

# Field-Aligned Mesh Joinery

PAOLO CIGNONI and NICO PIETRONI

CNR - ISTI

LUIGI MALOMO

University of Pisa

and

ROBERTO SCOPIGNO

CNR - ISTI

Mesh joinery is an innovative method to produce illustrative shape approximations suitable for fabrication. Mesh joinery is capable of producing complex fabricable structures in an efficient and visually pleasing manner. We represent an input geometry as a set of planar pieces arranged to compose a rigid structure, by exploiting an efficient slit mechanism. Since slices are planar, to fabricate them a standard 2D cutting system is enough.

We automatically arrange slices according to a smooth cross-field defined over the surface. Cross-fields allow representing global features that characterize the appearance of the shape. Slice placement conforms to specific manufacturing constraints.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Physically based modeling*

General Terms: Algorithms, Design

Additional Key Words and Phrases: Geometry processing, object fabrication, manufacturing

## ACM Reference Format:

Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. 2014. Field-aligned mesh joinery. ACM Trans. Graph 33, 1, Article 11 (January 2014) 12 pages.

DOI: <http://dx.doi.org/10.1145/2537852>

## 1. INTRODUCTION

In this article we introduce mesh joinery, a novel and practical approach to fabricate artistic illustrative shape approximations made

---

The research leading to these results was partially funded by EU FP7 project ICT FET Harvest4D (<http://www.harvest4d.org/>, G.A. no. 323567).

Authors' addresses: P. Cignoni (corresponding author) and N. Pietroni, Visual Computing Lab, CNR-ISTI, Italy; email: cignoni@isti.cnr.it; L. Malomo, Computer Science Department, University of Pisa, Italy; R. Scopigno, Visual Computing Lab, CNR-ISTI, Italy.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2014 ACM 0730-0301/2014/01-ART11 \$15.00

DOI: <http://dx.doi.org/10.1145/2537852>

up of several interlocked planar pieces, called slices. Such slices can be easily fabricated using any 2D cutting device and then assembled through a sequence of manual operations.

Compared to previous approaches (such as McCrae et al. [2011], Hildebrand et al. [2012], and Schwartzburg and Pauly [2012]) we oriented the slices according to a given cross-field defined on the surface. As most of the recent quadrangulation papers have shown [Ray et al. 2006; Kälberer et al. 2007; Bommes et al. 2009, 2012; Pietroni et al. 2011], cross-fields are an excellent instrument for capturing the global structure of a given shape.

We provide a novel formalism to design a slice-to-slice interlocking system. This formalism provides enough degrees of freedom to follow complex cross-fields and, consequently, to efficiently approximate the global structure that characterizes the input shape. Additionally, we ensure a sufficient degree of physical stability of the final structure along with the sequence of manual operations required for the assembly procedure.

Our approach provides limited but low-cost solutions due to the simple cutting technologies employed and the relatively inexpensive material used (such as cardboard). Although the proposed slice structure approximates, to some extent, the original geometry, it cannot be considered as a “physical copy”. Nevertheless, we believe that our approach could be attractive in specific markets, such as in artistic or illustrative contexts, in puzzles or toys, and where assembly is a key part of user experience.

## 1.1 Motivation

Rapid prototyping [Dimitrov et al. 2006] has been developed over the last decade to support the manufacturing process, especially for the production-quality parts in relatively small numbers. It exploits a wide variety of basic technologies to create real-world tangible reproductions from 3D digital models. While initially the range of materials was very limited, modern technologies enable a wide range of materials (plastic, glued gypsum, steel, ceramic, stone, wood, etc.) to be used. At the same time, the printing resolution has improved substantially and, consequently, accuracy in terms of reproduction has reached high standards. Nevertheless, rapid prototyping is still perceived as being too expensive for the mass market. Moreover, the input geometry has to satisfy certain geometric characteristics (manifoldness, watertightness, etc.) and static mechanical properties, in order to produce a compact, high-quality, fabricated model that is free of artifacts.

A few years ago radically new paradigms for shape fabrication were proposed [Mitani and Suzuki 2004; Shatz et al. 2006; Massarwi et al. 2007; Mori and Igarashi 2007; Li et al. 2010]. The main idea was to drastically simplify the overall printing procedure by fabricating a plausible representation of the digital model, instead of its exact copy. This class of methods relies on a simple



Fig. 1. Given a 3D shape with a smooth cross-field, we generate a set of planar slices that can be interlocked in a self-supporting structure.

concept: approximating an object does not necessarily mean that there will be a visual deficit.

A recent approach proposed approximating the surface using an orthogonal arrangement of planar pieces [Hildebrand et al. 2012]. The slices are plugged into each other to compose a rigid shape.

## 1.2 Contributions

We redesigned the traditional slice interlocking approach in order to approximate generic 3D surfaces with greater flexibility. We focused on building arrangements composed of shallow ribbon-shaped pieces which follow a cross-field defined on the surface. These structures are made up of planar pieces that interlock with each other using an extended slit mechanism. Specifically, our contributions are as follows.

—We propose a novel strategy to fabricate illustrative shape approximations based on ribbon-shaped planar slices. Compared to classical planar sections [Hildebrand et al. 2012], ribbon-shaped slices reduce the physical constraints involved in the assembling procedure, allowing for more complex structures.

—We extend the classical slit mechanism [Hildebrand et al. 2012] by providing additional structural degrees of freedom. In particular, we consider insertion movements that are not orthogonal to slices. In addition, we formulated nonorthogonal slice placement [McCrae et al. 2011; Schwartzburg and Pauly 2012] in a novel, structurally sound perspective. We have demonstrated how these additional degrees of freedom can be exploited to efficiently represent complex models.

—We propose a novel, efficient strategy to approximate a surface with a set of slices. Slice placement is driven by an input cross-field (such as Hertzmann and Zorin [2000], Bommes et al. [2009], and Ray et al. [2009]). It provides a set of appealing, uniformly distributed polylines lying on the surface of a mesh. In addition, the method also takes into account slice insertion constraints and, while it does not theoretically guarantee that the mounting sequence is collision free, it yields arrangements that are practically assemblable and that exhibit a sufficiently robust slice structure. Our method may also take advantage of field symmetrization techniques, such as Panizzo et al. [2012] (see Figure 2) for a better perception of the global structure of the generated structure.

—We propose an automatic procedure to ensure that the slice structure is physically achievable. First, it improves the final rigidity, acting upon the slit interlocking mechanism. Second, it ensures

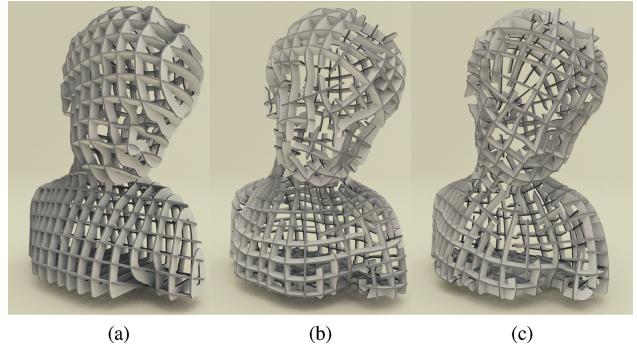


Fig. 2. (a) The classical waffle approach modeling technique (with axis-aligned slices); (b) our method applied to a cross-field calculated with Bommes et al. [2009]; (c) field symmetrization techniques [Panizzo et al. 2012] increase the visual appeal of the final result. The total length of the polylines for each method is approximately the same.

that the slice structure conforms to the physical constraints required by the manual assembling procedure. This procedure is specifically designed to deal with our extended slit mechanism.

## 2. RELATED WORK

Fabricating tangible models from a digital 3D shape is fundamental in many industrial production processes. The majority of current applications require a high level of accuracy, that is, the printed model needs to be a highly accurate physical copy of the digital shape. For example, several applications require this level of accuracy for aesthetic purposes or for performing functional tests. However, different contexts (toys, artistic reproductions) do not require the same level of accuracy, or even prefer the production of an illustrative version of the digital model.

On the basis of accuracy and reproduction we can classify the various methods into two broad categories.

—*Accurate*. Modern devices enable almost exact copies of a given shape to be reproduced. To guarantee high reproduction accuracy, the printer and the reproduction material may both be expensive.

—*Illustrative*. These methodologies fabricate approximate copies of a given object, usually by relying on standard and inexpensive printing technologies.

In both categories, the model can be fabricated as a single piece or it can be split into a set of separate pieces and assembled afterwards.

## 2.1 Accurate Methods

Rapid prototyping techniques [Dimitrov et al. 2006] have been created to support the design industry. Usually the digital model needs to be represented as a closed, piecewise, manifold mesh. Due to the physical properties of the material employed and the production procedure, specific mechanical constraints must be satisfied. These constraints guarantee that the model is kept physically compact throughout the printing procedure.

Recent research has focused on how to acquire the physical properties of a real object to transplant onto the fabricated model. For example, Bickel et al. [2010] proposed a technique to match the elastic properties of a given object. Other papers focus on appearance properties: Cignoni et al. [2008] proposed a technique to enhance colors for rapid prototyping; Weyrich et al. [2009] and Matusik et al. [2009] reported a method for the improved reproducibility of surface reflectance properties by adding microgeometry; and Hašan et al. [2010] and Dong et al. [2010] proposed a technique to print specific subsurface scattering characteristics.

One common strategy is to divide up the original shape into different components, which are fabricated separately but assembled together to produce the desired shape. One example is architectural modeling, where the original shape is subdivided into a finite set of triangular [Singh and Schaefer 2010] or quadrilateral [Fu et al. 2010; Eigensatz et al. 2010] basic panels. A method to fit a freeform shape with a set of single direction bendable panels (like wooden panels) is proposed in Pottmann et al. [2010]. To further improve the smoothness of freeform surfaces in architectural design, Bo et al. [2011] introduced the so-called circular arc structures.

In architecture, the decomposition of an object is usually mandatory, and depends on the dimensions of the fabricated shape. Conversely, generic shapes were deliberately decomposed into small pieces to create a puzzle-like structure in Lo et al. [2009] and Xin et al. [2011].

## 2.2 Illustrative Methods

The aim of illustrative methods is to fabricate an illustrative approximation of an input digital model.

Illustrative methods are generally designed to employ materials and devices that are very popular and inexpensive. Since the fabrication process does not require a sophisticated device, a number of inexpensive, accessible servicing companies have recently flourished. The interest in these technologies is testified by the recent release of software tools devoted to planar slice fabrication procedures (such as Autodesk 123DMake [Autodesk 2013]).

For example, Mori and Igarashi [2007] proposed a sketching interface to design plush toys. Li et al. [2010, 2011] put forward a strategy to automatically fabricate pop-up models made of paper. Pop-up models can remain in two different states: open (showing the modeled shape) and closed (reduced to a simple sheet of paper). A method to fabricate a three-dimensional shape illustrated through a stack of colored slices was reported by Holroyd et al. [2011]. Finally, several methods [Mitani and Suzuki 2004; Shatz et al. 2006; Massarwi et al. 2007] represent the input model through a set of foldable strips (usually made of paper), which can be glued together to create a layered 3D representation.

McCrae et al. [2011] create shape abstractions arranging planar slices to optimize the perception of the original object. This method allows nonorthogonal slices, however, it is not designed for the

fabrication of tangible objects and problems of the assembly of these slices have not been investigated.

Recently, Hildebrand et al. [2012] proposed a method to semi-automatically fabricate objects made up of planar slices. Although this method produces a wide range of visually appealing results, unfortunately, it does not fit well with complex geometries (models with a high degree of asymmetry or even complex topology) and it favors arrangements of orthogonal slices. Similarly, Schwartzburg and Pauly [2012] allow nonorthogonal slices, but their method tries to retain the simplicity of orthogonally intersecting pieces. Recently Schwartzburg and Pauly [2013] extended their approach to provide a more detailed formulation on the assembly of nonorthogonal slices by dealing with rigidity constraints. Given a set of predefined intersecting slices, Schwartzburg and Pauly [2013] optimize slice positions to restrict the possible movement of each slice, thus maximizing the rigidity of the resulting structure.

However, as demonstrated by the results, our method is capable of automatically sampling planar slices in a visually appealing manner. Our approach captures and represents the global structure of complex objects, providing, at the same time, a fabrication strategy that meets the physical rigidity constraints.

## 3. AN OVERVIEW OF THE COMPLETE PIPELINE

Our fabrication pipeline, as shown in Figure 3, has the following steps.

- (1) As input, we get a triangle mesh with a cross-field defined on its surface (see Figure 3(a)). We obtained the cross-field using the method proposed in Bommes et al. [2009] with the symmetrization of Panozzo et al. [2012].
- (2) We sample a set of planar polylines that lies on the original surface (see Figure 3(b)). These polylines need to be oriented consistently with the cross-field and uniformly distributed on the surface of the object. At the same time, the polylines need to conform to specific constraints thus ensuring the stability of the final structure. This step is detailed in Section 5.
- (3) The polylines are transformed into a set of ribbon-shaped slices (see Figure 3(c)). These profiles are obtained through a sequence of boolean operations performed in a 2D space (using ClipperLib [Johnson 2013]).
- (4) We derive the interlocking mechanism to produce a physically stable structure. At the same time we provide the sequence of inserting gestures that make up the assembly procedure. This step requires some slices to be split/carved (highlighted by the close-up in Figure 3(d)). This step is detailed in Section 6.
- (5) Each slice is then converted to a vectorial representation and organized into sheets ready for automatic laser cutting (see Figure 3(e)).
- (6) Finally the slices are assembled by following the sequence specified by our system (see Figure 3(f)). The derivation of the assembling sequence is detailed in Section 7.

## 4. INTERLOCKING PLANAR SLICES

In this section we provide an overview of the basic concepts regarding interlocking mechanisms between planar slices. For a more general discussion on interlocking shapes, see Séquin [2012].

For the sake of simplicity, consider the simple situation of two perpendicular slices fitting together (see Figure 4). One slice moves along a line parallel to the intersection between the two slices, to fit with the other one which is fixed (this is the typical configuration of waffle meshes). For each piece we create a rectangular slit at the

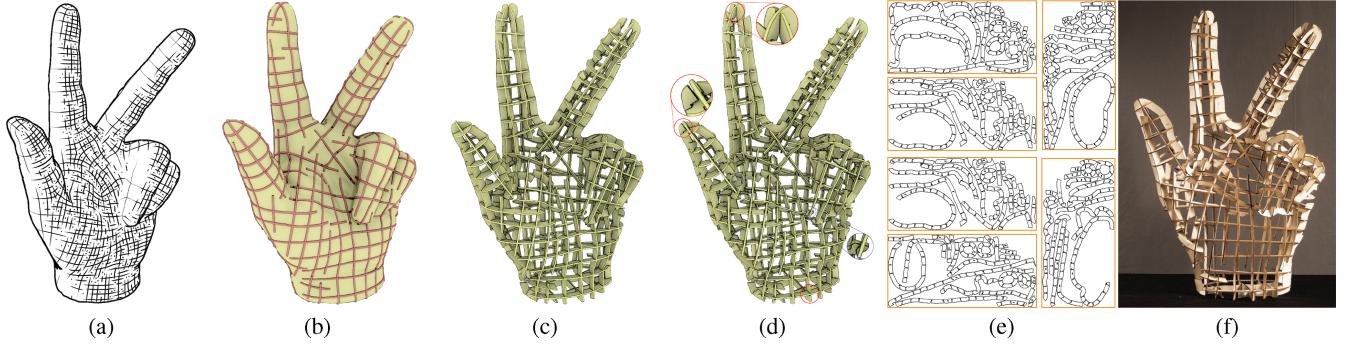


Fig. 3. A complete overview of our fabrication pipeline: (a) We get as input a triangle mesh and an associated smooth (possibly symmetric) cross-field; (b) we sample a set of well-distributed field-oriented planar polylines; (c) the polylines are transformed into ribbon-shaped slices; (d) the slice structure is modified to ensure that the final structure is physically achievable; (e) the slices are transformed into 2D vectorial profiles that are laser cut; (f) the pieces are assembled manually by following the instructions.

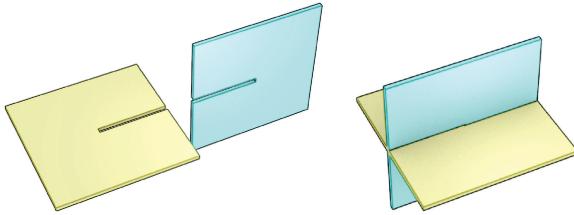


Fig. 4. The classical situation of two connected slices: for each piece we create a rectangular slit in correspondence with the intersection line.

intersection line. The width of the slit must be equal to the width of the material used to create the slicing structure.

This classical, well-known configuration is built on two hard constraints.

*Orthogonality constraint.* The angle between each pair of intersecting slices must be a right angle.

*Parallelism constraint.* For each pair of intersecting slices, the insertion movement is parallel to the segment defined by their intersection.

Conforming to these constraints means that the slice arrangement is mostly arranged as an axis-aligned grid, the well-known *waffle-shaped* configuration.

Unfortunately, orthogonality and parallelism constraints have several modeling limitations. These limitations produce serious artifacts, especially for an input shape with a low degree of axis alignment. Obviously, this reduces the range of possible shapes to which this method can be applied.

To overcome this problem (instead of increasing the sampling rate) we explicitly relax these two constraints.

#### 4.1 Relaxing the Orthogonality Constraint

The traditional slit insertion forces the two slices to be orthogonal to each other. This assembling mechanism is solid and strong because it relies on a tight grip of the slits around the slices, which ensures a firm interlock of the two pieces. If the two slices are not orthogonal, the slit has to be widened by the factor  $\lambda$

$$\lambda = (|\tan(\pi/2 - \alpha)| + 1) \cdot \tau, \quad (1)$$

where  $\tau$  is the slice thickness and  $\alpha$  is the angle between the two slice planes.

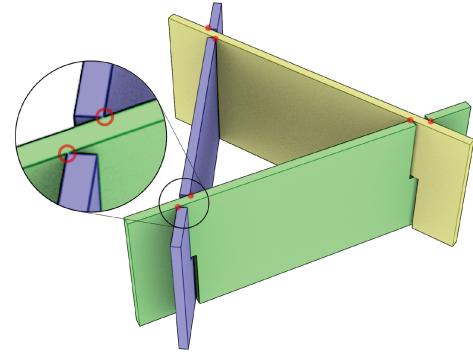


Fig. 5. Three interlocked slices are rigid and tightly connected, although the slices are not orthogonal and the wide slits are not tightly fitted onto the surface of the other slice. The red dots denote where the slices are pressed/forced against each other, such that the resulting friction ensures the stability of the structure.

On the other hand, if we consider arrangements consisting of multiple slices, the solidity of the grip can be guaranteed by a simple triangular arrangement (see Figure 5) or, alternatively, by four slices interlocked together with nonparallel intersections (see Figure 7). In the latter case, the rigidity derives from the fact that a nonorthogonal slit is like a hinge and the four connected slices form a four-bar linkage [McCarthy and Soh 2000]. Any spatial linkage formed by four links and four hinged joints, when in general position, is a highly constrained (rigid) mechanical system. Section 5 outlines how we exploit this mechanism to ensure stability in the final structure.

#### 4.2 Relaxing the Parallelism Constraint

Just allowing the angle between slice planes to deviate from  $90^\circ$  is not sufficient to deal with all the possible real scenarios. Indeed, as illustrated in Figure 6, when a slice (the green one) has to be inserted over four existing nonparallel slices (the blue ones), the direction of insertion will definitely not be parallel to some of the intersections. In these cases the slit has to be enlarged so that it can accommodate the insertion movement. The size and shape of the widened slit (trapezoidally-shaped) depend on the chosen direction for the insertion.

Guaranteeing that the inserted piece has a firm grip is important, so an insertion direction that is parallel with at least one of the

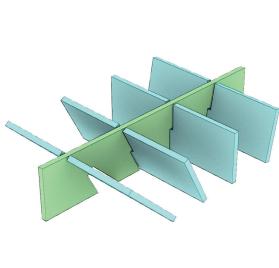


Fig. 6. The shape of the slit widening depends on the insertion direction. The divergence of the green slice is the maximum angle between the various intersection segments when the best insertion direction is chosen. On the right we show how the slit widening varies when different insert directions are chosen.

intersection segments is required, so that at least one of the slits holds the other piece steadily.

To increase the overall rigidity, arrangements that limit the slit widenings are clearly preferable. The size of the slit widening also depends on the order in which we insert the slices. In the example shown in Figure 6, we could have avoided any widening by simply placing the slices in a different order: for example, by inserting the four blue slices one at a time on the green slice. An even more complex example is shown in Figure 7 where four slices are interlocked together. Note that, given the ordering shown in the figure, just a single slit widening is enough to assemble the structure. To quantify how well a slice can be inserted over a set of existing slices we introduce the concept of *divergence*. Given a slice  $s$  that is inserted over a set of slices  $s_1, \dots, s_n$ , let  $\ell_i = s \cap s_i$  be the intersection segment formed between the slice  $s$  and the  $i$ -th slice; we define the divergence  $\Lambda$  of the slice  $s$  with respect to  $s_1, \dots, s_n$  as

$$\Lambda(s) = \min_i \left( \max_{j \neq i} \text{ANGLE}(\ell_i, \ell_j) \right). \quad (2)$$

In practice  $\Lambda(s)$  denotes the maximum slit widening that we are forced to make even when the best slice for the perfect slit is chosen. For the example in Figure 6, the divergence of the green slice is the angle indicated in the second row of the right part of the figure.

### 4.3 Exploiting Oblique Slice-to-Slice Arrangement

By relaxing the orthogonal and insertion constraints we considerably increase the resulting expressive power. However, this additional degree of freedom needs to be carefully tuned to ensure that the final structure is physically stable. This entails optimizing the overall structure. Thus:

- the physical stability for a given slice arrangement is influenced by the shape of the slits. As the slits become larger, there is less friction between the pieces, thus reducing their physical stability. When the slit between two pieces is not enlarged, then we have a *perfect plug*.
- the shape of the slit is directly related both to the position of the slice and its insertion direction. As the slices become less and less perpendicular and, likewise, as the divergence between the insertion direction and intersection segment increases, the slit increases in size.

Our framework must be general enough to guarantee a correct slice structure for a given, arbitrary placement. This means that

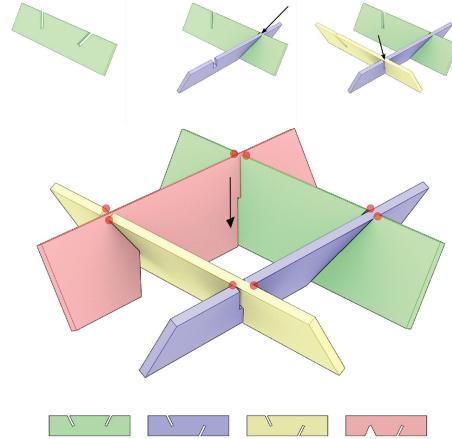


Fig. 7. Four interlocked slices that are rigidly and tightly connected, even though the slices are neither orthogonal nor inserted along a direction parallel to the intersections. Starting from the green slice, the blue and yellow slices are inserted one by one onto the previous slice along the intersection line (no slit widening needed). The last pink slice is inserted over two nonparallel slices, so widening is required. The red dots denote contact points.

the absolute position of slices must be maintained constant, though the insertion directions can be changed.

From an overall purely aesthetic perspective, the final slice structure does not depend on the sequence of gestures needed to assemble it. We only have to ensure the existence of a valid mounting sequence. Then, for a given set of slices, we optimize the insertion direction in order to increase the overall stability of the structure.

### 4.4 Ribbon-Shaped Slices

In our framework, we shaped the slices into ribbons, that is, the slices are not solid but they only define the main silhouette of the object. This kind of shape has particularly appealing visual results. Since it is possible to see through the slices, this provides a complete vision of the overall structure. Ribbon-shaped slices have additional advantages in terms of fabrication: there are considerable savings in terms of material and it is very uncommon for three slices to intersect at the same point.

Having three slices intersecting at the same point is, indeed, the standard situation of the approaches based on solid slices (such as Hildebrand et al. [2012]). The solution to these cases consists in decomposing the slices hierarchically using a BSP tree. Unfortunately, this approach means that the slices are excessively fragmented as the sampling resolution is incremented.

This situation may also arise in our approach, especially in a high curvature region, where ribbons degenerate into solid sections of the mesh. In this case, we follow a heuristic similar to Hildebrand et al. [2012]: we remove one intersecting slice by splitting the ribbon that has the smallest area.

## 5. FIELD-ALIGNED SLICE DISTRIBUTION

We define a set of *ribbons* by inflating *planar* polylines that lie on the surface of the input object.

As mentioned in Section 1 we exploit a smooth feature-aligned cross-field defined over the original surface. Given a manifold, single-connected component mesh and a cross-field, we automatically provide a set of polylines, on the original surface, which conform to the following characteristics.

*Cross-Field Alignment.* The polylines should be as aligned as possible to the input cross-field. In general, since gradient lines of a cross-field are not planar, it is impossible to provide a perfect alignment (unless we rely on tiny polylines). We must then make a trade-off between length and alignment.

*Uniform Distribution.* Polylines must sample the original surface as uniformly as possible. Since polylines intersect each other, then the intersection points must also be distributed uniformly on the original surface. This makes the overall shape seem more “regular”.

*Stability.* Once assembled, the fabricated structure must be rigid. As explained in Section 4.1, stability can be ensured locally by the orthogonality of the slices or, globally, by mutual interlocking.

### 5.1 Alignment to Cross-Field

We designed a simple procedure to trace field-aligned planar polylines. For each face and for each direction, we iteratively trace a polyline, called a *separatrix*, which follows the orientation of the field. Since the cross-field is invariant to  $90^\circ$  rotations, at each tracing step the separatrix follows one of four possible directions which has the smallest angle with the previous direction. At each tracing step, we also fit a plane to the current separatrix (the plane is constrained to lie on the initial face). We perform tracing steps iteratively while the maximum distance between the separatrix and its fitting plane stays below a certain threshold. Additionally, we may also stop the iterative tracing if the separatrix self-intersects.

The final set of *planar* polylines, which we call *traces*, is defined as the intersection between the mesh and the fitting planes. The extremes of each trace are chosen according to the extremes of the generating separatrix.

### 5.2 Distribution Constraints

We formalized a set of constraints between slices to distribute them uniformly on the surface of the object. Given a disk radius  $r$ , we sample a set of traces  $\Sigma = \{t_0, t_1, \dots, t_n\}$  generating a set  $C$  of intersections  $c_j$  such that:

- for each  $c_i, c_j \in C: D(c_i, c_j) > r$ ;
- for each  $x_i \in t_i, x_j \in t_j: D(x_i, x_j) < r \rightarrow \exists c_k \in t_i, t_j : D(x_i, c_k) < r \vee D(x_j, c_k) < r$ ,

where  $D()$  is the geodesic distance on the original surface. In practice, we search for traces whose intersections are well spaced and so that the geodesic distance between traces is larger than  $r$  (except in a neighborhood of the intersections). An example of the uniform distribution of polylines on the surface is shown in Figure 8.

Figure 9 shows a mesh sampled at different radius resolutions. Obviously the higher resolution (small values of  $r$ ) increases the details of the final model.

### 5.3 Stability Constraints

In order to keep the final structure stable, the slice arrangement must be a single-connected component.

Moreover, the slices should be almost orthogonal to each other. Indeed, orthogonality provides a good grip for the interlocking mechanism, by minimizing the slit widening.

We consider a slice *stable* if:

- it is the first slice placed on the structure;
- or it has a perfect fit with at least one other stable slice. We consider two slices to be in a perfect fit if the intersection between their planes is in between  $[\pi/2 - \delta, \pi/2 + \delta]$ ;

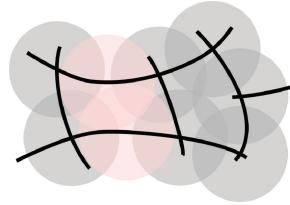


Fig. 8. The constraint used to guarantee an even distribution of the traces. Gray disks represent intersection distances, while the red disks show the distances between points that are far from the intersections.

—or the slice is interlocked in a rigid substructure (see Section 4.1, following the intuition of the triangular configuration in Figure 5).

### 5.4 The Sampling Strategy

We designed a simple algorithm to produce a slice arrangement that conforms to the constraints we mentioned before.

We build a candidate set by collecting two traces for each face (corresponding to each orthogonal direction of the cross-field). We then assign a priority value to each candidate trace. The priority of a candidate trace is the maximum length without violating the distribution constraints.

Initially we place the longest trace, and since it is the first one it is consequently stable. Then, we iteratively search for the longer trace which, when placed, would become stable.

By following this simple greedy strategy, we add candidates one by one, until no further trace can be inserted.

### 5.5 Global Regularization

Finally, we improve the distribution of the traces with a global regularization step in order to balance the space between slice intersections.

Given a trace with its intersection points, we evaluate the *optimal position* of each intersection point. Given an intersection point  $p_{int}$  its optimal position is the one that minimizes the squared sum of distances with the surrounding intersections (or endpoints). After we have calculated the optimal points, each trace is slightly moved to approach the optimal points. This operation is executed only if distribution and stability constraints are not violated.

We repeatedly execute optimization operations until the trace displacements become lower than a certain threshold.

A sequence showing the placement and optimization of slices is shown in Figure 10.

## 6. FROM RIBBONS TO ASSEMBLABLE SLICES

The planar polylines defined over the surface in the previous sections can be easily transformed into ribbons by simple extrusion.

However, if we consider a set of generic intersecting slices, there are several situations where physical assembly is impossible. For example, it is impossible to interlock two closed rings without opening at least one of them. In relation to this specific problem, Figure 11 shows a typical situation: three orthogonal ribbons, each one intersecting the other two in two different points. In this case the slices must be decomposed into at least four pieces leaving only one annular ribbon. We refer to the situation where two ribbons intersect in two different points as *multiple intersections*.

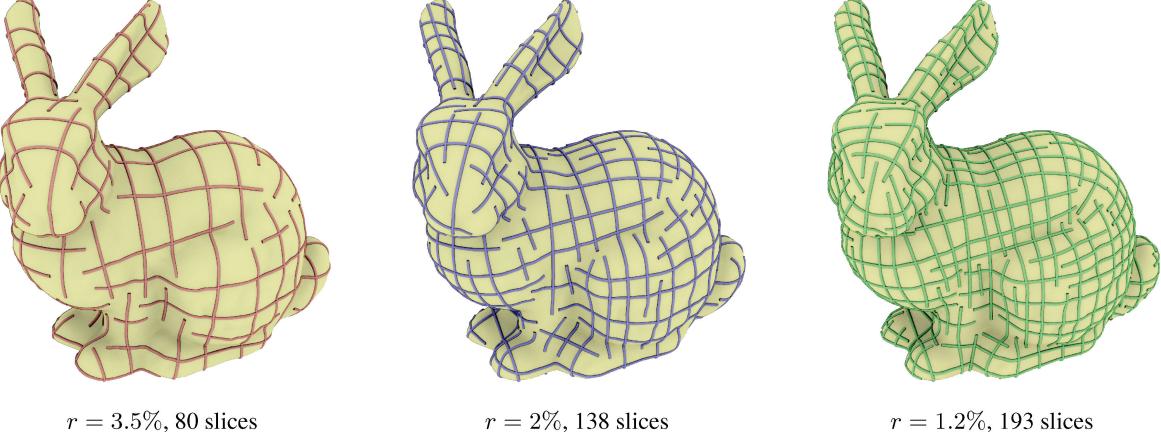


Fig. 9. The Bunny model sampled at different radius resolutions. Sampling radius  $r$  is given as a percentage of the diagonal of the model's bounding box.

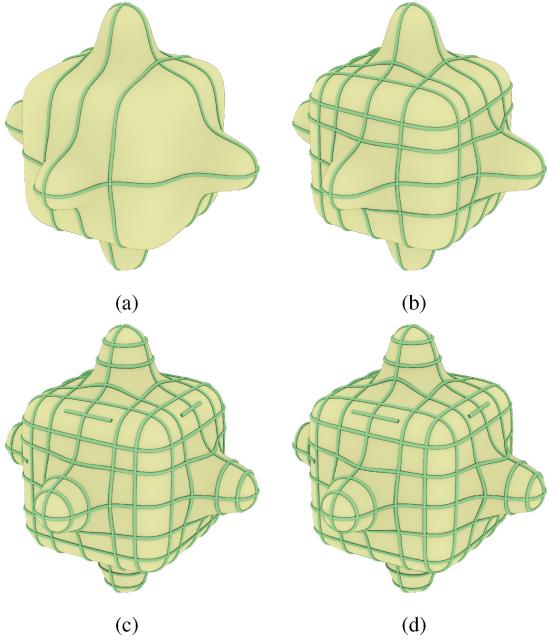


Fig. 10. A sequence of the slice sampling procedure: (a); (b) show two intermediate steps of the slice sampling procedure, composed of 6 and 12 slices respectively; (c) the final slice structure composed of 33 slices and its global regularization (d).

Let us assume that we have a set  $S = s_0, \dots, s_n$  of planar ribbons that approximates a given 3D surface  $M$ . We aim to transform  $S$  into a set  $S' = s_0, \dots, s_m$  of ribbons such that:

- (1) for each pair of ribbons  $s_1, s_2$ , the intersection  $s_1 \cap s_2$  is a proper segment  $\ell$  with exactly one of the two endpoints lying over the surface  $M$ ;
- (2) we have a proper assembly sequence, such that the resulting *divergence* is lower than a given threshold.

Under the aforesaid constraints, we are able to create the slit mechanisms described in Section 4 and, in order to fulfill them, we use the following two-step procedure which:

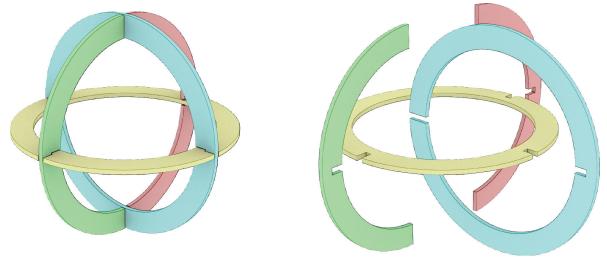


Fig. 11. Three interlocked looping ribbons must be split into four pieces so that they can be untangled.

- removes multiple intersections that limit the assembly procedure;
- minimizes the *divergence* by shuffling the slice order or if necessary by splitting some of the ribbons.

In the following sections we first introduce all the basic concepts behind the process, and then provide a more detailed description of each step.

## 6.1 Slice Graph

We model the relations between slices in the arrangement structure using a directed graph. Each node  $s_i$  of this graph represents a slice. Each arc corresponds to a physical intersection between two slices (and has to be transformed into a slit mechanism). The direction of each arc represents the priority in the partial ordering of the assembly sequence, for example, the arc  $s_i \rightarrow s_j$  means that the piece  $s_i$  must be plugged into  $s_j$ , which should already have been assembled.

Three simple examples of slice graphs with the corresponding slice arrangements are shown in Figure 12.

A valid slice graph must be acyclic. A cycle in the slice graph involves plugging one slice onto another slice that still needs to be inserted (in some geometric cases this may still be feasible by assembling all the pieces simultaneously), but this is obviously not desirable.

The orientation of the arcs in the *slice graph* can significantly affect the shape of the slit widenings, as described in Section 4 and shown in the last two rows of Figure 12 where the different arc orientations generate different slit widenings; the configuration in the middle row needs two slit widenings, while the bottom row needs only one.

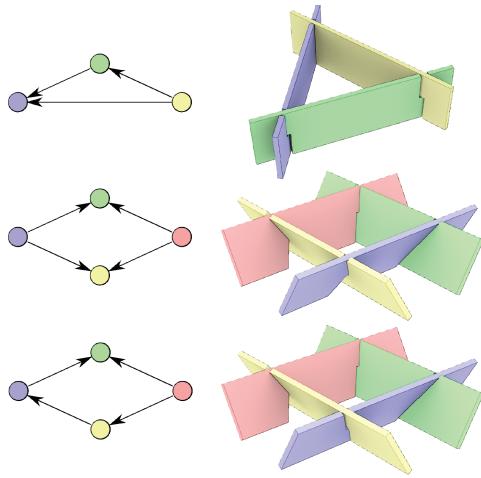


Fig. 12. The two slice graphs corresponding to the slice arrangements shown in Figures 5 and 7. The last two rows show two different arc orientations for the same slice arrangement: the slit widenings are affected by the orientation.

**6.1.1 Finding a Good Sink Set.** Initially we must select a *sink set*, that is, the initial set of disconnected, independent slices into which the remaining slices are inserted one by one. Intuitively, the sink set of a slice graph represents the *ribs* of the whole structure which we try to preserve in the various processing steps. More formally, we search for the sink set that is composed of a maximal independent set of nodes and exhibits the maximum number of arcs/relations. Unfortunately finding this optimal sink set is closely related to the problem of finding the maximum independent set in a graph: an NP-hard problem. For practical purposes, we verified that it is sufficient to randomize the procedure in order to build a maximal independent set (we randomly add nodes until the set is maximal), repeat it for a limited time, and then pick the best candidate. We found that for a typical set of slices (100 pieces), 10k to 100k attempts (a few hundred msec of computing time) are sufficient to get a stable sink set.

**6.1.2 Optimizing the Graph.** Once the sink set has been defined, we need to sort all the remaining nodes. In order to provide a good initial order, we sort all the nonsink nodes according to their maximal divergence between each pair of intersection segments. The idea is to minimize the variance of the insertion directions and their divergence once the arcs have been oriented.

Starting from this initial ordering, we swap the direction of each arc if this reduces the divergence between the insertion direction and intersection segment. We follow a greedy approach by swapping the arc that produces the greatest divergence improvement. Simultaneously, we reject any swap operation that would introduce cycles into the graph. The result of the optimization process is shown in Table I, which highlights how the graph optimization process improves the quality of the interlocking between slices. The table reports the number of slices that are perfect fits (i.e., slices with a divergence equal to zero) and the number of slices with a significant divergence (i.e., larger than 45 degrees).

## 6.2 Intersection Graph

Given a set of ribbons during the process of making it physically achievable, we need to control the degree of solidity of the assembled structure.

Table I. Slice Graph Optimization Results

The slice graph optimization allows us to increase the number of slices that make perfect fits (all the insertion directions are parallel) and to reduce the slices whose divergence is higher than a given threshold.

Model	Slice	Perf. Fit	Perf. Fit	>45	>45
		after	before	after	before
Man	112	71	54	6	17
Hand	123	82	68	0	26
Bimba	196	134	110	4	22
Ico	90	70	58	0	0

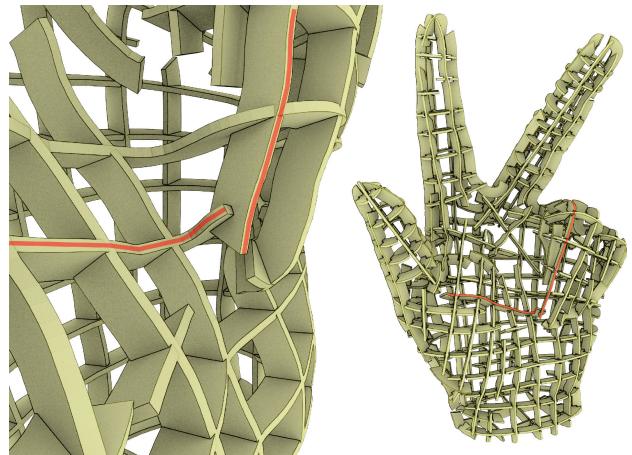


Fig. 13. A close-up of an improper intersection in the Hand model. The two ribbons marked in red have an intersection that does not touch the original surface.

For this purpose let us consider the *intersection graph*. Each node represents a ribbon intersection and an arc represents a slice that embeds two adjacent intersections.

We exploit the concept of *isoperimetric number* [Bobkov et al. 2000] (or Cheeger constant)  $h(G)$  of a graph  $G = \{V, E\}$ , a common measure of the presence of bottlenecks in a graph. The isoperimetric number  $h(G)$  is defined as

$$h(G) = \min_{0 < |U| \leq \frac{n}{2}} \frac{|\partial(U)|}{|U|}, \quad (3)$$

where the minimum is over all nonempty sets  $U \subset V$  of at most  $n/2$  vertices and  $\partial(U)$  is the *edge boundary* of  $S$ , that is, the set of edges with exactly one endpoint in  $U$ . In practice  $h(G)$  becomes small when a significant portion of the graph is connected to the rest of the graph by just a few arcs.

## 6.3 Splitting a Ribbon

Given two slices  $s_1, s_2$  with intersection segments  $\ell_1, \dots, \ell_k$ , we can improve the set of ribbons by using a *split* operation  $Split(s_1, \ell_j)$  which modifies  $s_1$  so that it no longer intersects  $s_2$  along  $\ell_j$ . The splitting operation  $Split(s_1, \ell_j)$  is performed by carving out from  $s_1$  all the points at a distance lower than  $\lambda$  from  $\ell_j$  (e.g., taking into account the relative orientation between  $s_1$  and  $s_2$ , as specified by Eq. (1)). This operation may split a slice into two separate components or, if the ribbon is a loop, it may open it.

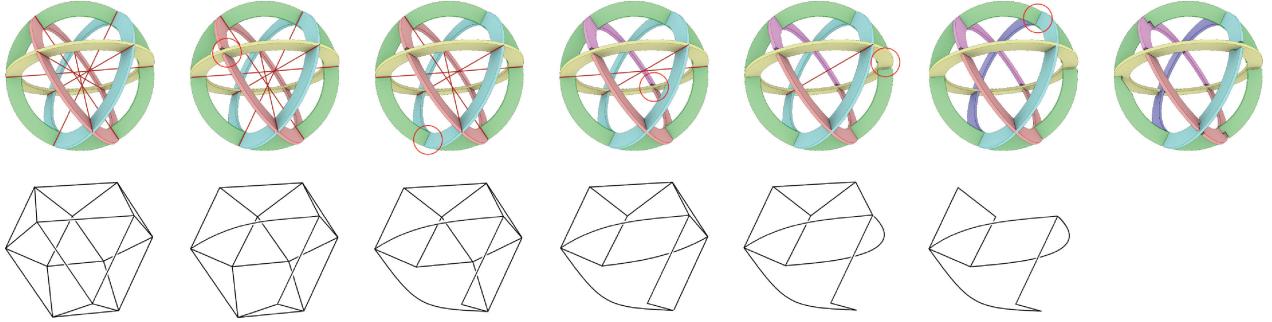


Fig. 14. An arrangement containing multiple double intersections (indicated by red lines) is corrected by means of repeated split operations (indicated with red circles). In the bottom row we show the intersection graph at each step of the process. The top-right image shows the arrangement when all the remaining six intersections are transformed into slit mechanisms.

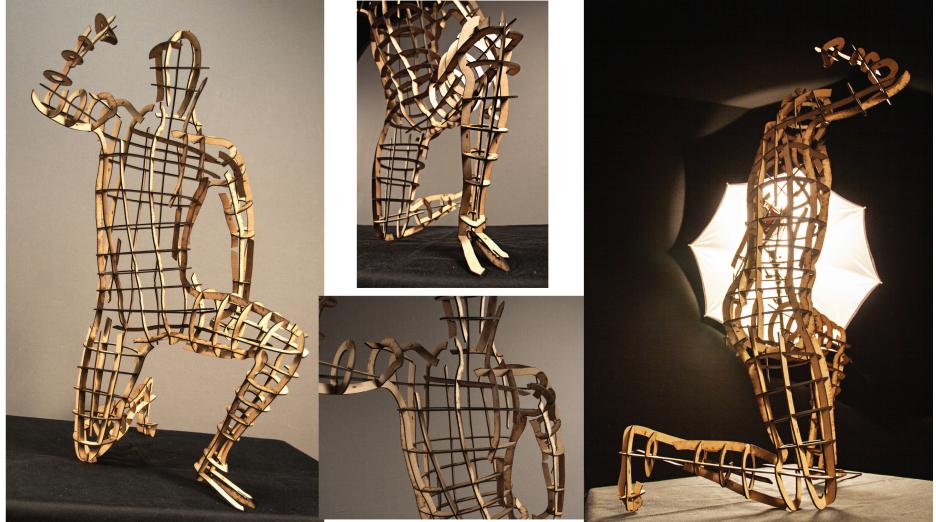


Fig. 15. The Kneeling Human model. The model is composed of 140 slices.

#### 6.4 Removing Improper Intersections

At the very beginning of the process we clean out all the improper intersections from  $S$ , for example, all the intersection segments  $\ell$  between two slices  $s_1, s_2$  that do not intersect the surface of  $M$ . These intersections do not correspond to any intersections of the generating polyline and are caused only by the intersections of the inner extrusion of the polylines. We simply remove all of them by applying two split operations for both the involved slices  $\text{Split}(s_1, \ell), \text{Split}(s_2, \ell)$ . In all the encountered examples there are only a few of these improper intersections and, once removed, we ignore their contribution for the rest of the process. In Figure 3(d) the two blue circles highlight the ribbons that were processed for removal of improper intersections. Figure 13 shows a close-up of one of these improper intersections: the two ribbons marked in red have an intersection that does not touch the original surface and therefore does not correspond to an intersection between the originating traces.

#### 6.5 Removing Double Intersections

There are two main reasons for splitting a ribbon:

- to remove double intersections;
- to lower slice divergence.

First, we remove all the double intersections, that is, pairs of slices  $s_i, s_j$  whose intersection is not a single segment  $\ell$ , but it is composed of two (or more) segments. A typical situation is depicted in Figure 11.

To clean out a double intersection, we have to carve out a portion of the slice from one of the two slices around the intersection. There is generally a choice of four different carvings (one for each slice/intersection pair). We opt for the split operation that maximizes the resulting isoperimetric number. If there are many slice splittings that lead to the same isoperimetric number, we split the nonsink slice that has the largest number of intersections with other slices.

We keep the slices in the sink intact because they were chosen specifically to increase the rigidity of the structure. Similarly, of the nonsink slices, we pick the one that will remain connected as much as possible with other slices.

Figure 14 shows an example of this process for a small arrangement made up of nonorthogonal looping ribbons on a sphere. The top row of the figure shows how the arrangement evolves during the process. The red circle highlights the result of the last split operation. The red lines highlight the double intersections that are still present in the arrangement. The last image in the top row shows the slice arrangement after transforming the remaining six intersections into slit mechanisms (machining tolerances are exaggerated

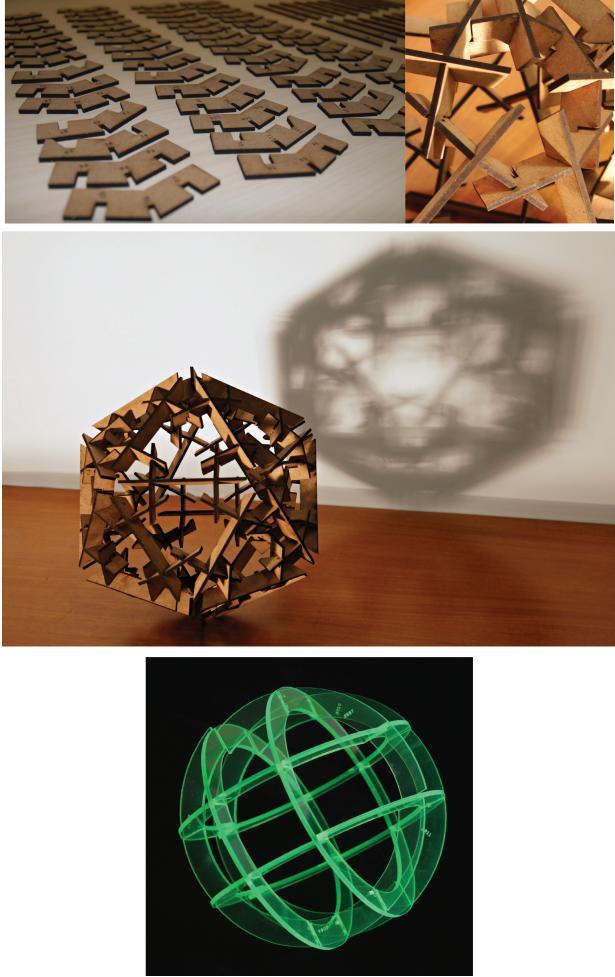


Fig. 16. A simpler slice arrangement (rather than following a cross-field) has been tested to assemble an icosahedron and a sphere (which has been built using plexiglass).

for sake of image readability). At the beginning the first sink set has just one random ribbon (in this case the yellow one). Each ribbon intersects every other ribbon in two points, so there are six double intersections. The intersection graph corresponding to each step of the process is shown in the bottom row of the figure. At the beginning, the intersection graph is equivalent to the edges of a cuboctahedron and its isoperimetric number is  $8/6$ , that is, the most fragile set of intersections has six intersections from which there are eight connections to other intersections.

We start with a sequence of five split operations and we remove the double intersections. Then the only slice that remains untouched is the original sink, two of the other ribbons have been split twice thus generating four ribbons and the last one has been split only once, thus remaining a connected component. At this point in the process there are no more double intersections and the whole structure is still rigid (see Section 4.1: each slice is involved in a four-cycle of nonparallel intersections).

## 6.6 Lowering Divergence by Splitting a Slice

Once all the double intersections have been removed and the slice graph has been optimized, we can still improve the overall



Fig. 17. Our algorithm applied to the Hand model. The arrangement is composed of 122 pieces.

arrangement by splitting those slices with a high divergence which could cause huge slit widenings. In general, when we have a slice with high divergence we can split it along one of its intersection segments. Of all the possible splitting operations that significantly minimize the divergence, we pick the operation that maximizes the resulting isoperimetric number.

Looking again at the final arrangement in Figure 14 there is a slice with a high divergence which causes slit widenings. We could remove this widening by splitting the slice, but this would lead to significant loss of rigidity. In fact, with another split, we would fail to satisfy the rigidity conditions described in Section 4.1.

In Figure 3(d) the three red circles highlight some of the split operations that were performed in order to remove double intersections (the two top red circles) and to lower the divergence (bottom red circle).

## 7. ASSEMBLING PROCEDURE

To facilitate the assembly procedure we provide basic references: all the slices and slits are labeled so that matching between pieces is unambiguous. We derive an appropriate assembling sequence as follows.

The *slice graph* optimization steps described in Section 6.1 generate a partial ordering which is tailored to minimize the divergence of the slices. Starting from this relation we want to generate a total ordering that is easy to assemble in the real world. We thus use a greedy procedure which, starting from the fully assembled slice arrangement, removes at each step the slice  $s_i$  that satisfies the following conditions.

- (1) the isoperimetric number of the *intersection graph* of  $S \setminus s_i$  is maximum (i.e., we remove the slice that leaves the structure as robust as possible);
- (2) of all the slices with the minimal  $h()$ ,  $s_i$  has the smallest number  $o(s_i)$  of outgoing arcs in the *slice graph*;
- (3) of all the slices with the minimal  $h()$  and  $o()$ ,  $s_i$  is the closest (in terms of Euclidean distance) slice to  $s_{i-1}$ .

In practice, given the fact that we consider  $h(S)$  as a measure of the robustness of the structure, we try to find an assembly order that



Fig. 18. Our algorithm applied to the Bimba model. The arrangement is composed of 178 pieces.

keeps the structure reasonably solid at each step, and in ambiguous cases, we proceed by adding the slice that has the most intersections with the already assembled structure and if possible close to the previous slices. This ordering is used to label both slices and slits.

## 8. RESULTS

We tested our method with several models from the Stanford 3D Scanning Repository (Bunny) and the AIM@SHAPE Repository (Hand, Bimba, and Kneeling Human). All the results presented in this article have been generated automatically.

If a cross-field is not available we may simply arrange slices procedurally. As an example, two configurations approximating an icosahedron and a sphere are illustrated in Figure 16.

We successfully applied the entire pipeline described in Section 3 to approximate input geometries with an associated feature-aligned cross-field as input. These structures are shown in Figures 15, 17, and 18. It took from about one to three hours to manually assemble each final model, with most of the time spent searching for the next slice. Once assembled, the resulting models were physically stable. Exploiting an input cross-field has several advantages over axis-aligned approaches, such as Hildebrand et al. [2012] (this comparison is shown in Figure 2). In addition, the cross-field can be further optimized in a preprocessing step to increase the quality of the results (see Figure 2).

Although the entire process is completely automatic, users can perform some simple editing operations to obtain a more visually pleasing result at the end of the process. Users can suggest which slice should be inserted in the sink set and force the split of a particular slice. We used the first option in the Bunny, preferring a vertical orientation of the sink slices, which is much easier to assemble.

## 9. CONCLUSIONS AND FUTURE WORK

We have proposed a novel method for the automatic fabrication of an illustrative representation of a given geometry made up of interlocked planar slices. We have shown the effectiveness of our method both in terms of illustrative quality and physical stability. To the best of our knowledge, no existing fabrication paradigms are able to represent such complex objects.

Our method is particularly efficient in terms of production costs. In fact, the production costs scale with the surface of the object since

slices are sampled almost uniformly over the surface. In addition, due to the slice decomposition, mesh joinery is also suitable for the production of medium-scale objects.

A useful extension of our framework would be to automatically generate effective instructions to simplify the manual assembly procedure, for example, a packing strategy that could preserve the partial ordering of the model to facilitate the search for the next piece.

### 9.1 Limitations

Although the range of shapes that we can efficiently approximate is wide, our method suffers from minor limitations. We did not account for the presence of other slices that could obstruct a straight insertion. However, in our experience, due to the ribbon shape of the slices, this never constitutes a serious limitation.

Moreover, we did not consider the physical issues regarding gravity and the position of the barycenter and the resulting stress acting on each individual slice. Again, in our experience, given the rigidity of the material, we had no stability problems for any of the assembled structures shown in the article.

### ACKNOWLEDGMENTS

We also thanks Giuliano Kraft and Tv@Area of the CNR Research Area of Pisa for the support in the production of the paper videos.

### REFERENCES

- Autodesk. 2013. 123D make. <http://www.123dapp.com/make/>.
- Bickel, M., Bacher, M. A., Otaduy, H. R., Lee, H., Pfister, M., Gross, and W. Matusik. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.* 29, 3, 63:1–63:10.
- P. Bo, H. Pottmann, M. Kilian, W. Wang, and J. Wallner. 2011. Circular arc structures. *ACM Trans. Graph.* 30, 101, 1–11.
- S. Bobkov, C. Houdr, and P. Tetali. 2000. Lambda and infinity, vertex isoperimetry and concentration. *Combinatorica* 20, 2, 153–172.
- D. Bommes, B. Levy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin. 2012. State of the art in quad meshing. In *EG'12 State of the Art Reports*, M.-P. Cani and F. Ganovelli, Eds., EuroGraphics Association.
- D. Bommes, H. Zimmer, and L. Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3, 77:1–77:10.

- P. Cignoni, E. Gobbetti, R. Pintus, and R. Scopigno. 2008. Color enhancement for rapid prototyping. In *Proceedings of the 9<sup>th</sup> International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST'08)*. EuroGraphics Association, 9–16.
- D. Dimitrov, K. Schreve, and N. De Beer. 2006. Advances in three dimensional printing state of the art and future perspectives. *Rapid Prototyp. J.* 12, 136–147.
- Y. Dong, J. Wang, F. Pellacini, X. Tong, and B. Guo. 2010. Fabricating spatially-varying subsurface scattering. *ACM Trans. Graph.* 29, 62:1–62:10.
- M. Eigensatz, M. Kilian, A. Schiftner, N. J. Mitra, H. Pottmann, and M. Pauly. 2010. Paneling architectural freeform surfaces. *ACM Trans. Graph.* 29, 4, 45:1–45:10.
- C.-W. Fu, C.-F. Lai, Y. He, and D. Cohen-Or. 2010. K-set tilable surfaces. *ACM Trans. Graph.* 29, 4, 44:1–44:6.
- M. Hasan, M. Fuchs, W. Matusik, H. Pfister, and S. Rusinkiewicz. 2010. Physical reproduction of materials with specified subsurface scattering. *ACM Trans. Graph.* 29, 4, 61:1–61:10.
- A. Hertzmann and D. Zorin. 2000. Illustrating smooth surfaces. In *Proceedings of the 27<sup>th</sup> Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'00)*. ACM Press/Addison-Wesley, New York, 517–526.
- K. Hildebrand, B. Bickel, and M. Alexa. 2012. Crdbrd: Shape fabrication by sliding planar slices. *Comput. Graph. Forum* 31, 583–592.
- M. Holroyd, I. Baran, J. Lawrence, and W. Matusik. 2011. Computing and fabricating multilayer models. *ACM Trans. Graph.* 30, 187:1–187:8.
- A. Johnson. 2013. Clipper library 5.1.6– An open source freeware polygon clipping library. <http://www.angusj.com/delphi/clipper.php>.
- F. Kalberer, M. Nieser, and K. Polthier. 2007. Quadcover- Surface parameterization using branched coverings. *Comput. Graph. Forum* 26, 3, 375–384.
- X.-Y. Li, T. Ju, Y. Gu, and S.-M. Hu. 2011. A geometric study of v-style pop-ups: Theories and algorithms. *ACM Trans. Graph.* 30, 4, 98:1–98:10.
- X.-Y. Li, C.-H. Shen, S.-S. Huang, T. Ju, and S.-M. Hu. 2010. Popup: Automatic paper architectures from 3D models. *ACM Trans. Graph.* 29, 4, 111:1–111:9.
- K.-Y. Lo, C.-W. Fu, and H. Li. 2009. 3D polyomino puzzle. *ACM Trans. Graph.* 28, 5, 157:1–157:8.
- F. Massarwi, C. Gotsman, and G. Elber. 2007. Papercraft models using generalized cylinders. In *Proceedings of the 15<sup>th</sup> Pacific Conference on Computer Graphics and Applications*. IEEE Computer Society, 148–157.
- W. Matusik, B. Ajdin, J. Gu, J. Lawrence, H. P. A. Lensch, F. Pellacini, and S. Rusinkiewicz. 2009. Printing spatially-varying reflectance. *ACM Trans. Graph.* 28, 5, 128:1–128:9.
- J. M. McCarthy and G. S. Soh. 2000. *Geometric Design of Linkages*, Vol. 11. Springer.
- J. McCrae, K. Singh, and N. J. Mitra. 2011. Slices: A shape-proxy based on planar sections. *ACM Trans. Graph.* 30, 6, 168:1–168:12.
- J. Mitani and H. Suzuki. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph.* 23, 3, 259–263.
- Y. Mori and T. Igarashi. 2007. Plushie: An interactive design system for plush toys. *ACM Trans. Graph.* 26, 45:1–45:8.
- D. Panizzo, Y. Lipman, E. Puppo, and D. Zorin. 2012. Fields on symmetric surfaces. *ACM Trans. Graph.* 31, 4, 111:1–111:12.
- N. Pietroni, M. Tarini, O. Sorkine, and D. Zorin. 2011. Global parameterization of range image sets. *ACM Trans. Graph.* 30, 6, 149:1–149:10.
- H. Pottmann, Q. Huang, B. Deng, A. Schiftner, M. Kilian, L. Guibas, and J. Wallner. 2010. Geodesic patterns. *ACM Trans. Graph.* 29, 4, 43:1–43:10.
- N. Ray, W. C. Li, B. Levy, A. Sheffer, and P. Alliez. 2006. Periodic global parameterization. *ACM Trans. Graph.* 25, 1460–1485.
- N. Ray, B. Vallet, L. Alonso, and B. Levy. 2009. Geometry aware direction field processing. *ACM Trans. Graph.* 29, 1:1–1:11.
- Y. Schwartzburg and M. Pauly. 2012. Design and optimization of orthogonally intersecting planar surfaces. In *Computational Design Modeling*, C. Gengnagel, A. Kilian, N. Palz, and F. Scheurer, Eds., Springer, Berlin, 191–199.
- Y. Schwartzburg and M. Pauly. 2013. Fabrication-aware design with intersecting planar pieces. *Comput. Graph. Forum* 32, 2pt3, 317–326.
- C. H. Sequin. 2012. Prototyping dissection puzzles with layered manufacturing. In *Proceedings of the Fabrication and Sculpture Track, Shape Modeling International Conference*.
- I. Shatz, A. Tal, and G. Leifman. 2006. Paper craft models from meshes. *Vis. Comput.* 22, 825–834.
- M. Singh and S. Schaefer. 2010. Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph.* 29, 4, 46:1–46:7.
- T. Weyrich, P. Peers, W. Matusik, and S. Rusinkiewicz. 2009. Fabricating microgeometry for custom surface reflectance. *ACM Trans. Graph.* 28, 3, 32:1–32:6.
- S. Xin, C.-F. Lai, C.-W. Fu, T.-T. Wong, Y. He, and D. Cohen-Or. 2011. Making burr puzzles from 3d models. *ACM Trans. Graph.* 30, 4, 97:1–97:8.

Received May 2013; revised October 2013; accepted October 2013