

Exploded View Diagrams of Mathematical Surfaces

Olga Karpenko, Wilmot Li, Niloy J. Mitra, and Maneesh Agrawala

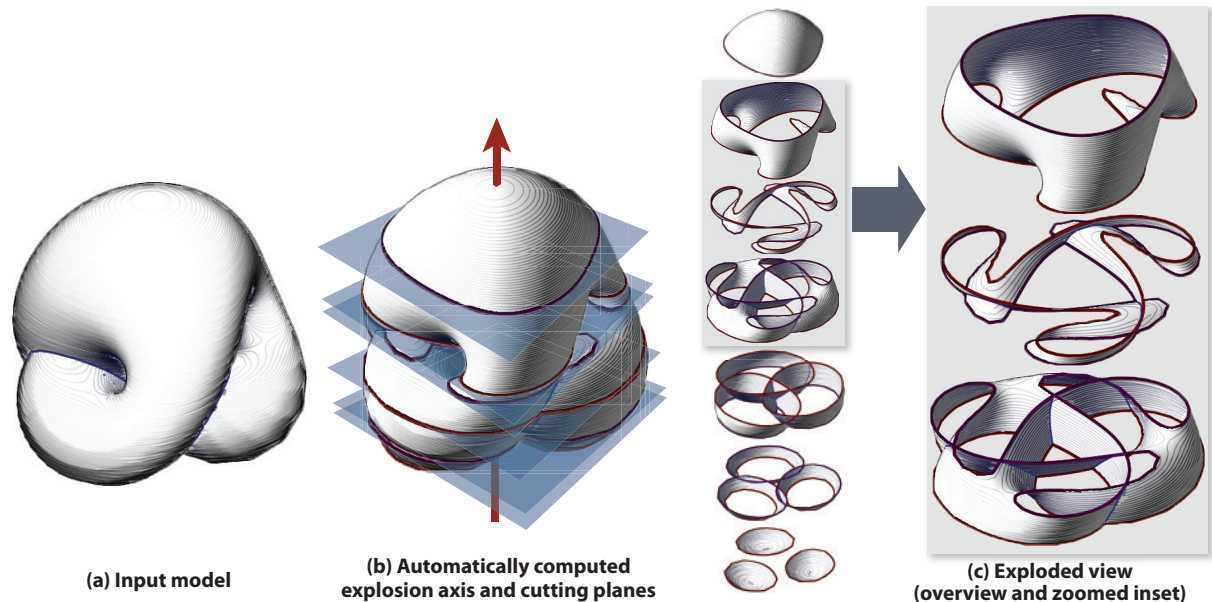


Fig. 1. **Exploded view of Boy's Surface.** Given a polygonal model of a mathematical surface (a), our system determines an explosion axis based on detected object symmetries and analyzes the surface geometry to position cutting planes (b). The resulting exploded view conveys the internal structure of the surface by exposing important cross-sections that contain key geometric features (c).

Abstract— We present a technique for visualizing complicated mathematical surfaces that is inspired by hand-designed topological illustrations. Our approach generates exploded views that expose the internal structure of such a surface by partitioning it into parallel slices, which are separated from each other along a single linear explosion axis. Our contributions include a set of simple, prescriptive design rules for choosing an explosion axis and placing cutting planes, as well as automatic algorithms for applying these rules. First we analyze the input shape to select the explosion axis based on the detected rotational and reflective symmetries of the input model. We then partition the shape into slices that are designed to help viewers better understand how the shape of the surface and its cross-sections vary along the explosion axis. Our algorithms work directly on triangle meshes, and do not depend on any specific parameterization of the surface. We generate exploded views for a variety of mathematical surfaces using our system.

Index Terms—exploded view diagrams, mathematical visualization, symmetry

1 INTRODUCTION

Complicated 3D surfaces are the primary subjects of study in several branches of mathematics, including most sub-fields of topology and geometry. For example, geometers study a variety of 3D surfaces, as well as 3D projections of higher dimensional surfaces, to understand how they evolve and what properties remain invariant. One way to characterize the overall shape and structure of such surfaces is to consider both their global geometric properties, such as symmetries, as

well as their local geometric features, such as self-intersections and critical points, i.e., minima, maxima and saddle points. However, many interesting mathematical surfaces twist, turn, and fold back on themselves, creating self-occlusions that can make it difficult to see these geometric features. Yet, seeing these geometric features is often crucial for understanding the overall shape and structure of the object. For example, in Boy's Surface, most of the interesting geometric features are hidden within the object, which makes it difficult to understand its internal structure (see Figure 1).

As Francis notes in *A Topological Picturebook* [15], topologists and geometers have a long history of creating mathematical illustrations to help convey the shape and internal structure of complicated surfaces. One technique for depicting such surfaces is to create an *exploded view* by cutting the object into parallel *slices* and then separating the slices from each other along a linear *explosion axis* (see Figure 2). Such exploded views expose cross-sections of the object and reveal its internal features and structure. The cross-sections also help the viewer understand how the shape of the surface varies along the explosion axis.

- Olga Karpenko is with UC Berkeley, E-mail: oakarpenko@gmail.com.
- Wilmot Li is with Adobe Systems.
- Niloy J. Mitra is with KAUST / IIT Delhi.
- Maneesh Agrawala is with UC Berkeley.

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

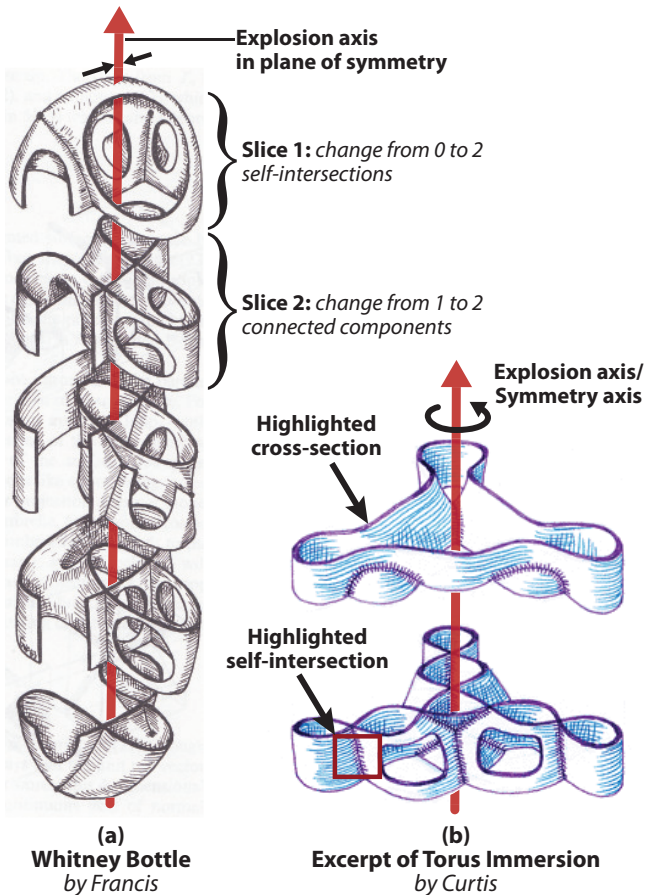


Fig. 2. **Hand-designed exploded view illustrations.** These example exploded views exhibit a few common conventions. In the Whitney Bottle illustration, the explosion axis lies in a plane of bilateral symmetry, and the first two slices show changes in the number of self-intersections and connected components of the cross-section (a). In the Torus Immersion example, the explosion axis coincides with the axis of rotational symmetry; also, self-intersections and visible cross-sections are emphasized in the final depiction (b).

There are two key challenges in creating effective exploded views: i) choosing the orientation of the explosion axis, and ii) positioning the cutting planes along the chosen axis to define the slices. Choosing the explosion axis sub-optimally can produce slices with unnecessarily complex geometry that do not convey important properties of the object’s shape, such as symmetries (see Figure 3). Choosing the cutting plane locations poorly is also undesirable. If the cutting planes are so far apart that many different geometric features appear or disappear within a single slice, it may be difficult to understand all the changes within that portion of the surface. On the other hand, if the cutting planes are too close together, adjacent cross-sections may look very similar to one another making it difficult for the viewer to identify the slices of geometric or topological interest. We examined hand-designed illustrations by Curtis [10] and by Francis [15] and found that in the best exploded views, the orientation, position and number of cutting planes are carefully chosen to convey the object’s overall structure and key geometric features in a clear and concise manner. Given the challenges in creating such illustrations, it is not surprising that well designed exploded view visualizations exist for just a tiny fraction of the wide variety of complicated surfaces that mathematicians study.

In this paper, we present an automated approach for generating exploded views of complicated mathematical objects. Our method takes as input triangle meshes, which are often easier to find (e.g., in on-

line repositories of 3D models [1]) and create them parametric surface representations. Our contributions include a set of prescriptive design rules for choosing an explosion axis and placing cutting planes. We developed these rules by analyzing several hand-designed exploded views and identifying the characteristics that make them effective. We also present a computational procedure to apply these design rules in order to generate an exploded view. The algorithm encodes the rules based on geometric properties, which we directly compute from the input polygon mesh. Our method does not depend on the specific parameterization of the surface. Using our approach, we are able to generate exploded views that expose and emphasize important geometric and topological features of mathematical surfaces, and help convey the overall shape and structure of the depicted object.

2 RELATED WORK

In mathematical visualization, the most common techniques for exposing the internal structure of surfaces are transparency, cutaways, and “banded views”, a type of cutaway that removes alternating bands from the surface [3]. There are several tradeoffs between these methods and exploded views. Making the entire surface semi-transparent is arguably the simplest technique for (partially) exposing occluded portions of the surface. While such transparency can sometimes convey the overall complexity of the surface, it can be difficult for viewers to distinguish and interpret the surface geometry in regions where multiple semi-transparent layers overlap (Figure 3b). Thus, cutaways, banded views, and exploded views are often more effective for visualizing surfaces with complex internal structure. Another key difference between these methods is that banded views and cutaways remove portions of the surface to expose internal structure, whereas exploded views include all of the surface geometry. On the other hand, exploded views distort spatial relationships by separating portions of the surface, whereas banded views and cutaways keep all of the geometry in place. As we discuss in Section 6, there may be benefits to combining these techniques in a single visualization. However, in this paper, we focus on an approach for automatically generating exploded views.

Outside the domain of mathematical visualization, there is a large body of existing work on conveying the internal structure of other types of 3D objects, e.g., CAD models, anatomical data, etc. While many of these methods focus on transparency [13, 31] and cutaways [6, 8, 9, 11, 22, 28], we limit this discussion to previous approaches for creating exploded views. Some existing systems require the user to specify certain explosion parameters in order to create interactive exploded views. Notable examples include McGuffin et al.’s interactive browsing widgets [24], Li et al.’s image-based exploded views [21], and Bruckner et al.’s force-based volumetric exploded views [7]. Fully automatic systems for generating exploded views focus primarily on CAD and architectural models, including work by Agrawala et al. [2], Driskill et al. [12], Li et al. [20], and Niederauer et al. [27]. While such methods can produce compelling visualizations, most of these techniques assume input models that have distinct parts or layers and come with auxiliary annotations. Thus, the techniques are not suitable for visualizing mathematical objects, which typically consist of a single 3D surface.

A key component of our approach is an automated algorithm for partitioning mathematical surfaces into multiple slices that are then separated to create the exploded view. A few previous approaches also use geometric analysis to determine where to cut the object into slabs. Mori et al. [26] propose an automatic technique for generating a characteristic cross-section of a given volume by analyzing its topological structure. Ruiz et al. [30] extend this idea and propose a similarity-based approach to identify the most informative cross-sections within a volumetric dataset; they then use these cross-sections to generate exploded views. Niederauer et al. [27] apply a similar high-level approach to visualize architectural scenes; they analyze the scene geometry to locate individual stories that are then rendered separately in an exploded view. Our system also uses geometric analysis to place cutting planes in order to form exploded views, but our approach directly analyzes the cross-sectional geometry of the input surface to determine where to make such cuts.

3 DESIGN RULES

Although a handful of books on mathematical drawing techniques exist [3, 15], they do not offer explicit guidelines on how to create exploded view illustrations. Moreover, because they are difficult to create, mathematical exploded view illustrations are not very common. Nevertheless, we have found several good examples of such illustrations by Curtis [10] and by Francis [15]. Based on our analysis of these illustrations, we propose a set of design rules that can guide how to *choose an explosion axis*, *place cutting planes*, and *render slices* to create effective exploded view visualizations.

Choosing the explosion axis

Many mathematical surfaces exhibit rotational symmetry, e.g., Boy's Surface, Roman Surface. Illustrators often explode mathematical surfaces along such symmetry axes (see Figure 2b) because it results in

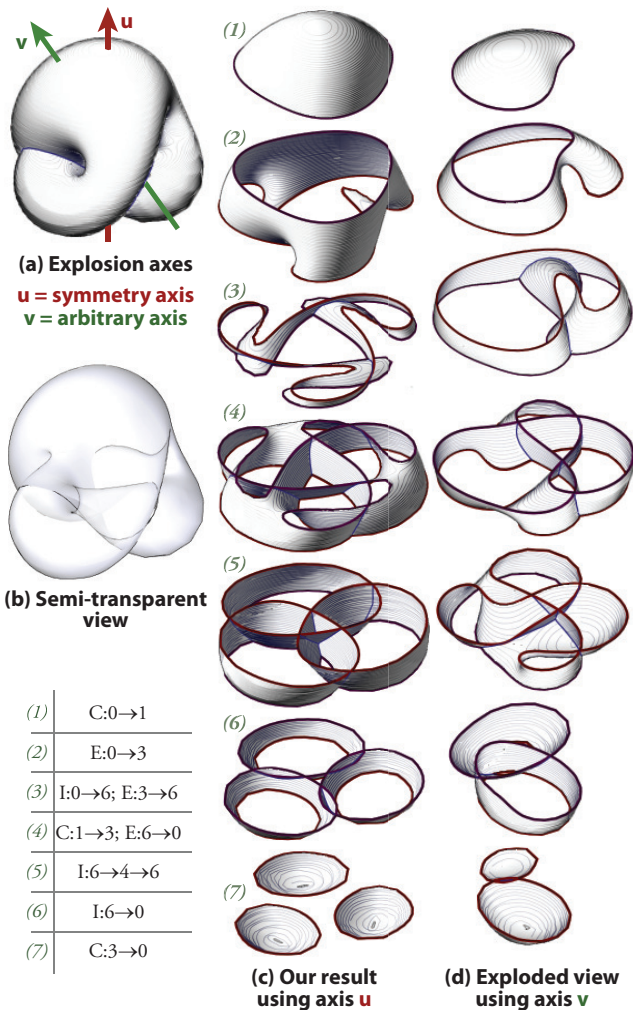


Fig. 3. **Choosing explosion axis based on symmetry.** (a) The complicated internal structure of the Boy's surface, which is obtained by sewing a Möbius strip to the edge of a disk, is mostly hidden in a normal rendering. (b) A semi-transparent view helps marginally as ambiguity in the depth ordering and intersection pattern remains. (c) An exploded view generated using our algorithm not only reveals the internal structure, but also the rotationally symmetric slices emphasize the object's overall symmetry. The table indicates the geometric change(s) within each slice (see Section 4). (d) In contrast, an exploded view along an arbitrary explosion axis v with the same number but uniformly spaced slices does not convey the object's inherent symmetry and fails to highlight key geometric features within the surface.

cross-sectional slices that are also rotationally symmetric, and thus emphasize the overall symmetry of the object. Furthermore, it can be easier for viewers to understand the shape of symmetric slices because they have repeated, regular geometry. Similarly, for objects with bilateral reflective symmetry, but no rotational symmetry, illustrators typically choose an explosion axis that lies in the plane of symmetry. The exact axis orientation is often chosen based on the dimensions of the surface projected onto the symmetry plane; aligning the axis with the direction of longest extent produces an exploded view that shows how the surface varies along its longest dimension (see Figure 2a). For objects without any prominent symmetries, illustrators vary in their choice of exploded view direction, sometimes just using the axis of longest extent as the explosion axis.

Placing cutting planes

Given an explosion axis, illustrators partition the surface into slices by placing parallel cutting planes along the explosion axis. The main goal is to expose planar cross-sections of the surface that help viewers understand how the shape of the surface varies along the explosion direction. Viewers should be able to see how the cross-section evolves from slice to slice. Simply placing cutting planes at regularly spaced intervals fails to highlight important geometric properties such as self-intersections and critical points (see Figure 3d). Instead, illustrators take into account the surface geometry and place cutting planes to emphasize changes in the following geometric attributes of the cross-sections:

- **Connected components:** Changes in the number of connected components occur at critical points where regions of the surface merge, pull apart, appear or disappear with respect to the explosion axis.
- **Self-intersections:** Changes in the number of self-intersections occur where the surface passes through itself.
- **Sharp curvature extrema¹:** Changes in the number of sharp curvature extrema indicate a significant change in the shape of the cross-section. Since curvature extrema often correspond to distinguishable feature points of a shape [4, 33], a change in the number of sharp curvature extrema indicates where it becomes more difficult for a human viewer to establish visual correspondence between consecutive cross-sections. If the correspondence is difficult to establish, this suggests that the shape of the cross-section has changed significantly.

To help viewers understand these geometric variations, illustrators place cutting planes on either side of each change, thereby exposing the surface cross-section both before and after the change occurs. This approach also emphasizes where self-intersections and critical points arise by isolating them in separate slices (see Figure 2a). Moreover, since each subsequent cross-section shows one significant change, the final exploded view contains no redundant cross-sections presenting identical information.

Rendering

Mathematical illustrations are often rendered in a simple pen-and-ink style, in which silhouettes and sharp creases are drawn with lines and hatching is used to indicate shading. In addition, illustrators typically emphasize the cross-sectional curves in each slice and self-intersection curves using a contrasting highlight color. In Figure 2b, Curtis uses dark purple lines to highlight both cross-sections and self-intersections.

¹In our system, a vertex of a polygonal cross-sectional curve is a *sharp curvature extremum*, if its discrete curvature value is larger than a certain threshold and greater than discrete curvature values of its neighbors by another threshold.

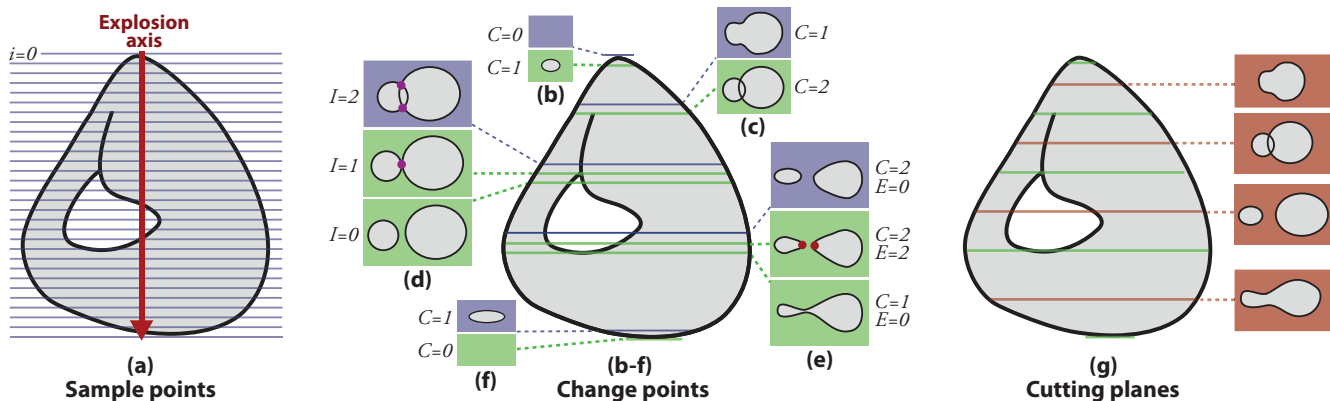


Fig. 4. **Placing cutting planes.** First, we uniformly sample the explosion axis (a). At each sample point, we compute a cross-section, as well as the number of connected components, self-intersections (purple dots), and curvature extrema (red dots). Wherever any of these values change, we mark the slice as a *change point*, shown in green (b-f). Finally, we place cutting planes halfway between each pair of adjacent stable change points, shown in pink (g).

4 OUR SYSTEM

Based on these design rules, we have developed an automated approach for generating exploded views of mathematical surfaces. The input to our system is a non-degenerate triangle mesh representing a surface immersed into 3D space.

As an initial preprocessing step, we determine whether the surface is orientable or non-orientable by examining polygon normals in a breadth-first manner; if we encounter a flipped normal, we label the surface non-orientable. Our approach consists of three main steps. Given an input triangle mesh, our system first chooses an explosion axis based on any detected symmetries in the surface. Next, our system analyzes the cross-sectional geometry of the surface to determine where to place cutting planes along the explosion axis. Finally, our system renders the slices from a user-specified viewpoint in a pen-and-ink style to create the final exploded view. We describe each of the steps in detail.

Choosing the explosion axis

To determine the explosion axis, we first detect symmetries of the input mesh using the algorithm introduced by Mitra et al. [25]. The method uses a transform domain voting scheme to identify potential symmetry candidates, which are subsequently verified and refined in the spatial domain. The algorithm is robust to noise and surface perturbations. If the analysis detects one or more rotational symmetries, our system picks the symmetry axis with the highest order of symmetry² as the explosion axis. Otherwise, if the symmetry analysis detects reflective symmetry, we project all vertices of the mesh onto the detected symmetry plane and perform Principal Component Analysis (PCA) on the projected 2D points. The algorithm then picks the largest principal component as the explosion axis for the model. If no rotational or reflective symmetry is detected, we use the first principal component of the shape as the default explosion axis. Alternatively, we allow the user to specify the explosion axis as part of the input.

Placing cutting planes

To compute the locations of the cutting planes along the explosion axis, we identify all the *change points* — i.e., the points where the number of connected components, self-intersections, or sharp curvature extrema of the surface cross-section changes. We then place cutting planes halfway between each pair of adjacent change points.

To identify the change points we start by densely sampling the explosion axis uniformly across the entire extent of the surface (Figure 4a). In our experiments we use 120 as the default number of

samples. For each position i of the cutting plane, we compute the corresponding cross-section s_i by first intersecting each triangle edge of the input mesh with a plane that passes through i and is perpendicular to the explosion axis. Then we string together the intersection points with polylines using the underlying mesh connectivity for topology. Thus for each cross section we obtain a set of polygonal curves.

For each such set of cross-sectional curves, we estimate three attributes: i) the number of connected components C_i , ii) the number of self-intersections I_i , and iii) the number of sharp curvature extrema E_i . To find C_i , we count the number of polyline curves in s_i . To compute I_i , we add up the number of intersections between the curves and the number of self-intersections within each curve in s_i . Note that in algebraic geometry, a single intersection point \mathbf{p} may be counted twice if the intersecting curves are tangent at \mathbf{p} [16, pp. 74–83]. However, in our system, we count such coincident intersection points only once. Finally, to compute E_i , we regularize the polylines using Gaussian smoothing and then estimate the discrete curvature at each polyline vertex by computing the turning angle between the adjacent line segments [17]. To further reduce noise, we perform a smoothing pass on the curvature values by taking distance-weighted averages of the curvature at adjacent vertices. Using the smoothed curvature values, we label a vertex as a *sharp curvature extrema*, if its absolute curvature value is larger than a threshold ϕ and greater than the absolute curvature values of its neighbors by another threshold θ . We used default values of $\phi = 11.5^\circ$ and $\theta = 0.5^\circ$ for all examples in the paper.

Having computed this information at each cross-section, we identify a change point as any i where the neighboring cross-sections s_i and s_{i-1} have $C_i \neq C_{i-1}$, $I_i \neq I_{i-1}$, or $E_i \neq E_{i-1}$ (see Figures 4b–f).

Unstable change points. In some cases, as portions of the cross-section merge or self-intersect, the values of C , I and E may change several times over a small window of sample points, resulting in what we call *unstable change points* — change points that are almost immediately followed by another change point. Unstable change points often arise around changes in the number of self-intersections I . For example, Figure 4d shows the change points that arise as two elliptical components of the cross-section separate from each other. As the ellipses separate, I changes from 2 to 1 and then to 0 in three consecutive cross-sections, which results in an unstable change point at $I = 1$. Unstable change points can also arise around changes in the number of critical points C , as shown in Figure 4e where two curvature extrema appear immediately before the two elliptical components merge. Finally, since our system works with mesh approximations of mathematical models, unstable change points sometimes arise due to numerical errors. For example, if on the original continuous surface E should change from 1 to 3, our system, due to numerical instability, may find three consecutive cross-sections where E changes from 1 to 2, and then to 3.

²A shape S has n -order rotational symmetry with respect to a particular axis \mathbf{u} if S is invariant under any rotation about \mathbf{u} by a multiple of $360/n$ degrees.

Even though multiple change points arise in these three scenarios, in each case there is a single stable change that represents the important visual change in the cross-section: in Figure 4d, I changes from 2 to 0 as the ellipses separate; in Figure 4e, C changes from 2 to 1 as the two components merge into one; and in the final case, E changes from 1 to 3 as the shape of the cross-section changes. To ensure that every change point corresponds to exactly one important visual change, we eliminate unstable change points. Specifically, we detect clusters of nearby change points and replace them with a single change point that is centered within the window of change. We consider adjacent change points to be nearby if they are closer than 2% of the total extent of the surface measured along the explosion axis. Figure 4g illustrates how cutting planes are placed with respect to change points.

Rendering

To generate the final result, our system renders each slice in a pen-and-ink style from the specified viewpoint. We draw silhouettes and sharp creases with black lines, and render the surface itself using diffuse shading with a single directional light. We color non-orientable surfaces white, and orientable surfaces purple and blue.

We render additional shading lines on the surface of the object that run parallel to the cutting planes. The RGB color at each point \mathbf{p} along a shading line is computed using

$$(R, G, B) := (d_p S_R, d_p S_G, d_p A_p S_B) \quad \text{where, } d_p = |L \cdot N_p|$$

and (S_R, S_G, S_B) is the RGB color of the surface, A_p is the ambient occlusion at \mathbf{p} , L is the light direction, and N_p is the surface normal at \mathbf{p} . The diffuse shading component d_p ensures that shading lines are darker where the surface normal bends away from the light direction, and the ambient occlusion component A_p makes the lines more blue on internal portions of the surface. On external portions of the surface, shading lines are mostly grey because there is typically a small amount of ambient occlusion.

Our system also detects and emphasizes self-intersections and visible cross-sections. Each cross-section is drawn with a thick red or purple line. We choose the color using an alternating pattern such that the two facing cross-sections of adjacent slices are always rendered in the same color. Self-intersections are drawn using blue lines.

5 RESULTS

We found several sources of complicated mathematical surfaces online [18, 32], for the majority of which good hand-drawn diagrams are not available. We used our system to generate exploded view illustrations of several of these surfaces (Figures 3, 5 and 8). For all of the models, computing the cutting plane locations required less than three minutes on a MacBook Pro with a 2.6 GHz Intel Core 2 Duo and 4GB of memory. The size of the meshes varied from 5K to 60K triangles.

Boy's Surface is an immersed projective plane that is constructed by sewing a Möbius strip to the edge of a disk; the model shown in this paper minimizes Willmore elastic bending energy. The Slippers Surface is a surface designed by Roger Bagula that resembles (from some angles) a pair of slippers [5]. The Roman Surface is another mapping of the real projective plane into 3D that exhibits a high degree of symmetry, including 3-fold rotational symmetry. We also generated a variant of Roman that only has 2-fold symmetry³. We created the "Flower" Surface by inflating and indenting the surface of a sphere using regularly spaced Gaussian lobes. Everting Spheres 1 and 2 are surfaces obtained at two different stages in the temporal sequence of a sphere eversion [14] [29]; we generated these meshes using McGuffin's software [23] that implements Thurston's realization of the sphere eversion [19]. All of these surfaces exhibit symmetry: it is reflective symmetry in the case of Slippers Surface and rotational symmetry for the other surfaces. The explosion axis was chosen automatically by the system for all of the results. The only user input required in these examples is setting the viewpoint.

³ $\mathbf{r}(u, v) = (\sin(2u) \sin(v^2), \cos(u) \sin(2v), \sin(u) \cos(2v))$,
with $0 \leq u \leq \pi$ and $-\pi/2 \leq v \leq \pi/2$.

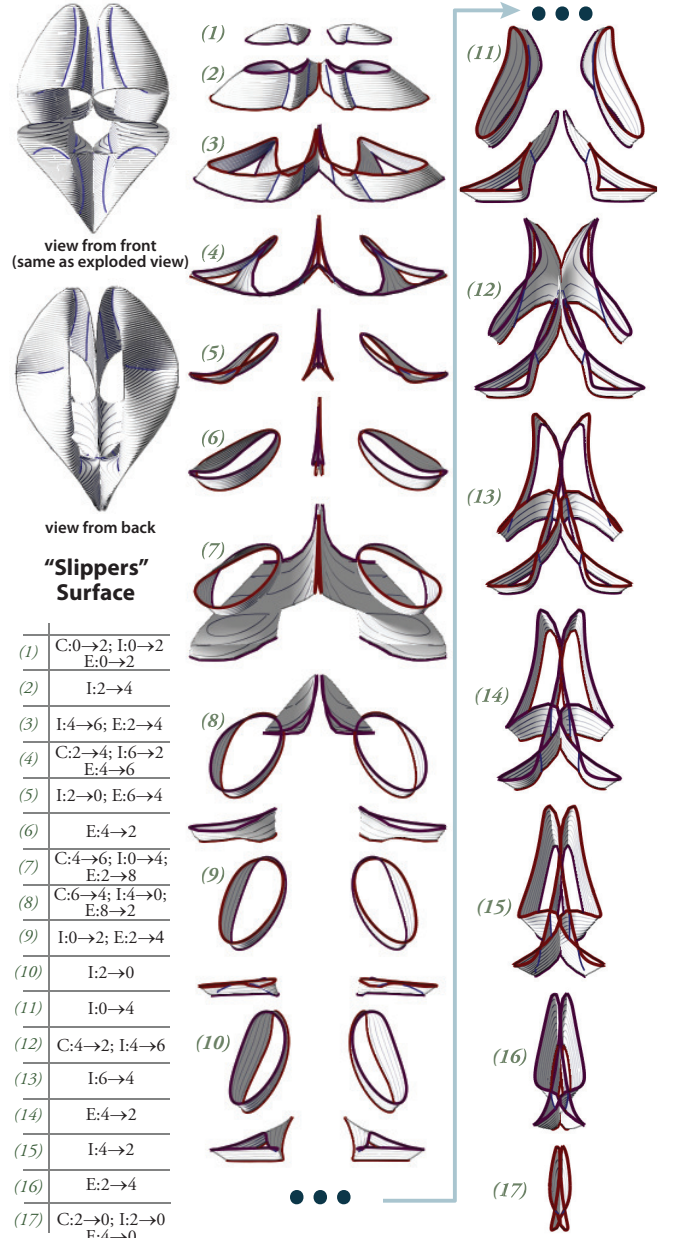


Fig. 5. **Slippers surface.** At the top, we show two views of the surface. The exploded view contains 17 slices that we display in two columns. The table indicates what change(s) happen at each slice.

Boy's Surface has several self-intersections that make it difficult to understand how the surface twists and folds back on itself. Our exploded view shown in Figure 3 exposes several important cross-sections that help the viewer understand where these self-intersections occur and how the cross-section evolves. For example, slice 3 shows that the surface passes through itself in three places. Slice 4 shows how a single cross-section component turns into three; the three "arms" of the cross-section at the top of slice 4 merge to form three closed loops.

Slippers Surface is a complicated surface where many topological changes occur in the cross-section as the plane is swept along the explosion axis (see Figure 5). It is interesting to note that for several slices in the exploded view (slices 4, 7, 8) all three geometric attributes change within the same slice. For surfaces like Slippers where the cross-section undergoes many changes along the explosion axis, our

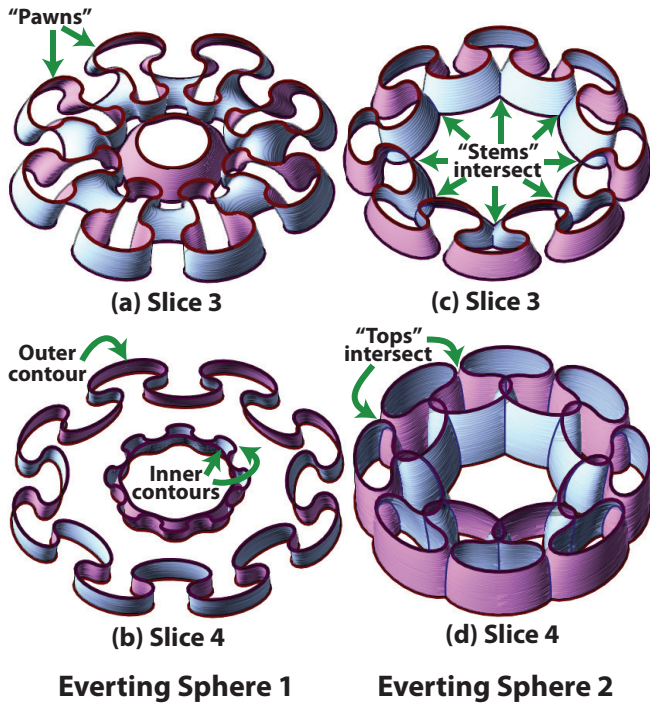


Fig. 6. **Zoomed in slices.** These images show specific features of slices from Everting Sphere 1 (a–b) and Everting Sphere 2 (c–d).

system detects a large number of change points, which produces an exploded view with many slices. Although these slices highlight all of the key geometric variations along the explosion axis, they also add visual clutter to the illustration.

Tetrahedral symmetry of the Roman surface is emphasized in the exploded view generated with our approach (see Figure 8). Three out of the four “lobes” located in the corners of a tetrahedron are visible in the exposed cross-sections.

Roman variant does not exhibit 3-fold symmetry, but is 2-fold rotationally symmetric about the y axis and reflective symmetric about xy and yz planes. Slice 2 of the exploded view shows how four “lobes” intersecting along a line move apart to form a cross-like shape as the number of intersections changes from 1 to 5.

The “Flower” Surface demonstrates the importance of using curvature extrema points for detecting significant changes in the cross-section. The shape of the cross-section changes significantly over the shape, yet it contains no other geometric changes (in the number of self-intersections or connected components). Slice 2 shows the cross-section changing into a (roughly) triangular shape as three prominent lobes begin to appear. This causes E to change from 0 to 3. In the next slice, as E changes from 3 to 6, three more lobes grow from the edges of the triangle and turn the cross-section into a six-sided star.

Several transitions occur in Everting Sphere 1. Slice 3 shows the change in the number of connected components: the 8 “pawns” around the perimeter of the cross-section change shape to merge with each other, forming two contours instead of eight (Figure 6a). Slice 4 demonstrates a change in the number of self-intersections — the two internal contours collapse onto each other (Figure 6b). Note that because of the symmetry present in the shape, the top and bottom halves of the exploded view are mirror images of each other.

Everting Sphere 2 contains many self-intersections. First, in slice 3 the number of self-intersections changes from 0 to 16 when the “stems” of the 8 components intersect each other (Figure 6c). Slice 4 shows how the number of self-intersections changes from 16 to 32 as the “top” of the 8 components intersect as well (Figure 6d).

LIMITATIONS

Although our system creates effective visualizations for a class of mathematical surfaces, our approach does have some limitations.

First, we use a single linear explosion axis to generate exploded views. However, some surfaces have a curved axis of symmetry (e.g., the medial axis of a generalized cylinder). Using such a curved axis as the explosion axis would produce regular, symmetric slices that emphasize the overall symmetry of the object. Unfortunately, since the slices would no longer be parallel, neighboring slices might intersect. It is not clear how to handle this issue.

Second, our approach for detecting significant changes in the cross-section of the surface only considers changes in the *number* of connected components C , self-intersections I , and curvature extrema E . As a result, we do not detect certain types of geometric variations. For instance, in some cases C , I and E can stay constant while the *positions* of self-intersections, curvature extrema or the cross-section components themselves change significantly as in the case of a twisting surface. Such positional changes can produce major variations in the relative positions of connected components that result in significant self-occlusions between parts of the surface. Since we do not detect such changes, our exploded views would not expose these variations.

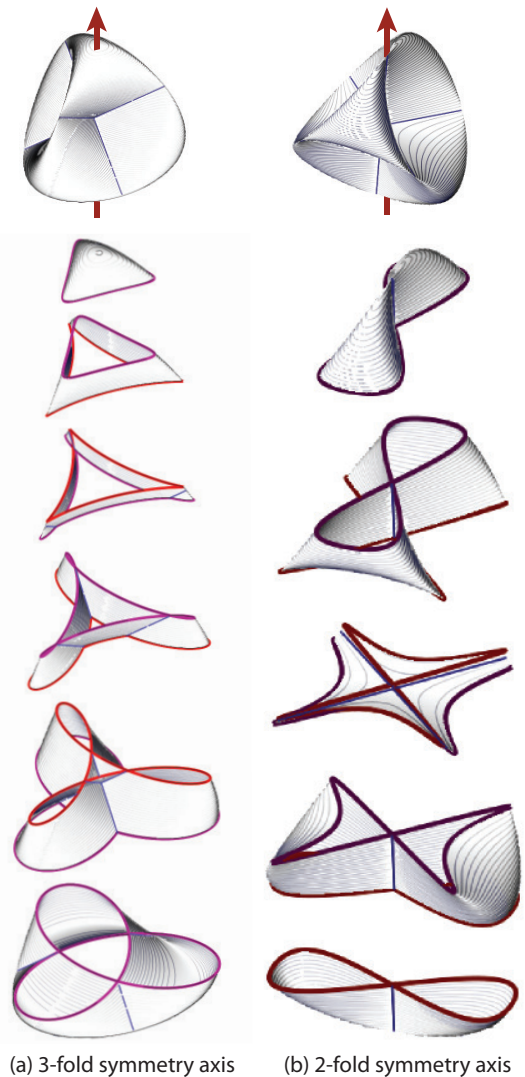


Fig. 7. **Multiple symmetries.** The Roman Surface (also shown in Figure 8) has both 3-fold and 2-fold rotational symmetries. Our system automatically uses the 3-fold axis to generate an exploded view (a). However, our algorithm also detects the 2-fold axis, which can be used to produce a different visualization (b).

Our sampling approach may miss some important geometric changes (e.g., unstable intersections where I goes from 6 to 4 to 6). We could implement intersection detection in a more stable fashion by computing self-intersections using a thin band of the surface around each sample location along the explosion axis. Alternatively, we could allow users to control the spacing adaptively in an interactive system.

Finally, our system computes the exploded view diagram for a given immersion of the surface. Mathematicians using this visualization tool might be more interested in the surface itself and not in a particular immersion. Deriving a good immersion of the surface is an important problem that we leave to future work.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose several design rules for generating exploded views of mathematical surfaces. These rules provide guidance on how to choose an explosion axis, place cutting planes, and render slices. We also present an automated system based on these design principles that we used to generate effective illustrations of several complicated surfaces. We see several directions for future work:

Evaluation. Although our results have similar characteristics to the hand-drawn exploded views that we analyzed, we have not done a formal evaluation of our visualizations. Ideally, such an evaluation would help us understand how well our illustrations convey the internal structure of mathematical surfaces to human viewers. Another interesting question to address is how the choice of explosion axis for objects with multiple symmetries affects the quality of the final result. For example, the Roman surface has both 3-fold and 2-fold rotational symmetries. Our algorithm automatically chooses the 3-fold axis as the explosion axis, but as shown in Figure 7, the 2-fold axis also produces a reasonable exploded view. We leave this type of evaluation for future work.

Incorporating window cuts. In the future, we plan to add window cuts to our system. Window cuts are often incorporated in illustrations of mathematical objects (see Figure 2a) [15] to expose occluded regions or features of interest. Although our exploded views show many important geometric features of the surface in the top cross-sections of slices, window cuts could help to expose regions of interest within a single slice.

Interactive visualizations. The primary goal of our work is to create effective static illustrations of mathematical surfaces. However, our approach could also be incorporated into an interactive application for viewing 3D shapes. One simple extension is to enable constrained direct manipulation of slices along the explosion axis, as described by Li et al. [20]. Another potential approach is to present the user with an automatically generated exploded view diagram of the shape, where interesting geometric features are emphasized and important cross-sections are shown in the context of the shape. Then, the user can further explore the shape by interactively dragging the cutting plane within each slice, or by changing the placement of cuts.

ACKNOWLEDGMENTS: This work was partially supported by NSF grant CCF-0643552. Olga Karpenko's research was funded by AAUW postdoctoral fellowship. Niloy Mitra was partially supported by a Microsoft outstanding young faculty scholarship.

REFERENCES

- [1] Google 3D Warehouse. <http://www.google.com/sketchup/3dwh/>.
- [2] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, and B. Tversky. Designing effective step-by-step assembly instructions. *ACM Transactions on Graphics*, 22(3):828–837, 2003.
- [3] T. Banchoff. *Beyond the third dimension : geometry, computer graphics, and higher dimensions*. Scientific American Library : Distributed by W.H. Freeman, New York, 1990.
- [4] W. V. Baxter, P. Barla, and K.-i. Anjyo. Compatible embedding for 2D shape animation. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):867–879, 2009.
- [5] P. Bourke. Geometry, surfaces, curves, polyhedra. <http://local.wasp.uwa.edu.au/pbourke/geometry/>.
- [6] S. Bruckner and M. E. Gröller. Volumeshop: An interactive system for direct volume illustration. In *Proceedings of IEEE Visualization 2005*, pages 671–678, 2005.
- [7] S. Bruckner and M. E. Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, 2006.
- [8] C. D. Bruyns, S. Senger, A. Menon, K. Montgomery, S. Wildermuth, and R. Boyle. A survey of interactive mesh-cutting techniques and a new method for implementing generalized interactive mesh cutting using virtual tools. *The Journal of Visualization and Computer Animation*, 13(1):21–42, 2002.
- [9] M. Burns and A. Finkelstein. Adaptive cutaways for comprehensible rendering of polygonal scenes. *ACM Transactions on Graphics*, 27(5):1–7, 2008.
- [10] C. Curtis. A strange immersion of the torus in 3-space. website, 1992. <http://www.otherthings.com/uw/torus.html>.
- [11] J. Diepstraten, D. Weiskopf, and T. Ertl. Interactive cutaway illustrations. In *Computer Graphics Forum*, pages 523–532, 2003.
- [12] E. Driskill and E. Cohen. Interactive design, analysis, and illustration of assemblies. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 27–34, 1995.
- [13] S. Feiner and D. D. Seligmann. Cutaways and ghosting: satisfying visibility constraints in dynamic 3D illustrations. *The Visual Computer*, 8(5&6):292–302, 1992.
- [14] G. Francis and J. M. Sullivan. Visualizing a sphere eversion. *IEEE Transactions on Visualization and Computer Graphics*, 10(5):509–515, 2004.
- [15] G. K. Francis. *A Topological Picturebook*. Springer, 2 edition, 2007.
- [16] W. Fulton. Algebraic curves. In *Mathematics Lecture Note Series*, 1974.
- [17] E. Grinspun and A. Secord. Introduction to discrete differential geometry: the geometry of plane curves. In *ACM SIGGRAPH 2005 Courses*, 2005.
- [18] D. Hoffman and J. T. Hoffman. The scientific graphics project. <http://www.msri.org/about/sgp/jim/models/index.html>.
- [19] S. Levy, D. Maxwell, and T. Munzner. Outside in (video), 1994.
- [20] W. Li, M. Agrawala, B. Curless, and D. Salesin. Automated generation of interactive 3D exploded view diagrams. *ACM Transactions on Graphics*, 27(3):1–7, 2008.
- [21] W. Li, M. Agrawala, and D. Salesin. Interactive image-based exploded view diagrams. In *Proceedings of Graphics Interface 2004*, pages 203–212, 2004.
- [22] W. Li, L. Ritter, M. Agrawala, B. Curless, and D. Salesin. Interactive cutaway illustrations of complex 3D models. *ACM Transactions on Graphics*, 26(3):31, 2007.
- [23] M. J. McGuffin. Sphere eversion program, 2005. <http://www.dgp.toronto.edu/mjmcguff/eversion/>.
- [24] M. J. McGuffin, L. Tancu, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proc. of IEEE Visualization*, page 53, 2003.
- [25] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics*, 25(3):560–568, 2006.
- [26] Y. Mori, S. Takahashi, T. Igarashi, Y. Takeshima, and I. Fujishiro. Automatic cross-sectioning based on topological volume skeletonization. In *Smart Graphics*, pages 175–184, 2005.
- [27] C. Niederauer, M. Houston, M. Agrawala, and G. Humphreys. Non-invasive interactive visualization of dynamic architectural environments. In *Proc. of the symp. on Interactive 3D graphics*, pages 55–58, 2003.
- [28] S. Owada, F. Nielsen, M. Okabe, and T. Igarashi. Volumetric illustration: designing 3D models with internal textures. *ACM Transactions on Graphics*, 23(3):322–328, 2004.
- [29] A. Phillips. Turning a sphere inside out. *Scientific American*, 214:112–120, 1966.
- [30] M. Ruiz, I. Viola, I. Boada, S. Bruckner, M. Feixas, and M. Sbert. Similarity-based exploded views. In *Proceedings of Symp. on Smart Graphics*, pages 154–165, 2008.
- [31] I. Viola, A. Kanitsar, and M. E. Gröller. Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):408–418, 2005.
- [32] E. Weisstein. Eric weisstains world of mathematics. Wolfram Research, <http://mathworld.wolfram.com>.
- [33] X. Zabulis, J. Sporring, and S. C. Orphanoudakis. Perceptually relevant and piecewise linear matching of silhouettes. *Pattern Recognition*, 38(1):75–93, 2005.

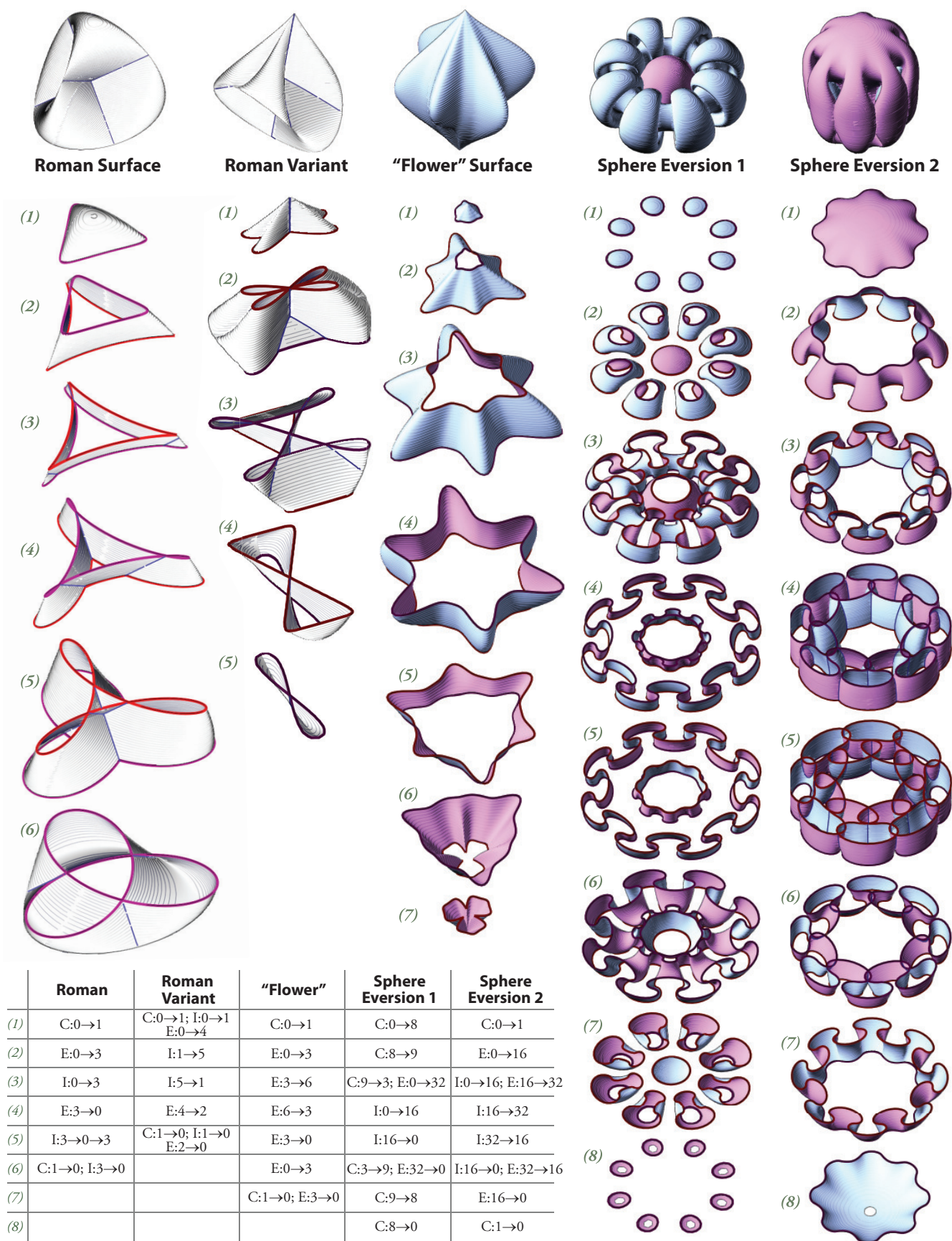


Fig. 8. Summary of results. Each column of images shows the input surface at the top and our exploded view below. The table indicates what change(s) each slice shows. Our system colors non-orientable surfaces white, and orientable surfaces purple and blue.