 496 The sum of the singularity indexes of all singularities in an N -ReSy
 497 field is equal to the Euler characteristic of the underlying surface.

498 The notion of separatrices can also be adapted from N -Rosy
 499 fields to N -ReSy fields. Figure 3 shows an N -ReSy field on a surface.
 500 The colored dots represent singularities, while the colored curves
 501 represent separatrices. The member vectors in the N -ReSy field is
 502 visualized using LIC [Palacios and Zhang 2011].

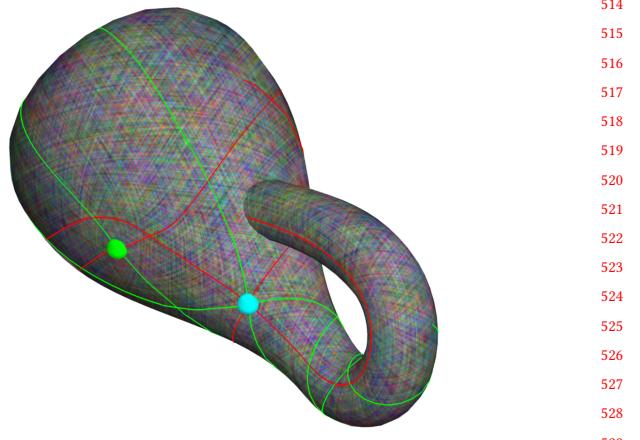


Fig. 3. This figure illustrates the topology of a 3-ReSy field on the Klein bottle. There are five positive singularities (green dots) and five negative singularities (blue dots). The outgoing and incoming separatrices are shown with the red and the green curves, respectively.

4.1 Representation

We start our description with the representation of ReSy fields. Since the underlying surface can be non-orientable, we also need to take into account this non-orientability.

Most existing research in geometry processing focuses on orientable surfaces, which are usually represented by triangular meshes with consistent vertex orderings between adjacent triangles. Such ordering can also lead to consistent surface normal. To facilitate geometry processing tasks such as surface traversal and curvature computation, an additional data structure called *corners* [Rossignac 1999] is also used. Figure 4 illustrates the concepts with a simple mesh. Each triangle has three corners. A corner has three important operators: (1) $.p$, (2) $.n$, and (3) $.o$. All of these operators refer to another corner. For example, $c.n$ is the corner in the same triangle that comes after corner c according to the vertex ordering of the triangle, $c.p$ is the one from the same triangle that proceeds c . For example, in Figure 4, $c_3.n = c_4$ and $c_3.p = c_5$. The $.o$ operator refers to the corner in an adjacent triangle that points at the same edge as c . In Figure 4, $c_4.o = c_0$ and $c_3.o = \text{NULL}$ as there is no adjacent triangle facing corner c_3 .

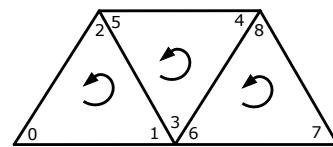


Fig. 4. This figures illustrates the corner data structure [Rossignac 1999] for orientable surfaces.

For non-orientable surfaces such as Möbius strips and Klein bottles, the aforementioned globally consistent vertex orderings do not

exist. To address this challenge and adapt the above geometry processing tasks to non-orientable surfaces, we develop the following representation.

A surface is again represented as a triangular mesh, with potentially non-compatible vertex orderings between neighboring triangles. An edge whose incident triangles have non-compatible vertex orderings is considered as a *seam edge*. The set of all seam edges is the *seam* of the mesh. An orientable surface represented as a mesh with a globally consistent vertex order has an empty seam.

In addition, each triangle has three corner pairs. Each corner pair corresponds to one corner in the original data structure: one for the clockwise vertex order inside the triangle, and one for the counterclockwise vertex order. As an example, in Figure 5 (left), c_0 and c_1 (c omitted in the figure for legibility) form a corner pair that together corresponds a vertex in the triangle. In addition, corners c_0, c_2 , and c_4 correspond to one vertex ordering of the triangle while c_1, c_5 , and c_3 correspond to the other. Two triangles with the same orientation has a compatible edge (colored in green) such as the one between the middle and right triangles in Figure 5 (left). In this case, the corners in the triangles behave as in the case of orientable surfaces. That is, the first corners in the pairs from the middle triangle c_8, c_{10} , and c_6 are related to the first corners in the pairs from the right triangle c_{12}, c_{14} , and c_{16} . Similarly, the second triangles in the triangle pairs from the middle and right triangles also are related. On the other hand, between the left and middle triangles, the vertex orders are incompatible. Consequently, the first corners in the left triangle corresponds to the second corners in the middle triangle, and vice versa.

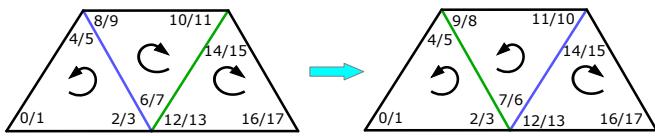


Fig. 5. Seam editing operation performed on the middle triangle, reverses the vertex ordering of that triangle. This makes the seam and the non-seam edges (marked in blue and green, respectively in the left image) become a non-seam and a seam edge as can be seen in the image on the right (marked in green and blue, respectively).

4.2 Topological Analysis

4.3 Design and Topological Editing

Our system allows the user to create N -ReSy fields on orientable and non-orientable surfaces either from scratch or based on some input N -ReSy fields.

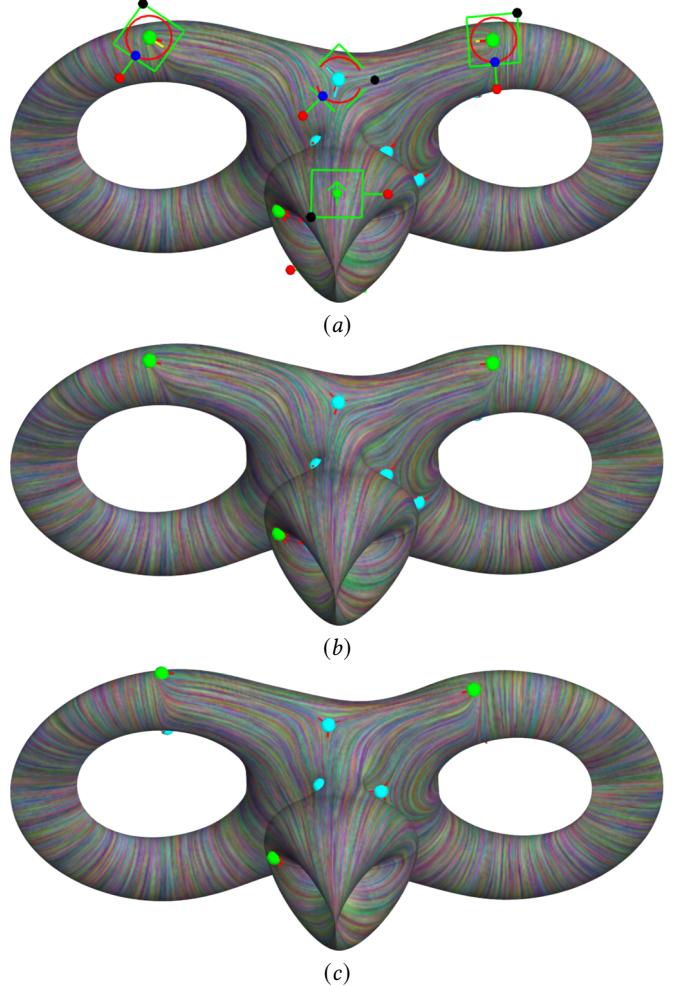


Fig. 6. Our design system of a 2-ReSy field on a non-orientable surface (with an Euler characteristic of -3) generated by attaching two pinched torus and one Möbius band together: (a) the user can place and orient regular element (bottom), positive element (top-left and top-right), and negative element (top-middle). This results in nine singularities of index $+\frac{1}{2}$ (green dots) and fifteen singularities of index $-\frac{1}{2}$ (blue dots), leading to a total singularity index of -3 matching the Euler characteristic of the surface as shown in (b). The user has also the option of smoothing the field globally resulting into a different set of singularities (eight with index $+\frac{1}{2}$ and fourteen with index $-\frac{1}{2}$) as shown in (c).

Similar to [Palacios and Zhang 2007], the user can provide a set of constraints in the forms of a desired N -ReSy value at a given point (a regular element) and a desired singularity pattern at the point of interest (a singular element). Each of the constraint is then converted into N -ReSy values at the vertices of the triangle containing the location of the design element. These values are then used as the boundary **conditions** of a Laplace equation which is a system of linear equations that can be solved efficiently. The solution to the system of equations gives rise to the N -ReSy values at the free

vertices, i.e. where constraints were not given. Figure 6 illustrates an example of our ReSy field design system.

To compute the Laplace at a vertex, one needs to parallel transport the N -ReSy value from an adjacent vertex to the tangent plane of the vertex of interest. Since the two triangles sharing the edge are locally orientable, we can convert the problem of parallel transporting N -ReSy's to that of parallel transporting N -RoSy's as illustrated in Figure 7. Note that in this case vector directions do not change when crossing the seams but the local frames in the tangent planes need to be reflected. Consequently, the aforementioned system of linear equations can be set up properly.

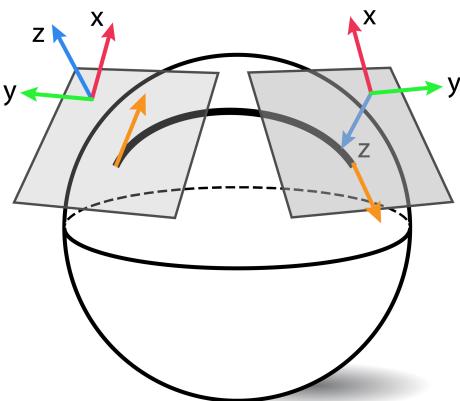


Fig. 7. The figure shows a parallel transport of the given vector across an orientability seam. Starting from the vector (shown in yellow) and a frame on the left tangent plane, the frame flips across a seam edge as shown on the right tangent plane. The vector, however, remains continuous over the geodesic curve (thick black) as shown on the right tangent plane.

Given the similarity between the topology of an N -ReSy field and that of an N -RoSy field, we adapt the topological editing operations of [Palacios and Zhang 2007] to N -ReSy fields. There are two such operations: *singularity movement*, and *singularity pair cancellation*. The former allows to relocate a singularity to a more desired location with changing the number of singularities in the field, while the latter reduces the total number of singularities in the field by two. Neither operation changes the sum of the total singularity index, which must match the Euler characteristic of the underlying surface.

Both operations require identifying a neighborhood that contains the singularity (to be moved) or the singularities (to be canceled). For *N*-RoSy fields on orientable surfaces, Palacios and Zhang [Palacios and Zhang 2007] achieve this by constructing a surface parameterization called polar maps. The polar map enables the topological editing of the *N*-RoSy field into that of the representation vector field, for which the regions surrounding the singularities can be found using Conley index theory [Zhang et al. 2006]. Once the region is constructed, a vector-valued Laplace equation is set up with the boundary condition ensuring the desired effect, i.e. moving the singularity to the new location or removing a singularity pair. The resulting vector field is then converted back to an *N*-RoSy field using the same surface parameterization.

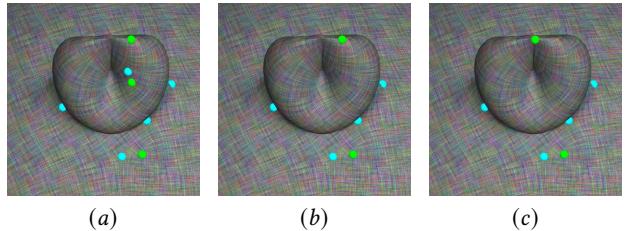


Fig. 8. The figure illustrates the topological editing of a 4-ReSy field on a non-orientable surface generated by attaching a Möbius band on sphere. The fields are: a 4-ReSy field with eight singularities (3 positive and 5 negative) (a), after canceling a pair of singularities (b), and after singularity movement (c).

N-ReSy editing operations can be achieved through a similar framework. The only place that requires care is when converting the *N*-ReSy field to an *N*-RoSy field. On a non-orientable surface, it is impossible to convert an *N*-ReSy field into an *N*-RoSy field. However, since we are only concerned with the local orientability around the singularities of interest, we can choose the polar map and normal assignment such that the *N*-ReSy field is locally orientable. That is, we can push the seams to places far away from the region. Figure 8 demonstrates the capability of our ReSy field editing system.

4.4 Geometry-Aware Field Generation

In addition to manual design, our system also allows feature-aware automatic generation of N -ReSy fields when $N = 4$ or $N = 6$.

The major and minor principal curvature directions are mutually perpendicular except at the umbilical points. Consequently, the principal curvature directions can be treated as the member vectors of a 4-ReSy. The resulting 4-ReSy field can be used for quadrangular remeshing. Meshes generated this way have their edges naturally aligned with the ridges and valleys in the underlying surface.

Similarly, the principal curvature directions can also be used to guide triangular remeshing [Nieser et al. 2012], which can make use of a 6-ReSy field. However, in this case, only one of the principal curvature directions can be used to generate the 6-ReSy field. Nieser et al. [Nieser et al. 2012] extract nearly cylindrical regions in the surface and align the 6-ReSy field with either the major principal curvature direction or the minor principal curvature direction, depending on whether the surface is locally convex or concave. These values are then propagated to the whole surface using the aforementioned relaxation process.

In addition, the number of singularities on the surface can be automatically reduced based on some noise measurement.

Our feature-aware *N*-ReSy field generation follows closely that of [Nieser et al. 2012]. Since the surface is locally orientable, curvature computation, cylindrical region identification, and field generation can be done using a similar framework.

4.5 Global Parameterization

An N -ReSy field ($N = 4, 6$) can be used for texture and geometry synthesis on surfaces as well as quadrangular and triangular remeshing.

To achieve this, it is essential to generate a global parameterization of the surface based on the N -ReSy field.

Recall that we represent an N -ReSy field as piecewise N -RoSy fields on the surface such that across seams there is a flip of surface normal. Consequently, we compute the parameterization for each patch with the same mixed-integer constraints, and ensure the continuity of the parameterization across patch boundaries by ensuring the following additional conditions of the (u, v) parameters on the both sides of a given edge:

$$(u', v') = \text{Rot}^i \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^l (u, v) + (j, k) \quad (1)$$

with integer coefficients (j, k) , i the gap across the edge and l being 1 or 0 depending on whether the edge is an orientability seam or not.

Once the **parameterization** is available, we can apply textures of compatible types, such as those respecting four-fold or six-fold reflectional symmetries.

In addition, we can perform quadrangular and triangular remeshing using the same parameterization. Figure 9 shows the results of performing pattern synthesis on the bunny (with **cross-handle** ears) and the dragon (with a Klein bottle horn).

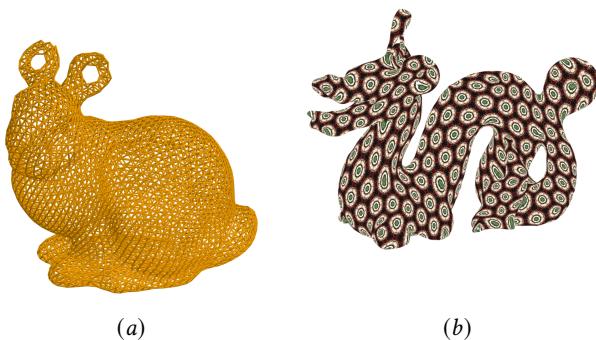


Fig. 9. **Geometry synthesis of a 6-ReSy field on bunny with two cross-handles attached (a), and texture synthesis of a 6-ReSy field on a dragon with one Klein bottle and three cross-caps attached (b).**

5 NON-ORIENTABLE SURFACE DESIGN

We now describe how non-orientable surfaces can be created and modified using our system.

5.1 Non-Orientable Surface Generation through Topological Surgeries

Our system allows a surface to be modified through *part replacement*. The user can draw a loop on the surface **that can separate the surface into two parts when cut along it**. The user can then select one part of the surface to be replaced, through this loop, with one of the following canonical shapes (Figure 10): (upper-left) a disk, (lower-left) a handle, (upper-middle) a cross-cap, (lower-middle) a Möbius band, (upper-right) a cross-handle, and (lower-right) the Klein bottle.

These canonical parts are generated from mathematical formations [Ferréol 2017; Ferréol and Mandonnet 2017]. Figure 11 illustrates the generation of a non-orientable surface by performing surface modifications using our design system.

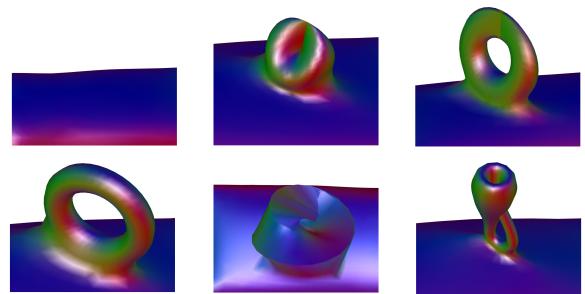


Fig. 10. These are the previews of the attachment available in our design system. Left column: A topological disk (top), and a Topological handle (bottom). Middle column: A cross-cap (top), and a Möbius band (bottom). Third Column: A cross-handle (top), and the Klein bottle (bottom).

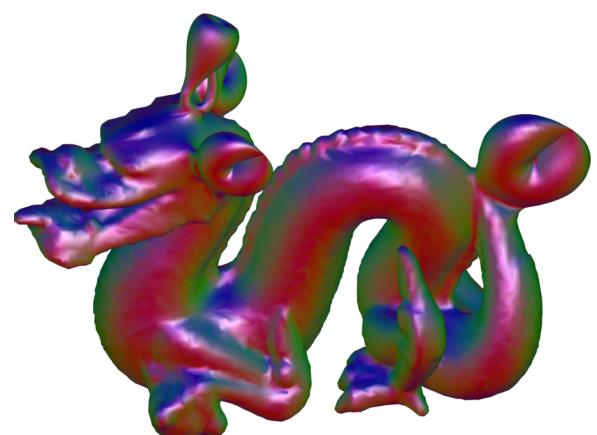


Fig. 11. A non-orientable surface generated by our design system by attaching a Klein bottle replacing the horn of a dragon, and attaching three cross-caps (two replacing the **ears** and one on the tail).

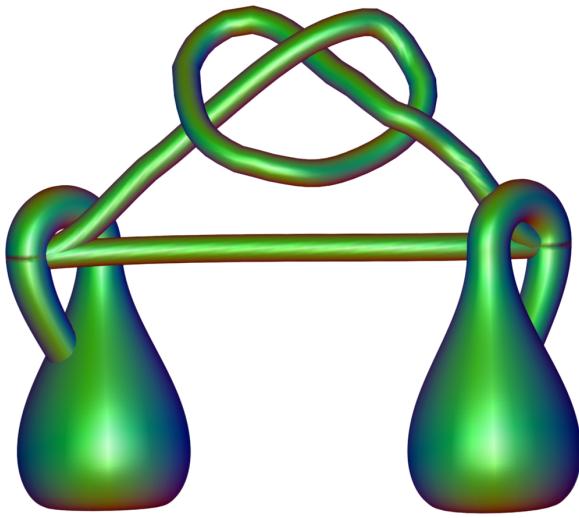


Fig. 12. This is a 2-fold BCS of a Klein bottle. The Euler characteristic of this surface with a knot is zero and is a topological torus.

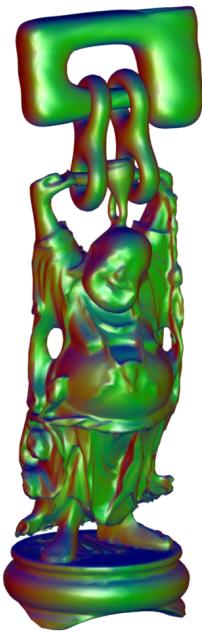


Fig. 13. A link generated by our system by linking two double torus, and a Klein bottle through the hands of the Buddha.

5.3 Proofs for Equivalence among Cross-Caps, Handles, and Cross-Handles

Closed non-orientable surfaces cannot be embedded in \mathbb{R}^3 , they can only be immersed, i.e., with self-intersection. This makes a number of mathematical properties of non-orientable surfaces difficult to see, such as the following:

- (1) Attaching a cross-handle to a surface is equivalent to adding two cross-caps.
- (2) Attaching a handle to a non-orientable surface is equivalent to adding a cross-handle.

While there are illustrations for these facts, they all involve cutting and re-gluing of the surface. This can make it difficult to keep track of the correspondence between the two surfaces, e.g. adding a cross-handle and adding two cross-caps. In this work, we seek to provide animations between the two surfaces without cutting and gluing.

We provide animations for the above statements without the need for cutting and gluing. The main idea behind our approach is the following: when performing Laplacian smoothing of a non-orientable surface, the pinch lines in the surface tend to coincide with the seam edges in the mesh. Therefore, when starting with a topological non-orientable surface with one geometric immersion (such as the two connected Möbius bands in Figure 14 (a)), if one can change the location and connectivity of the seam edges (Figure 14 (b)) and then perform Laplacian smoothing on the resulting mesh, it tends to converge onto a different immersion of the same topology. The top row of Figure 14 shows the continuous transition from the connected sum of two Möbius bands (equivalently two cross-caps) to

Additionally, we also support cutting a topological handle (or a cross-handle) and attaching a disk on the two boundaries.

5.2 Knots and Links Editing

Our editing system can process multiple geometry models, thus enabling links among them. Figure 13 shows the result of linking a modified buddha with a variant of the Klein bottom through two double tori. We also enable the ability of generating knots, such as the one shown in Figure 12.

1027 a pinched torus (Figure 14 (d)) which is topologically homeomorphic
 1028 to gluing a cross-handle to a disk.

1029 To edit the seams, the user can select a region in the mesh and
 1030 reverse the vertex orders of all triangles in the same region. This
 1031 leads to the swap of roles of the two corners in a corner of these
 1032 triangles. As illustrated in Figure 5, the middle triangle’s vertex order
 1033 is flipped from the left subfigure to the right subfigure, resulting the
 1034 change of location of the seam (the blue edges). Note that before
 1035 the change, $c_{0,o} = c_{10}$. After the change, $c_{0,o} = c_{11}$.

1036 Using the same approach, we also constructed an animation to
 1037 show that attaching a handle to a non-orientable topological surface
 1038 is homeomorphic to attaching a cross-handle to the same topological
 1039 surface (Figure 14 (bottom)).

1041 1042 1043 1044 1045 1046 6 BRANCHED COVERING SPACE GENERATION AND 1047 VISUALIZATION

1048 Given a surface S and an N -ReSy field F defined on it, there is an
 1049 N -fold branched covering space (BCS) \bar{S} for S such that F is lifted
 1050 to a vector field \bar{F} on \bar{S} .

1051 Roy et al. [Roy et al. 2018b] provide two methods of computing a
 1052 BCS given an orientable surface with an N -RoSy field. In the first
 1053 method, they compute what they refer to as an *essential cut graph*
 1054 based on the singularities in F . They then cut the surface open
 1055 along the edges in the cut graph, make $N - 1$ copies of the cut-open
 1056 surface, and then stitch the N layers together along the cut based
 1057 on the gaps along the cut. This results in a topologically correct
 1058 BCS \bar{S} whose N layers are co-located in space and thus visually
 1059 indistinguishable from the original surface S . Users are then given
 1060 the ability to manually modify the geometry of \bar{S} such as moving a
 1061 layer away from other layers. When deformation is introduced due
 1062 to these editing operations, it is treated as an optimization problem
 1063 whose objective is to balance between maintaining the similarity of
 1064 each layer of \bar{S} to S and ensuring smooth transition between layers.

1065 The aforementioned approach, while effective in separating the
 1066 layers geometrically, often incurs large surface deformation. To
 1067 overcome this problem, Roy et al. [Roy et al. 2018b] provide a second
 1068 approach in which the N layers are initially placed so that there
 1069 are no mutual intersections between any pair of layers. When two
 1070 layers are to be connected, a tube (a topological cylinder) is added to
 1071 fill the openings on both layers. In the case of more than two layers
 1072 are to be connected, Roy et al. introduce a geometric construct that
 1073 they refer to as *docking stations*, which is a topological sphere or a
 1074 torus with a topologically correct number of opening loops to which
 1075 the tubes from the involved layers can be attached. Compared to
 1076 the first approach in which deformation design is used to separate
 1077 the layers, the tube-based approach has the advantage of having
 1078 relatively small overlaps among the layers, which are concentrated
 1079 near the docking stations.

1080 In our work, we adapt Roy et al.’s approaches to the construction
 1081 of BCSs for surfaces with an N -ReSy field. In addition, we introduce
 1082 a novel tube-based construction method that removes the need for
 1083 docking stations for both N -ReSy and N -RoSy fields.

1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 6.1 ReSy Field BCSs

The essential cut graph for an N -ReSy field has a similar structure as that of an N -RoSy field. Let $\chi(S)$ be the *Euler characteristic* of S . Assuming only first-order singularities, it is possible to generate an essential cut graph with the following structure. There are $|\chi(S)|$ groups of N singularities, all of which have the same singularity index, and an arbitrary number of singularity pairs of which one is positively indexed and the other negatively indexed. Note that each group of k singularities is a *linear graph*. In addition, there can be a loop for each generator in the homotopy of the surface without any singularities on it.

Due to the similarity between the structure of an essential cut graphs for N -ReSy fields to that of N -RoSy fields, the process of BCS construction is also similar.

First, based on the given essential cut graph, we select a vector per triangle such that the place and amount of jumps between adjacent triangles match that of the essential cut graph. Second, the surface S is cut open along the essential cut graph. Next, the cut open surface is duplicated $N - 1$ times but spatially co-located with the original copy. For each copied layer, a pair of two frames sharing the same X -axis (i.e. a single 1-ReSy) is selected from each N -ReSy of the triangles in that layer, and the shared X -axis becomes the vector of that triangle in the resulting vector field. Finally, the openings on the N layers are filled by connecting them to the openings of appropriate layers, chosen to make the vectors consistent.

For the tube-based BCS construction, we following a similar pipeline with surface cutting, vector selection, layer duplication, and vector transformation. The only difference is in the last step. To fill the openings on the layers, we construct docking stations with the correct number of holes, which we use to receive the tubes sent from the layers.

6.2 Tube-Based BCSs without Docking Stations

We observe that in the cases of N -RoSy and N -ReSy fields in which $N \geq 2$, the connectivity among the layers can still be unclear. See Figure 15 (top) for an example.

To address this difficulty, we introduce a new approach to the construction of BCSs using tubes **without** docking stations. The key insight is to realize that the connectivity among N layers along a cut edge can be considered as a permutation of the N layers, i.e., an element in S_N , the permutation group on N elements. With N -RoSy fields, Roy et al. [Roy et al. 2018b] use docking stations to help connect tubes in order to satisfy the permutations among layers, which are *cyclic permutations*.

On the other hand, if the permutation along a cut edge is the product of disjoint transposition permutations, then it is possible to connect the layers without using a central docking station. For example, the following permutation $(1, 2)(4, 6)$ along a cut edge in an 6-ReSy field means that the openings on layers 1 and 2 can be connected to each other with a single tube. Similarly, layers 4 and 6 can have their opening connected by a single tube. In contrast, the corresponding opening on layer 3 is closed as if it had not been cut, so is the opening on layer 5.

A permutation can be decomposed as the product of disjoint cycles [Shumyatsky 2013], with the order of the permutation being

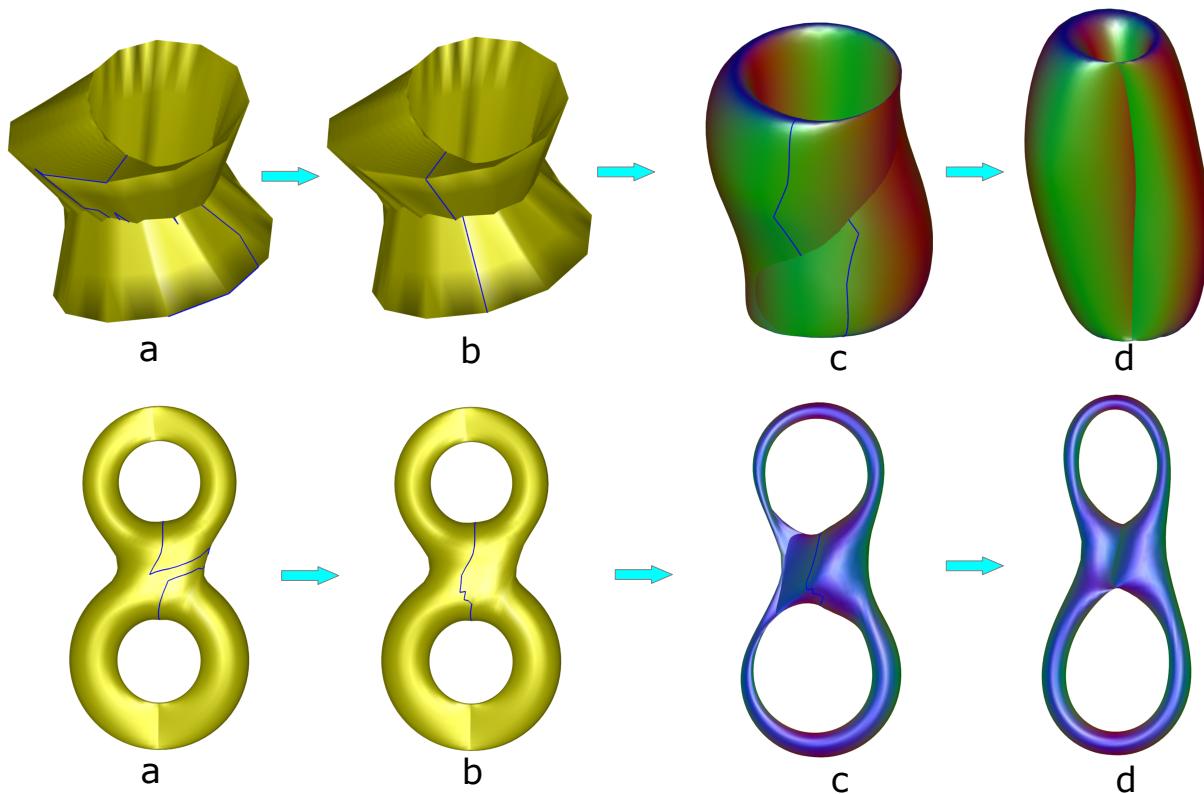


Fig. 14. Top row: (a) two cross caps attached to themselves with the orientability seam shown as blue curve, (b) seam edited to facilitate the smoothing sequence, (c) intermediate snapshot of the smoothing process, and (d) the final cross-handle. Bottom row: (a) two cross-handles attached to themselves with the orientability seam shown as blue curve, (b) seam edited to facilitate the smoothing sequence, (c) intermediate snapshot of the smoothing process, and (d) the final result with a cross-handle and a handle.

the least common multiple of the order of each cyclic permutation in the product. More specifically, if a permutation has an order of two, then it can be decomposed as the product of transposition permutations. Therefore, we seek to generate a cut graph along which the permutations are of order two.

The need for a docking station occurs when $N \geq 2$ and there is a group of either 2 singularity (positive and negative pair) or N singularities (with the same singularity index).

We can construct a cut graph that has a different structure than the essential cut graph. We refer to this new type of cut graph as a *polygonal cut graph*, which consists of a number of polygons. Each polygon corresponds precisely to a group of singularities in the essential cut graph, i.e. the nodes in the polygon are the singularities in the corresponding group. What is different between a polygon and a group is the connectivity. Instead of having a linear graph connecting the singularities, we use a polygon with one vertex for each singularity.

The jump for the edges on the path between two adjacent singularities is constant. By considering \mathbb{D}_n as a subgroup of S_n , we see that the reflection elements, having order two, correspond to reflections. To make the gap along path be a reflection in \mathbb{D}_n , we

multiply the rotation by a reflection, which always results in a reflection. Note that the dihedral group \mathbb{D}_N consists of the identity, $N - 1$ cyclic permutations, and N order-two permutations. After multiplying with one of the order-two dihedral group permutations s , each of the $N - 1$ cyclic permutations on the original linear graph becomes an order-two dihedral permutations. Finally, the gaps can be made consistent with the singularities by assigning the last edge (the new edge) a gap of s . This leads to having an order-two permutation per edge of the polygon, thus removing the need for a docking station.

We then select a vector per triangle in the mesh to be consistent with the polygon cut graph by flipping the normal of triangles inside the polygon.

Next, we cut the surface open along the polygon cut graph, before duplicating the surface into $N - 1$ layers, each of which is applied a unique dihedral transformation.

Finally, we connect two layers i and j with a tube if they have openings that correspond to the same path in the polygon cut graph whose corresponding jump is a swap between i and j . For the other layers, their openings corresponding to the same path in the polygon cut graph are sealed without being connecting to other layers.

Figure 15 shows the difference between a BCS using docking station [Roy et al. 2018b] (top) and using our pairwise connection approach (bottom). Notice the twisting near the intersections of the tubes in the top figure that makes it difficult to understand the exact connectivity. With using only pairwise layer connection, this difficulty is not present.

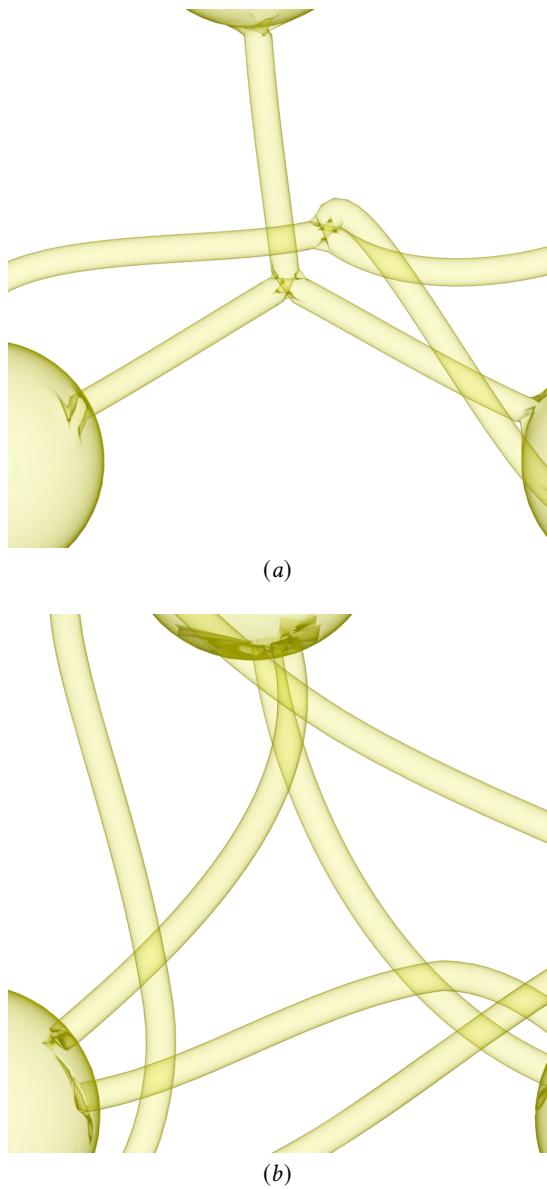


Fig. 15. Our pairwise connection of tubes in (b) avoids the possible overlapping of the tubes when connected to a docking station in (a).

7 PERFORMANCE

Our design system has been tested on a system with Intel(R) Core(TM) CPU with 2.59 Ghz of CPU with a RAM of 12 GB. The performance

time, which is dependent on the mesh size, ranges from a few seconds to about twenty seconds to complete the entire pipeline of the BCS construction. Global parameterization performance depends on the number of the triangles and also on the grid size of the parameterization lines. It ranges from about 10 seconds to about a minute for mesh size ranging from 10000 triangles elements to 50000 triangles. The remeshing and surface attachment only took a few seconds for lower size meshes to upto 10 seconds for higher resolution meshes.

8 CONCLUSION

In this paper, we address geometry processing of non-orientable surfaces. By extending the corner data structure from orientable surfaces to non-orientable surfaces, we are able to perform important tasks such as editing the geometry of topology of non-orientable surfaces as well as constructing knots and links involving them. Furthermore, we provide the capability of editing the seams in the surface across which vertex orderings are inconsistent. This, coupled with Laplacian smoothing, allows us to create animations that provide intuitions behind several important results from topology without the need to cut and glue some planar polygons.

For pattern synthesis and semi-regular remeshing of non-orientable surfaces, we introduce the notion of reflectional symmetry fields and adapt the analysis, design and processing of rotational symmetry fields on orientable surfaces to that of reflectional symmetry fields on non-orientable surfaces.

Finally, we extend algorithms for branched covering spaces for orientable surfaces to non-orientable surfaces. To better show the interconnectivity among layers, we introduce a novel tube-based visualization using results on the dihedral groups. This novel visualization can also handle higher-order singularities, for which existing tube-based method fails.

Our system is not without limitations. For example, editing seams on non-orientable surfaces requires accessing vertices in the mesh that are inside other part of the same model, which can be challenging for users who are not yet familiar with 3D geometry. Also, our system currently focuses on the processing of closed surfaces. We plan to improve our system by addressing these limitations.

In addition, we wish to explore additional types of symmetries for non-orientable surfaces, such as the equivalent of wallpaper groups for the plane.

Given two non-orientable surfaces that are topological equivalent, automatically generating an animation that provides an intuitive transition between the surfaces is a future research avenue that we plan to explore.

Finally, we plan to investigate better visualization and interaction approaches for non-orientable surfaces where surface self-intersection is prevalent.

REFERENCES

- Ergun Akleman, Jianer Chen, Qing Xing, and Jonathan L. Gross. 2009. Cyclic Plain-Weaving on Polygonal Mesh Surfaces with Graph Rotation Systems. *ACM Trans. Graph.* 28, 3, Article 78 (July 2009), 8 pages. <https://doi.org/10.1145/1531326.1531384>
- Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. 2003. Anisotropic Polygonal Remeshing. In *ACM SIGGRAPH 2003 Papers* (San Diego, California) (*SIGGRAPH '03*). Association for Computing Machinery, New York, NY, USA, 485–493. <https://doi.org/10.1145/1201775.882296>

- 1369 M.A. Armstrong. 1979. *Basic topology*. McGraw-Hill Book Co. <https://books.google.com/books?id=x0PvAAAAMAAJ>
- 1370 Martina Bátorová, Miroslava Valíková, and Pavel Chalmovianský. 2013. Desingularization of ADE Singularities via Deformation. In *Proceedings of the 29th Spring Conference on Computer Graphics* (Smolenice, Slovakia) (SCCG '13). ACM, New York, NY, USA, Article 035, 8 pages. <https://doi.org/10.1145/2508244.2508249>
- 1371 David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. 2013. Integer-Grid Maps for Reliable Quad Meshing. *ACM Trans. Graph.* 32, 4, Article 98 (July 2013), 12 pages. <https://doi.org/10.1145/2461912.2462014>
- 1372 David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-Integer Quadrangulation. *ACM Trans. Graph.* 28, 3, Article 77 (July 2009), 10 pages. <https://doi.org/10.1145/1531326.1531383>
- 1373 Marcel Campen, David Bommes, and Leif Kobbelt. 2012. Dual Loops Meshing: Quality Quad Layouts on Manifolds. *ACM Trans. Graph.* 31, 4, Article 110 (July 2012), 11 pages. <https://doi.org/10.1145/2185520.2185606>
- 1374 Robert Ferréol. 2017. Real Projective Plane. <https://www.mathcurve.com/surfaces.gb/planprojectif/planprojectif.shtml>.
- 1375 Robert Ferréol and Jacques Mandonnet. 2017. Klein Bottle. <https://www.mathcurve.com/surfaces.gb/klein/klein.shtml>.
- 1376 Robin Hartshorne. 1977. *Algebraic geometry*. Springer, New York. <http://opac.inria.fr/record=b1102947>
- 1377 Aaron Hertzmann and Denis Zorin. 2000. Illustrating Smooth Surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (SIGGRAPH '00). ACM Press/Addison-Wesley Publishing Co., USA, 517–526. <https://doi.org/10.1145/344779.345074>
- 1378 Felix Kälberer, Matthias Nieser, and Konrad Polthier. 2007. QuadCover - Surface Parameterization using Branched Coverings. *Computer Graphics Forum* (2007). <https://doi.org/10.1111/j.1467-8659.2007.01060.x>
- 1379 Matthias Nieser, Jonathan Palacios, Konrad Polthier, and Eugene Zhang. 2012. Hexagonal Global Parameterization of Arbitrary Surfaces. *IEEE Trans. Vis. Comput. Graph.* 18, 6 (2012), 865–878. <http://dblp.uni-trier.de/db/journals/tvcg/tvcg18.html#NieserPPZ12>
- 1380 Matthias Nieser, Konstantin Poelke, and Konrad Polthier. 2010. Automatic Generation of Riemann Surface Meshes. In *Advances in Geometric Modeling and Processing*, Bernard Mourrain, Scott Schaefer, and Guoliang Xu (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 161–178.
- 1381 Jonathan Palacios and Eugene Zhang. 2007. Rotational Symmetry Field Design on Surfaces. *ACM Trans. Graph.* 26, 3 (July 2007), 55–es. <https://doi.org/10.1145/1276377.1276446>
- 1382 Jonathan Palacios and Eugene Zhang. 2011. Interactive Visualization of Rotational Symmetry Fields on Surfaces. *IEEE Trans. Vis. Comput. Graph.* 17, 7 (2011), 947–955. <https://doi.org/10.1109/TVCG.2010.121>
- 1383 Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Trans. Graph.* 36, 2, Article 16 (April 2017), 16 pages. <https://doi.org/10.1145/2983621>
- 1384 Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. 2006. Periodic Global Parameterization. *ACM Trans. Graph.* 25, 4 (Oct. 2006), 1460–1485. <https://doi.org/10.1145/1183287.1183297>
- 1385 Jarek Rossignac. 1999. Edgebreaker: Connectivity Compression for Triangle Meshes. *IEEE Transactions on Visualization and Computer Graphics* 5, 1 (Jan. 1999), 47–61. <https://doi.org/10.1109/2945.764870>
- 1386 L. Roy, P. Kumar, S. Golbabaei, Y. Zhang, and E. Zhang. 2018a. Interactive Design and Visualization of Branched Covering Spaces. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan 2018), 843–852. <https://doi.org/10.1109/TVCG.2017.2744038>
- 1387 L. Roy, P. Kumar, S. Golbabaei, Y. Zhang, and E. Zhang. 2018b. Interactive Design and Visualization of Branched Covering Spaces. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan 2018), 843–852. <https://doi.org/10.1109/TVCG.2017.2744038>
- 1388 Pavel Shumyatsky. 2013. The dihedral group as a group of automorphisms. *Journal of Algebra* 375 (2013), 1 – 12. <https://doi.org/10.1016/j.jalgebra.2012.04.035>
- 1389 Xavier Tricoche. 2002. *Vector and Tensor Topology Simplification, Tracking, and Visualization*. Ph.D. Dissertation. University of Kaiserslautern.
- 1390 Miroslava Valíková and Pavel Chalmovianský. 2015. Visualisation of complex functions on Riemann sphere. *The Visual Computer* 31, 2 (2015), 141–154. <https://doi.org/10.1007/s00371-014-0928-3>
- 1391 Amir Vaxman, Marcel Campen, Olga Diamanti, David Bommes, Klaus Hildebrandt, Mirela Ben-Chen, Technion, and Daniele Panozzo. 2017. Directional Field Synthesis, Design, and Processing. In *ACM SIGGRAPH 2017 Courses* (Los Angeles, California) (SIGGRAPH '17). Association for Computing Machinery, New York, NY, USA, Article 12, 30 pages. <https://doi.org/10.1145/3084873.3084921>
- 1392 Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. 2009. State of the Art in Example-based Texture Synthesis. In *Eurographics '09 State of the Art Reports (STARs)* (eurographics '09 state of the art reports (stars) ed.). Eurographics. <https://www.microsoft.com/en-us/research/publication/state-of-the-art-in-example-based-texture-synthesis/>

1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482