Construction and Visualization of Branched Covering Spaces

Sanaz Golbabaei* Oregon State University
Lawrence Roy† Roy Family Homeschool
Prashant Kumar‡ Oregon State University
Eugene Zhang§ Oregon State University

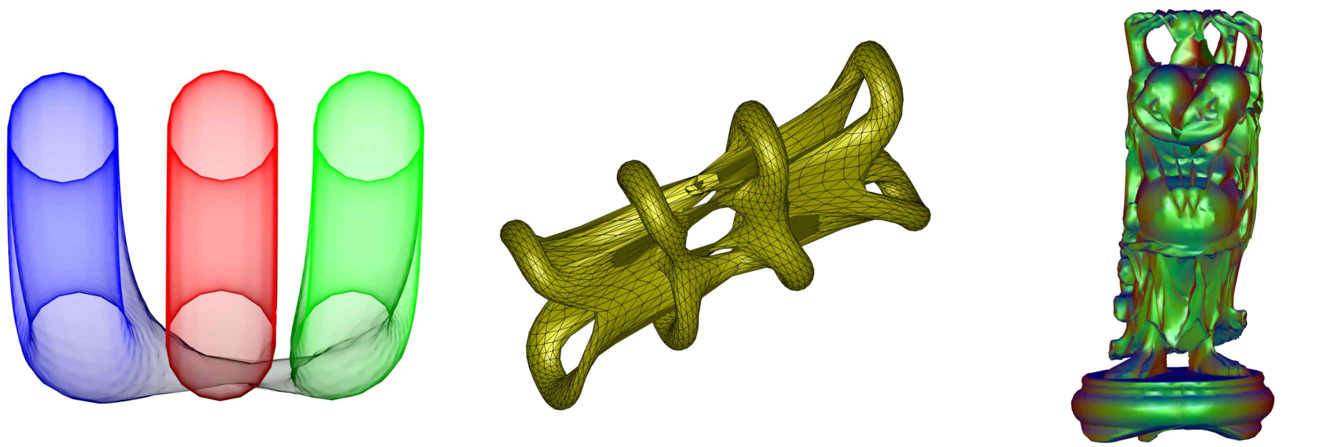

Figure 1: *Three branched covering spaces (BCSs) constructed using our system: (left) a three-fold covering of the torus visualized using the SmokeSurface technique, (middle) a four-fold cover of a genus-2 surface, and (right) a two-fold covering of the Stanford Buddha model using normal maps. Our system allows the construction and visualization of BCSs.*

Abstract

Branched covering spaces are a mathematical concept which originates from complex analysis and topology and has found applications in geometry remeshing. Given a manifold surface and an N-way rotational symmetry field, a branched covering space is a manifold surface that has an N-to-1 map to the original surface except at the so-called *ramification points*, which correspond to the singularities in the rotational symmetry field. Understanding the notion and mathematical properties of branched covering spaces is important to researchers in geometry processing.

In this paper, we provide a framework to construct and visualize the branched covering space (BCS) of an input mesh surface and a rotational symmetry field defined on it. In our framework, the user can visualize not only BCSs but also their construction process. In addition, our system allows the user to design the geometric realization of the BCS using mesh deformation techniques. This enables the user to verify important facts about BCSs such as that they are manifold surfaces around singularities and the *Riemann-Hurwitz formula* which relates the Euler characteristic of the BCS to that of the original mesh.

Keywords: branched covering space, topology, differential geometry, digital geometry processing

Concepts: •**Human-centered computing** → **Visualization techniques;** •**Computing methodologies** → **Shape modeling;**

*e-mail:golbabbas@eecs.oregonstate.edu
†email:ldr709@gmail.com
‡email:kumarpra@eecs.oregonstate.edu
§e-mail:zhange@eecs.oregonstate.edu

SA '16 Technical Briefs ,, December 05-08, 2016, , Macao
ISBN: 978-1-4503-4541-5/16/12 $15.00
DOI: http://dx.doi.org/10.1145/3005358.3005367

1 Introduction

The *Branched covering space* is a mathematical concept in topology, which has found applications in quadrangular remeshing, i.e., the generation of a mesh of quads from an input triangle mesh. In quadrangular remeshing, the edges in the quads are often required to be approximately aligned with a given *cross* field that is usually derived from the principal curvature directions in the underlying surface. To prevent T-junctions from occurring in the quad mesh, Kälberer et al. [2007] propose to lift the cross field in the input mesh to a vector field on the four-fold branched covering space of the surface. They then perform Hodge-decomposition to remove the divergence-free part of the vector field, thus preventing T-junctions in the remesh.

Mathematically, BCS is the extension of the concept of covering space, with the additional notion of *ramification points*. While this addition can seem minor, the behaviors of a BCS are significantly more complex than a covering space and thus more difficult to understand. The additional complication of BCSs can lead to misconceptions such as that the BCS of a manifold surface is no longer a manifold surface due to the presence of ramification points.

In addition, existing research that requires some visual descriptions of a BCS often does so with hand-drawn illustrations of some patch on the BCS, usually around a ramification point. To the best of our knowledge, there is no published algorithm to explicitly compute the BCS of an input mesh, nor to visualize the BCS. This can make it difficult to grasp for graphics researchers who are interested in remeshing but have not acquired sufficient mathematical background in topology and differential geometry. In this paper, we address this by providing efficient algorithms to construct the BCS of an input surface. Moreover, we provide functionalities that enable the user to deform the BCS as well as visualize them in a number of visualization modes, thus enabling the user to see and interact with the BCS and gain intuitions and verify important known results about BCS. In particular, we strive to enable the understanding of the following known facts about BCSs through our system:

1. An N-way rotational symmetry (N-RoSy) field on the input mesh leads to an N-fold branched covering space of the original mesh.
2. The singularities in the N-RoSy field become the ramification points of the BCS.
3. Away from ramification points, every point in the original mesh corresponds to N points in the BCS, each of which is assigned one of the vectors in the N-RoSy at the base point.
4. If the input mesh represents a manifold surface, then its BCS is also a manifold surface, i.e., the points in the BCS corresponding to a ramification point are manifold points.
5. The index of a singularity in the vector field on the BCS of the mesh is decided by the index of the corresponding singularity in the N-RoSy field.
6. The *Riemann-Hurwitz formula*, which states that the Euler characteristic of its BCS is related to that of the base surface and the number of ramification points in the BCS.
7. The BCS is independent of the way in which it is constructed.

2 Field-Guided Branched Covering Spaces

In this section, we briefly review necessary concepts and facts regarding branched covering spaces. For a more detailed and rigorous description, we refer our readers to Appendix A.

Roughly speaking, given a topological space X, a covering space of X is another space C along with a continuous and surjective map from C to X such that every point in X corresponds to the same number N of points in C. Here, N can be ∞. For example, the real line $\mathbb{R}^1$ provides an infinite cover of the circle $\mathbb{S}^1$ with the map $p(\theta) = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$. In this case, every point $\begin{pmatrix} \cos\theta_0 \\ \sin\theta_0 \end{pmatrix}$ ($\theta_0 \in [0, 2\pi)$ in $\mathbb{S}^1$ corresponds to an infinite set $\{\theta_0 + 2k\pi | k \in \mathbb{Z}\}$.

BCSs are a generalization of the covering in that there are *ramification points* in the map. For example, the set $\mathbb{R}^1$ is not a cover of $\mathbb{R}^+$, the set of non-negative real numbers. While every positive number in $\mathbb{R}^+$ corresponds to two points in $\mathbb{R}^1$, 0 has only one preimage. In this case, $\mathbb{R}^1$ is a *branched* cover of $\mathbb{R}^+$ with 0 being the ramification point.

The notion of branched covering spaces also applies to surfaces, which are the main focus in this paper. Moreover, we are interested in branched covering spaces driven by a frame field on the input surface.

The computational setting is as follows. The input to our system is a triangular mesh M with an N-way rotational symmetry field F defined on the triangles of M. We follow the convention of the geometry processing community and refer to the field as an N-RoSy field. More specifically, on each triangle $t \in M$, there are N vectors that are evenly spaced angularly. A branched covering space C of M is also a triangular mesh with N times as many triangles as M. For each triangle $t \in M$, there are N triangles in C that correspond to t. Moreover, there is a vector defined in each of these triangles, whose collection is identical to the set of the N-RoSy defined in t. In other words, if one stacks the N triangles on top of each other and collapse the stack, the vectors in these triangles will form the original N-RoSy in t. This collection of triangles can be sorted in the counterclockwise sense of the N-RoSy in t. The vector field on C is said to be *lifted* from the N-RoSy field on M.

For two adjacent triangles t_1 and t_2 in M, there are two sets of N triangles in C that correspond to them. This can be visualized as two N-layered structures being glued together. While it is desirable to connect the triangles on the same layer, this is not always the case. Sometimes, a triangle of layer p for the triangle t_1 is connected with layer $p + k$ for the triangle t_2. In fact, in this case, any layer q for the triangle t_1 will be paired with layer $q + k$ for the triangle t_2. Here, k is referred to as the *gap* from t_1 to t_2. It is straightforward to verify that the gap from t_2 to t_1 is $-k$. Figure 2 illustrates this with an example.

The number of vertices in C is not N times the number of vertices in M, due to the existence of the ramification points in M, which have fewer than N corresponding vertices in C. The ramification points are caused by the singularities in the N-Rosy field, and their corresponding vertices in C are also singularities with respect to the vector field lifted from M. The Poincaré index of a singularity p in the N-RoSy field is related to that of the corresponding singularity q in the lifted vector field by the formula $I(q) = N(I(p) - 1) + 1$.

The topology of C is also related to that of M, according to the Riemann-Hurwitz formula: $\chi(C) = N\chi(M) - \sum_{p \in R}(e_p - 1)$ where $\chi(S)$ is the *Euler characteristic* of a surface S, R is the set of ramification points in the branched cover and e_p is the *ramification index* of a ramification point p.

The above theoretical results are important, yet they are difficult to convey intuitively. We have designed our BCS visualization system to address these challenges.

3 Our Approach

We have constructed a BCS visualization system in which the user can interactively design the geometric embedding of the BCS given an input triangle mesh and an N-RoSy field defined on the mesh. In addition, the user can control the opacity of the BCS as well as the vector field on the BCS lifted from the N-RoSy field on the input mesh.

Given the input mesh M and the N-RoSy field $\mathbb{F}$, our system automatically constructs an initial BCS as follows (Figure 3). First, we compute a cut graph G on M such that G consists of edges in M [Kälberer et al. 2007]. As part of the cut graph computation step, we also select a vector from the N-RoSy inside every triangle of M so that the vector field is continuous across all the edges in

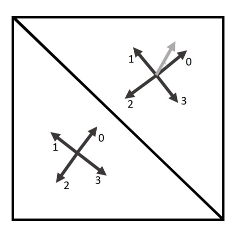

Figure 2: *The gap from the lower-left triangle (first) to the upper-right triangle (second) is 0, since the parallel transport of the* 0*-indexed vector in the first triangle has the smallest angle with the* 0*-indexed vector in the second triangle.*

M except those on the cut graph G. Across edges on G, the discontinuity between the vectors on the opposing triangles is $\frac{2k\pi}{N}$ where $0 \le k < N$ is an integer and is referred to as the *gap* between the two triangles. Note that the gap along each edge on the cut graph is not constant. We now cut the mesh open along edges in G and replicate it $N-1$ times. This leads to N identical copies of the cut open geometry. The original cut open mesh is referred to as layer 0, while the $N-1$ copies are numbered as layers 1 through $N-1$. For a layer k, the vector defined on a triangle t is obtained by rotating the vector of the corresponding triangle in layer 0 by an angle $\frac{2k\pi}{N}$ counterclockwise. Finally, we stitch all N layers together based on the gaps of the edges in the original cut graph G. This leads to the BCS of the input surface along with a continuous vector field.

The initial BCS will geometrically look identical to the input mesh, since all the layers are co-located. This makes the understanding of important properties of BCSs challenging. We address this by allowing the user to interactively change the shape of the BCS (geometric design) so that the layers in the BCS are no longer co-located. By reusing frameworks from mesh-based shape deformation [Sumner and Popović 2004], we allow the user to specify handles on the BCS in order to move, rotate, and scale them. The location of the vertices outside the handle regions are computed by using the energy term from [Sumner and Popović 2004], which seeks to maximize the geometric similarity between the BCS and the original input mesh. As the user continuously moves, rotates, and scales a handle on the BCS, our system interactively solves the energy minimization and updates the geometry of the BCS, thus providing instantaneous feedback. Figure 4 shows an example BCS design before and after the deformation.

In addition to photorealistic rendering, the user can also visualize the vector field on the BCS using well-known techniques such as LIC [Cabral and Leedom 1993] (Figure 5). Furthermore, due to the self-intersecting nature of BCSs, our system enables translucent rendering of the surface based on the SmokeSurface technique [von Funck et al. 2008] (Figure 1 (left) and Figure 7 (bottom)). We have also found that normal map rendering, as shown in Figure 1 (right) and Figure 7 (top), useful.

While the above capabilities already allow the user to see and interact with the BCS, there are a number of important properties of the BCS that are still difficult to visualize.

First, the Poincaré indexes of the singularities in the N-RoSy field are intrinsically connected to those of the corresponding singularities in the vector field on the BCS. However, due to the self intersections in the BCS, it is difficult to see the complete neighborhood around a singularity in the vector field on the BCS.

Second, the topology of the BCS (e.g., the number of connected components and handles) is determined by the topology of the input mesh and N (as in N-RoSy). In addition to each handle in the original surface being replicated N times, additional handles can occur due to connecting the N layers of the BCS. These additional handles are usually geometrically folded and part of the self-intersections in the BCS.

To address these challenges, we again make use of the mesh deformation framework describe above.

To see the complete neighborhood around a singularity in the BCS, we provide a deformation animation of the BCS such that the neighborhood is unfolded continuously onto a plane or a hemisphere, thus allowing the vector field pattern around the singularity to be completely visible (Figure 5). We notice that the unfolding animation is also useful in showing the transition from the original singularity of the N-RoSy field to its corresponding singularity in the vector field on the BCS.

To see the folded handles, we first extract all the topological handles [Zhang et al. 2005], each of which is highlighted using a loop (a homological generator associated with the topological handle). If the user wishes to see a handle in its entirety, we extract a neighborhood of the handle and unfold it so that it is visible in its entirety through mesh deformation. Again, we note that the animation sequence can help the user see the transition and thus better understand the fact that there is a handle folded in the BCS (Figure 6).

4 Results and Evaluation

We have presented our system to 40 graduate and undergraduate students of Mathematics and Computer Science as well as a number of Mathematics faculty members teaching classes in topology, differential geometry, and complex analysis. We have received positive feedback on the effectiveness of our system in demonstrating the mathematical properties of BCSs using mesh deformation as well as SmokeSurface, Normal Maps, and LIC visualizations. In fact, visualizing BCS construction using animation was suggested by our audience and thus implemented. They also have provided suggestions for reducing the amount of self-intersections in BCSs, which we plan to leverage in our future research on the subject.

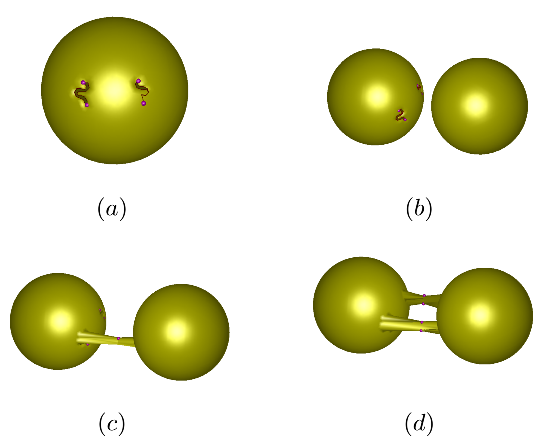

Figure 3: *The construction of a BCS from an input mesh with a field (not shown): (a) mesh cutting along the cut graph, (b) layer replication, (c) layer stitching, and (d) final BCS.*

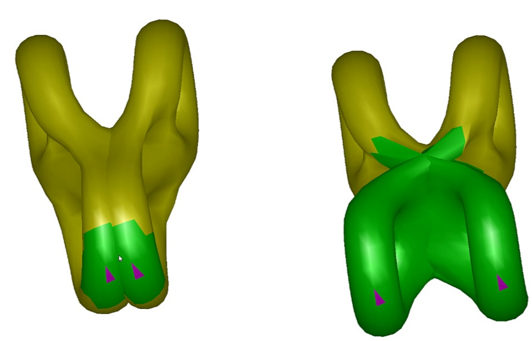

Figure 4: *Our system allows the BCS to be deformed: (left) regions to be deformed (green) around the deformation handle (purple triangles), and (right) the deformed BCS.*

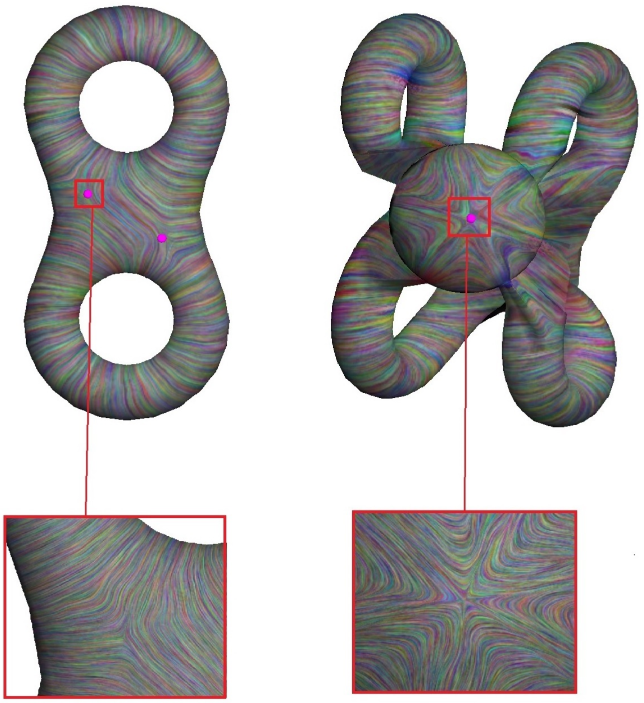

Figure 5: *A* $-\frac{1}{2}$ *indexed singularity in the* 2*-RoSy field on the double torus (left) corresponds to a* -2 *indexed singularity in the vector field in the BCS (right).*

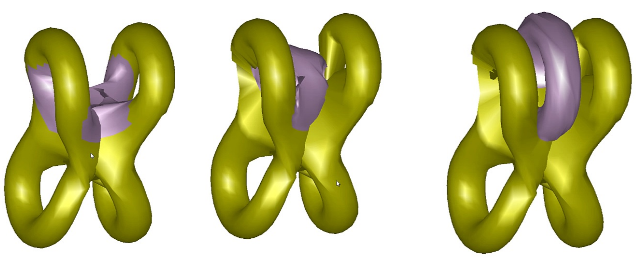

Figure 6: *A set of frames in the animation of inflating a handle that is otherwise difficult to see due to the self-intersections in the BCS.*

References

ARMSTRONG, M. 1979. *Basic topology*. McGraw-Hill Book Co.

CABRAL, B., AND LEEDOM, L. C. 1993. Imaging vector fields using line integral convolution. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '93, 263–270.

HARTSHORNE, R. 1977. *Algebraic geometry*. Graduate texts in mathematics. Springer, New York.

KÄLBERER, F., NIESER, M., AND POLTHIER, K. 2007. Quadcover - surface parameterization using branched coverings. *Comput. Graph. Forum 26*, 3, 375–384.

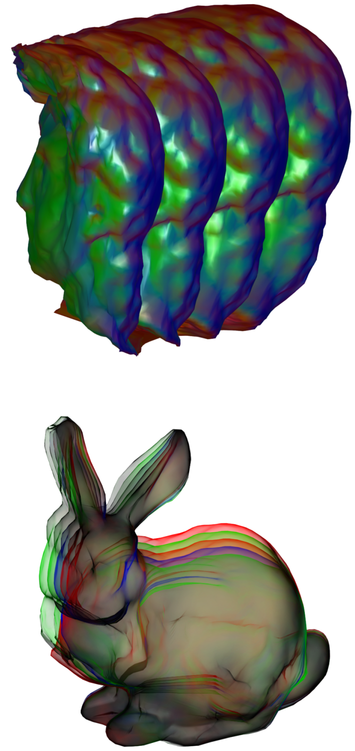

Figure 7: *BCSs for complex geometry with relatively high* N*s as in* N*-RoSy: (a) four-fold cover of David, and (b) five-fold cover of the Stanford Bunny.*

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH '04, 399–405.

VON FUNCK, W., WEINKAUF, T., THEISEL, H., AND SEIDEL, H.-P. 2008. Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2008) 14*, 6 (November - December), 1396–1403.

ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2005. Feature-based surface parameterization and texture mapping. *ACM Trans. Graph. 24*, 1 (Jan.), 1–27.