Field Aligned Mesh Joinery

PAOLO CIGNONI and NICO PIETRONI and LUIGI MALOMO and ROBERTO SCOPIGNO
Visual Computing Lab, National Research Council of Italy

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 to represent global features that characterize the appearance of the shape. Slice placement conforms to specific manufacturing constraints.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Physically based modeling*

General Terms: Experimentation, Human Factors

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

1. INTRODUCTION

In this paper we introduce mesh joinery, a novel and practical approach to fabricate artistic illustrative shape approximations made 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 [Hildebrand et al. 2012; McCrae et al. 2011; Schwartzburg and Pauly 2011]) we oriented the slides according to a given cross field defined on the surface. As most of the recent quadrangulation papers have shown [Ray et al. 2006; Bommes et al. 2009; Kälberer et al. 2007; Pietroni et al. 2011; Bommes et al. 2012], 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 characterize 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 cheap 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, e.g. 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 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; Li et al. 2010; Mori and Igarashi 2007]. 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 concept: approximating an object does not necessarily mean that there will be a visual deficit.

A recent approach proposed approximating the surface using a 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 shallow approximate waffles 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, 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

Paolo Cignoni acknowledges ???. Nico Pietroni acknowledges...
Authors' addresses: paolo.cignoni@isti.cnr.it; nico.pietroni@isti.cnr.it; luigi.malomo@isti.cnr.it;roberto.scopigno@isti.cnr.it.

DOI 10.1145/1559755.1559763
http://doi.acm.org/10.1145/1559755.1559763

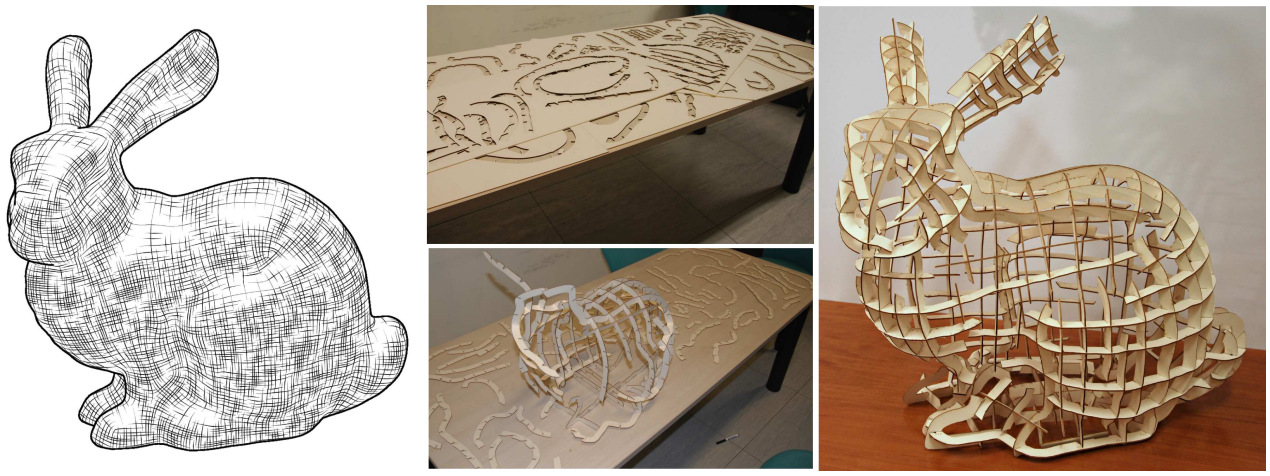
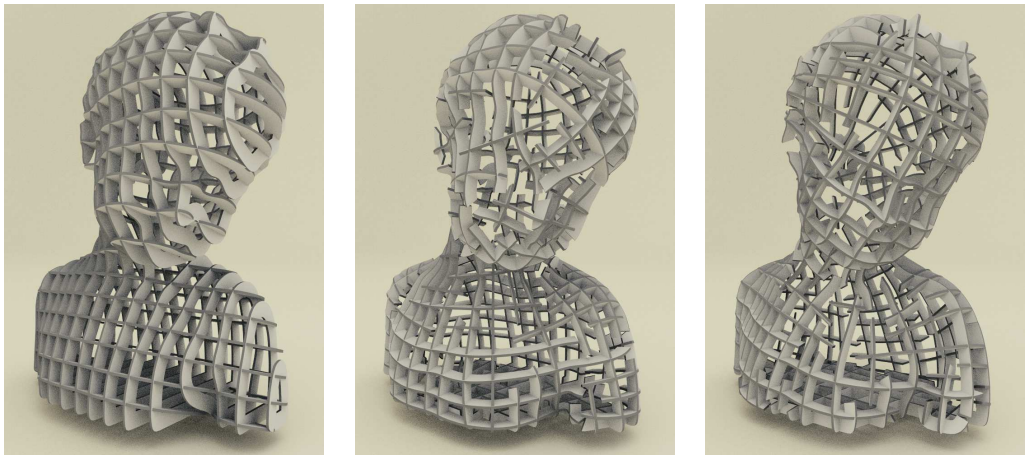


Fig. 1. Given a 3D shape with a smooth cross-field, we generate a set of planar slices that can be interlocked in a self supporting structure.



(a) (b) (c)

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 [Panozzo et al. 2012] increase the visual appeal of the final result. The total length of the polylines for each method is approximately the same.

slices reduce the physical constraints involved in the assembling procedure, allowing for more complex structures.

—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 non orthogonal slice placement [McCrae et al. 2011; Schwartzburg and Pauly 2011] in a novel , structurally sound, perspective. We have demonstrated how these additional degrees of freedom can be exploited to efficiently represent complex models.

—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 [Bommes et al. 2009; Ray et al. 2009; Hertzmann and Zorin 2000]). It provides a set of appealing, uniformly distributed polylines lying on the surface of a mesh. In addition, the method also guarantees manufacturability and a sufficiently robust slice structure. Our method may also take advantages of field symmetrization techniques, such as [Panozzo et al. 2012] (see figure 2).

—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. Secondly, it ensures 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, i.e. 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 cheap 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 micro geometry; 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 cheap, 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 [123DMake]).

For example, [Mori and Igarashi 2007] proposed a sketching interface to design plush toys. [Li et al. 2010] and [Li et al. 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.

Recently, [Hildebrand et al. 2012] proposed a method to semi-automatically fabricate objects made up of planar slices This method produces a wide range of nice results, however, it does not fit well with complex geometries (models with a high degree of asymmetry or even complex topology). [McCrae et al. 2011; Schwartzburg and Pauly 2011] extended this method to allow for non orthogonal slices. [McCrae et al. 2011] create shape abstractions based on non orthogonal planar slices, which are arranged to optimize the perception of the original object. However this method is not designed for the fabrication of tangible objects. Recently [Schwartzburg and Pauly 2013] extended [Schwartzburg and Pauly 2011] to provide a more detailed formulation on assembling non-orthogonal slices by dealing with assembly and rigidity constraints. Given a set of predefined intersecting slices, [Schwartzburg and Pauly 2013] optimizes 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 more effectively. It 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 needs 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 (we relied on ClipperLib [ClipperLib]).

(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.

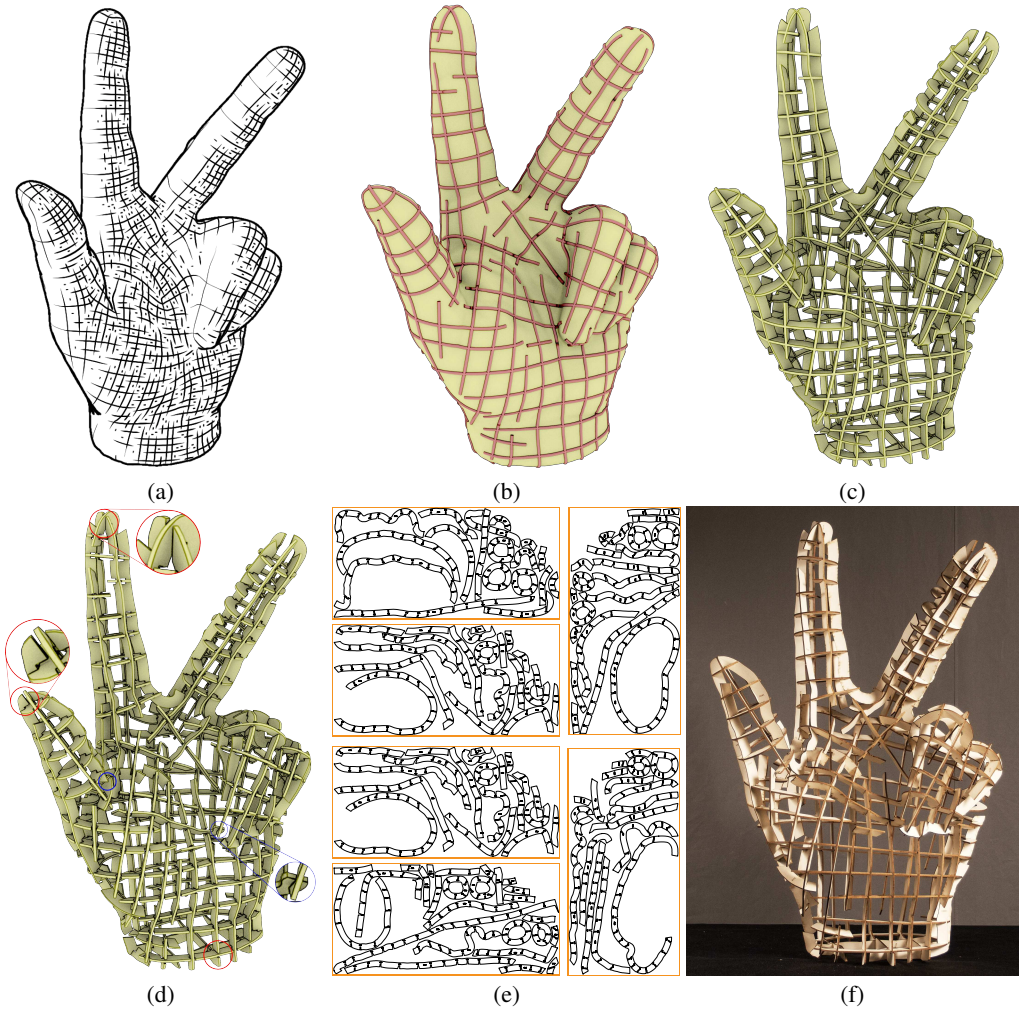


Fig. 3. A complete overview of our fabrication pipeline: (a) We get as input a triangle mesh and an associated smooth (possibly symmetric) cross field; (b) We sample a set of well-distributed field oriented planar polylines; (c) The polylines are transformed into ribbon-shaped slices; (d) The slice structure 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.

4. INTERLOCKING PLANAR SLICES

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

For the sake of simplicity, consider the simple situation of two perpendicular slices fitting together (see Figure 5). 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 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 that this method can be applied to.
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

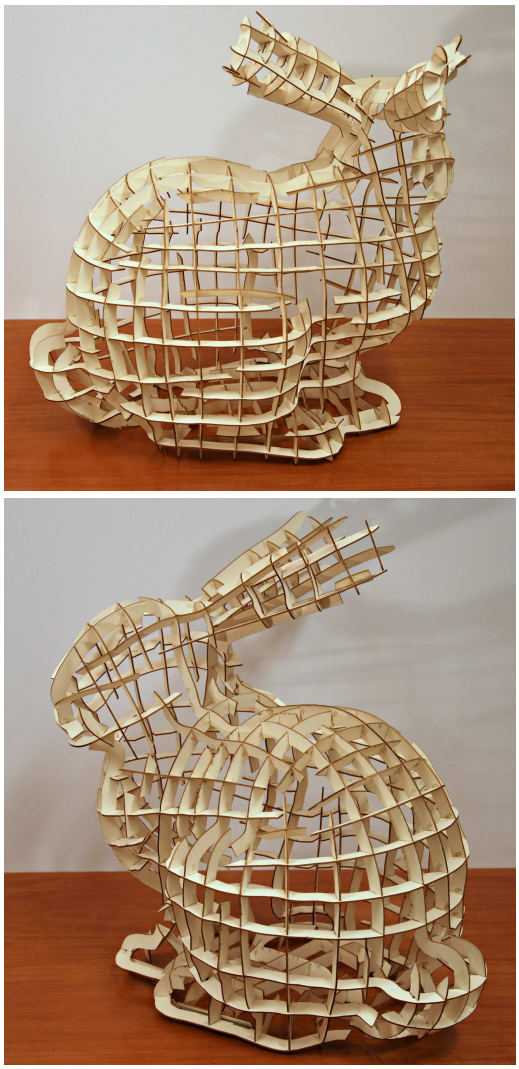


Fig. 4. Our algorithm applied on the classic Stanford bunny. Note how the slices follow the cross field, thus capturing the global structure of the object. The model is composed by 152 pieces.

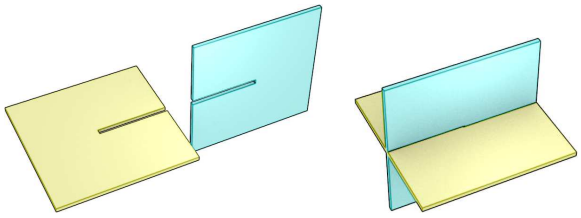


Fig. 5. The classical situation of two connected slices: for each piece we create a rectangular slit in correspondence with the intersection line.

ensures a firm interlock of the two pieces. If the two slices are not orthogonal, the slit has to be widened by the factor λ:

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

where τ is the slice thickness and α is the angle between the two slice planes.

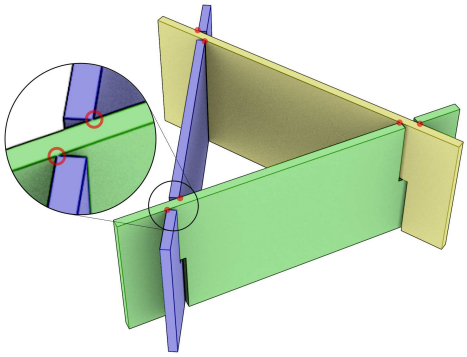


Fig. 6. Three interlocked slices are rigid and tightly connected, although the slices are not orthogonal and the wide slits are not tightly fitted onto the surface of the other slice. The red dots denote where the slices are pressed/forced against each other, such that the resulting friction ensures 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 arrangements (see Fig. 6) or, alternatively, by four slices interlocked together with non-parallel intersections (see Fig. 8). In the latter case, the rigidity derives from the fact that a non orthogonal slit is like a hinge and the four connected slices form a four-bar linkage[McCarthy 2000]. In general any spatial linkage formed by four links and four hinged joints 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° is not sufficient to deal with all the possible real scenarios. Indeed, as illustrated in Figure 7, when a slice (the green one) has to be inserted over four existing non parallel slices (the blue ones), the direction of insertion will definitely not be parallel to some of 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 intersection segments is required, so that at least one of the slits holds the other piece steady.

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 we insert the slices. In the example shown in Figure 7, 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 8 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 $s_1, ...s_n$, 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 Λ of the slice s with respect to $s_1, ...s_n$ as:

$$\Lambda(s) = \min_i(\max_{j \neq i} \textsc{Angle}(\ell_i, \ell_j)) \quad (2)$$

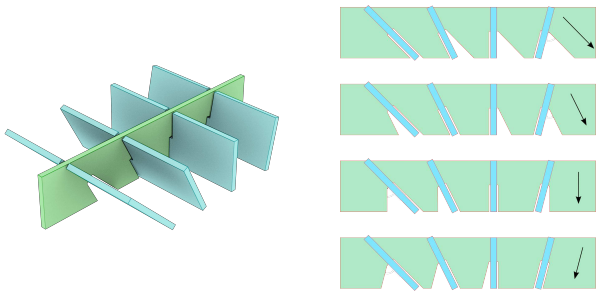


Fig. 7. The shape of the slit widening depends on the insertion direction. The divergence of the green slice is the maximum angle between the various intersection segments when the best insertion direction is chosen. On the right we choose how the slit widening varies when different insert directions are chosen.

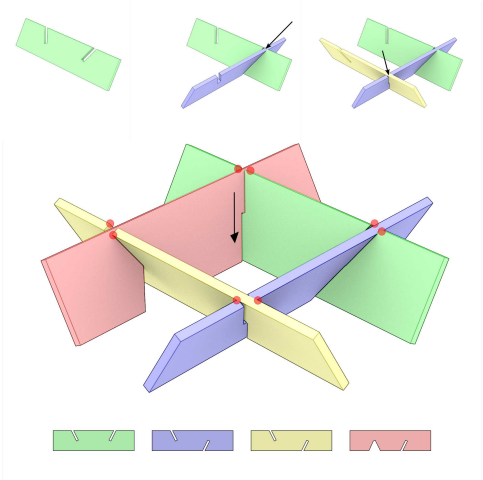


Fig. 8. Four interlocked slices that are rigidly and tightly connected, even though the slices are neither orthogonal nor inserted along a direction parallel to the intersections. Starting from the green slice, the blue and yellow slices are inserted one by one onto the previous slice along the intersection line (no slit widening needed). The last pink slice is inserted over two non parallel slices, so widening is required. The red dots denote contact points.

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 of Figure 7 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 optimizating 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 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, i.e. 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: it is a considerable save 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 lies 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 perfect alignment (unless we rely on tiny polylines). We must then make a tradeoff 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.

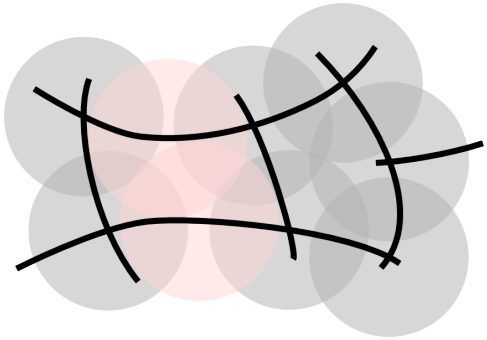


Fig. 9. The constraint used to guarantee an even distribution of the traces. Gray disks represent intersection distances, while the red disks show the distances between points that are far from the intersections.

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° 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 stay 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 $\Sigma = \{t_0, t_1, ..., t_n\}$ 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 $x_i \in t_i, x_j \in t_j$:
$D(x_i, x_j) < r \rightarrow$
$\exists\, c_k \in t_i, t_j : D(x_i, c_k) < r \bigvee D(x_j, c_k) < r$

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 9.
Figure 10 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 another stable slice. We consider two slices to be in a perfect fit if the intersection between their planes is in between $[\pi/2 - \delta, \pi/2 + \delta]$.

—or the slice is interlocked in a rigid substructure (see sec 4.1, following the intuition of the triangular configuration in Figure 13).

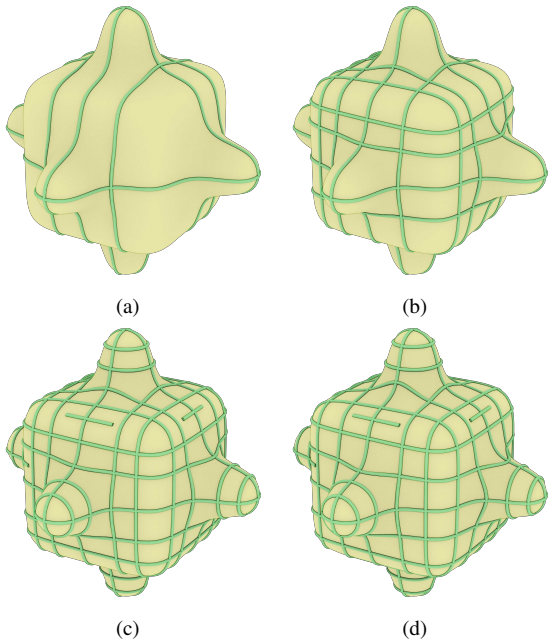


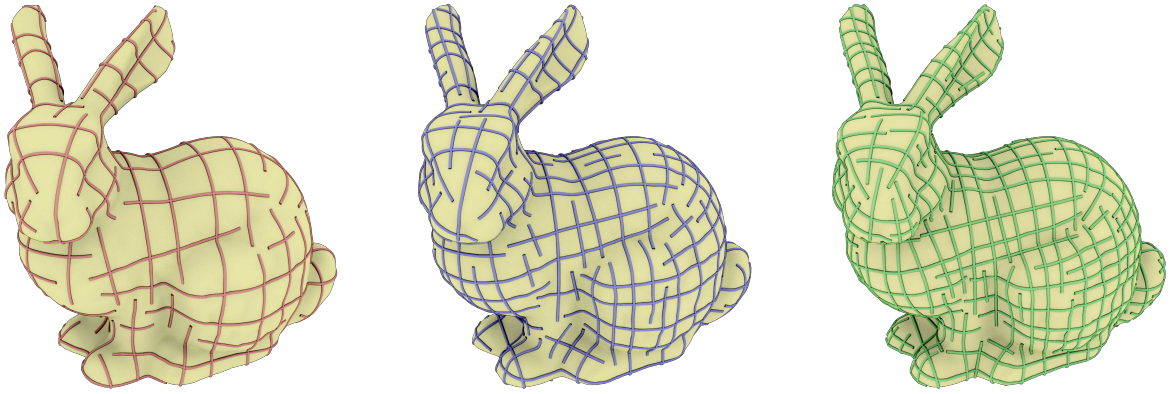
Fig. 11. A sequence of the slice sampling procedure: (a) (b) show two intermediate steps of the slice sampling procedure, composed by 6 and 12 slices respectively; (c) the final slice structure composed by 33 slices and its global regularization (d).

5.4 The sampling strategy

We designed a simple algorithm to produce a slice arrangement that conforms to the constraints we mentioned above.
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 it's the maximum length without violating the distribution constraints.
Initially we place the longest trace, since it is the first one, then it is 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}



$r = 3.5\%$, 80 slices $r = 2\%$, 138 slices $r = 1.2\%$, 193 slices

Fig. 10. The bunny model sampled at different radius resolutions. Sampling radius r is given as percentage of the diagonal of the model's bounding box.

its optimal position is the one that minimizes the squared sum of distances with the surrounding intersections (or endpoints). After we 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 11.

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, Fig. 12 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 intersects in two different points as *multiple intersections*.

Let us assume that we have a set $S = s_0, ..., s_n$ of planar ribbons that approximate a given 3D surface M.

We aim to transform S into a set $S' = s_0, ..., s_m$ 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 ℓ with exactly one of the two endpoints lying over the surface S;

(2) we have a proper assembly sequence, such that the resulting *divergence* is lower than a given threshold.

Under the above 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 that:

—remove the multiple intersections that limit the assembly procedure

—minimize the *divergence* by shuffling the slice order or eventually by splitting some of the ribbons.

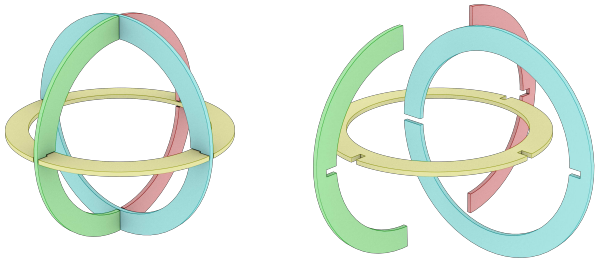


Fig. 12. Three interlocked looping ribbons must be split in four pieces so that they can be untangled.

In the following subsections, 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, e.g. 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 Fig. 13.

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 13 where the different arc orientations generate different slit widenings; the configuration in the middle row needs two slit widenings, while the bottom row need only one.

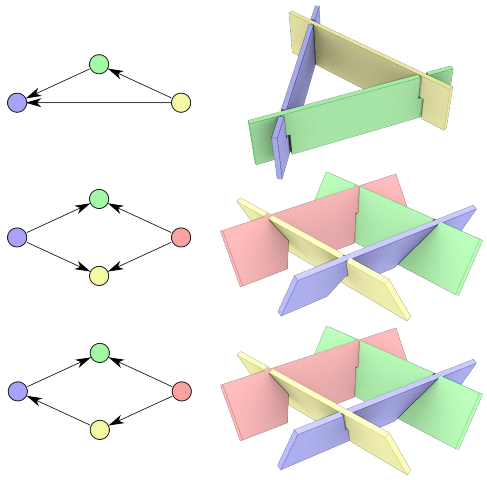


Fig. 13. The two Slice Graphs corresponding to the slice arrangements shown in Figs. 6 and 8. The last two rows show two different arc orientations for the same slice arrangement: the slit widenings is affected by the orientation.

6.1.1 *Finding a good sink set.* Initially we must select a *sink* set, i.e. 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 by 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 of nodes in a graph: an NP-complete problem. For practical purposes, we verified that it is sufficient to randomize the procedure in order to build a maximal independent set, iterate it for a limited time, and then pick the best candidate. We found that for a typical set of slices (100 pieces), 10k $\sim$ 100k attempts (a few hundred msecs 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 non-sink 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 graph. The result of the optimization process is shown in table 16 which reports how the graph optimization process allow to improve the quality of the interlocking between slices. The table reports the number of slices that are perfect fits (e.g. slices having divergence equal to zero) and the number of slices with a significant divergence (e.g. larger than 45 degree).

6.2 Intersection Graph

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

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:

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

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, i.e., 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 $\ell_1, ..., \ell_k$, 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 ℓ_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 λ from ℓ_j (e.g. taking into account the relative orientation between s_1 and s_2, as specified by equation 1). This operation may split a slice into two separate components or, if the ribbon is a loop, it may open it.

6.4 Removing Improper Intersections

At the very beginning of the process we clean out all the improper intersections from S, e.g. all the intersection segments ℓ between two slices s_1, s_2 which 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 $Split(s_1, \ell), Split(s_2, \ell)$. In all the encountered examples there are only a few of these improper intersections are once removed, we ignore their contribution. In figure 3.d the two blue circles highlight the ribbons that were processed for removal of improper intersections. Fig. 14 show 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, e.g. pairs of slices s_i, s_j whose intersection is not a single segment ℓ, but it is composed by two (or more) segments. A typical situation is depicted in Figure 12.

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 non-sink slice that has the largest number of intersections with other slices.

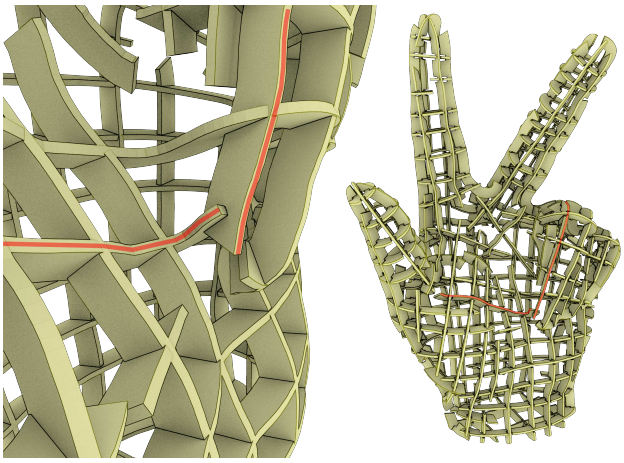


Fig. 14. A close up of an improper intersection in the Hand model. The two ribbons marked in red have an intersection that does not touch the original surface

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

Figure 15 shows an example of this process for a small arrangement made up of non orthogonal looping ribbons on a sphere. The top row of the figure shows how the arrangement evolve during the process. The red circle highlight the result of the last split operation. The red lines are icons showing the double intersections still present in the arrangement. At the beginning the first sink set has just a 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, i.e. the most fragile set of intersections has 6 intersections from which there are eight connections to other intersections.

We start and iteratively remove double intersections with a sequence of five splits operations. Then the only slice that remains untouched is the original sink, two of the other two ribbons have been split twice generating four ribbons and the last one has been split only once remaining a connected component. At this point in the process there are no more double intersections and the whole structure is still rigid (See sec. sec. 4.1: any slice is involved in a four-cycle of non parallel 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 arrangement by splitting the 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 Fig.15 there is a slice with a high divergence which causes the slit widening. We could remove this widening by splitting the slice, but this would cause a significant loss of rigidity. In fact, with another split, we would fail to satisfy the rigidity conditions described in sec. 4.1.

In figure 3.d the three red circles highlight some of the split operations that were performed for removing double intersections (the two top red circles) and for lowering the divergence (bottom red circle).

Model	Slice	Perf. Fit after	Perf Fit before	> 45 after	> 45 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. 16. The slice graph optimization allows to increase the number of slices that make perfect fits (all the insertion directions are parallel) and to reduce the slices with divergence higher than a given threshold.

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 (e.g. 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 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 slice and slits.

8. RESULTS

We tested our method with several models. All the results presented in this paper have been generated automatically.

In the case a cross field is not available we may simply arrange slices procedurally. For example two configurations approximating an icosahedron and a sphere are illustrated in Figure 18.

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 4,17,19 and 20. 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

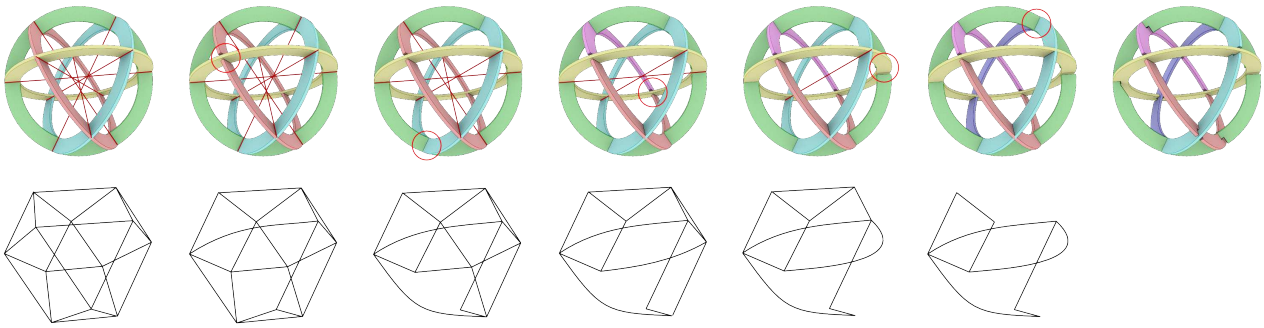


Fig. 15. An arrangement containing multiple double intersections (indicated by red lines) is corrected by mean of repeated split operations (indicated with red circles). In the bottom row we show the intersection graph at each step of the process.

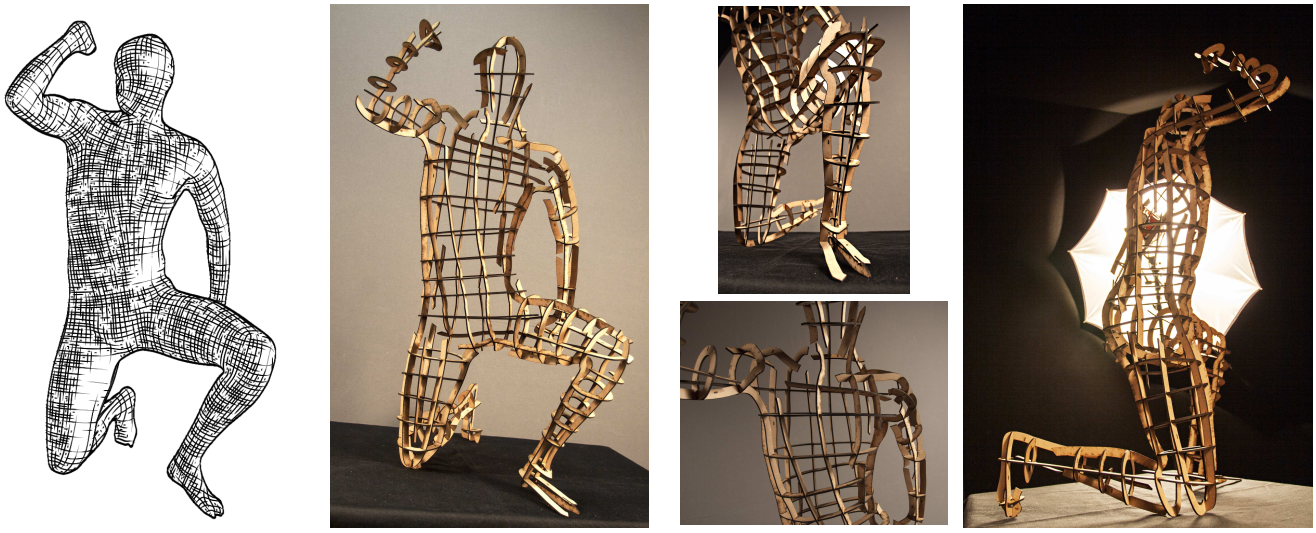


Fig. 17. The 'kneeling human' model.The model is composed by 140 slices.

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 nicer 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 particular 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

Even if the range of shape we can efficiently approximate is wide, the proposed method suffers of minor limitations.
We didn't account for the presence of other slices that should obstacle a smooth insertions movement. However, in our practical experience, due to the ribbon-shape of the slices, this never constitutes a serious limitation.
Moreover, we didn't considered the physical issues regarding gravity and so the position of the barycenter and the resulting stress acting on each individual slice. Again, in our practical experience , given the rigidity of the material, we had any stability problem for any of the assembled structures shown in the paper.

REFERENCES

123DMake. Autodesk 123d make. http://www.123dapp.com/make/.

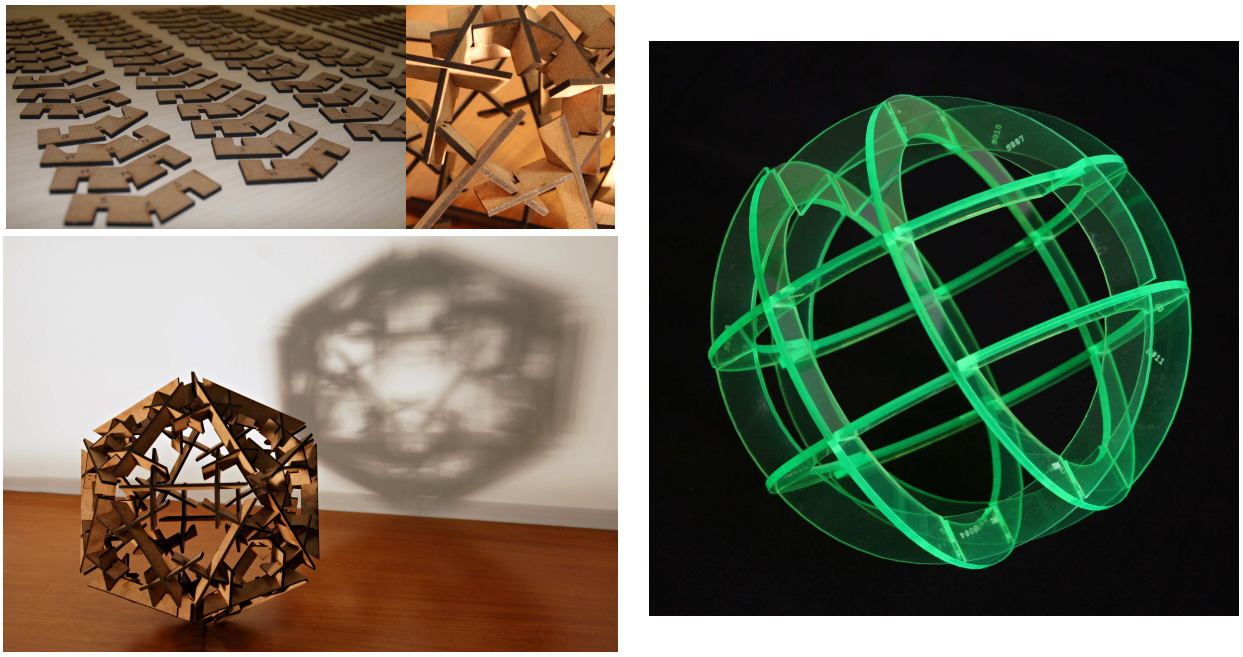


Fig. 18. A simpler slice arrangement (rather than following a cross field) has been tested to assemble an icosahedroan and a sphere (which has been build using plexyglass).

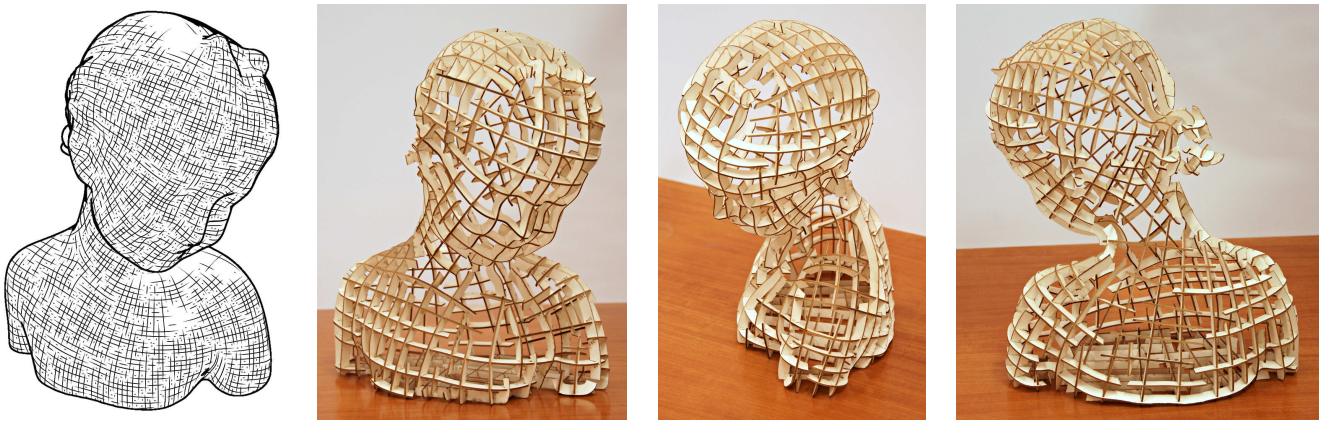


Fig. 20. Our algorithm applied the 'bimba' model. The arrangement is composed by 178 pieces.

BICKEL, B., BÄCHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Trans. on Graphics 29,* 3, 63:1–63:10.

BO, P., POTTMANN, H., KILIAN, M., WANG, W., AND WALLNER, J. 2011. Circular arc structures. *ACM Trans. on Graphics 30*, #101,1–11.

BOBKOV, S., HOUDR, C., AND TETALI, P. 2000. λ_∞,vertex isoperimetry and concentration. *Combinatorica 20,* 2, 153–172.

BOMMES, D., LÉVY, B., PIETRONI, N., PUPPO, E., A, C. S., TARINI, M., AND ZORIN, D. 2012. State of the art in quad meshing. In *Eurographics STARS*.

BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. In *ACM SIGGRAPH 2009 papers*. SIGGRAPH '09. ACM, New York, NY, USA, 77:1–77:10.

CIGNONI, P., GOBBETTI, E., PINTUS, R., AND SCOPIGNO, R. 2008. Color enhancement for rapid prototyping. In *The 9th VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage*. Eurographics, 9–16.

ClipperLib. Clipper - an open source freeware polygon clipping library. `http://www.angusj.com/delphi/clipper.php`.

DIMITROV, D., SCHREVE, K., AND DE BEER, N. 2006. Advances in three dimensional printing state of the art and future perspectives. *Rapid Prototyping Journal 12*, 136–147.

DONG, Y., WANG, J., PELLACINI, F., TONG, X., AND GUO, B. 2010. Fabricating spatially-varying subsurface scattering. *ACM Trans. Graph. 29*, 62:1–62:10.

EIGENSATZ, M., KILIAN, M., SCHIFTNER, A., MITRA, N. J., POTTMANN, H., AND PAULY, M. 2010. Paneling architectural freeform

Fig. 19. Our algorithm applied the 'hand' model. The arrangement is composed by 122 pieces.

surfaces. In *ACM SIGGRAPH 2010 papers*. ACM, New York, USA, 45:1–45:10.

FU, C.-W., LAI, C.-F., HE, Y., AND COHEN-OR, D. 2010. K-set tilable surfaces. In *ACM SIGGRAPH 2010 papers*. SIGGRAPH '10. ACM, New York, NY, USA, 44:1–44:6.

HAŠAN, M., FUCHS, M., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2010. Physical reproduction of materials with specified subsurface scattering. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29,* 3 (Aug.).

HERTZMANN, A. AND ZORIN, D. 2000. Illustrating smooth surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '00. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 517–526.

HILDEBRAND, K., BICKEL, B., AND ALEXA, M. 2012. crdbrd: Shape fabrication by sliding planar slices. *Computer Graphics Forum (Eurographics 2012) 31*.

HOLROYD, M., BARAN, I., LAWRENCE, J., AND MATUSIK, W. 2011. Computing and fabricating multilayer models. *ACM Trans. Graph. 30*, 187:1–187:8.

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

LI, X.-Y., JU, T., GU, Y., AND HU, S.-M. 2011. A geometric study of v-style pop-ups: theories and algorithms. In *ACM SIGGRAPH 2011 papers*. SIGGRAPH '11. ACM, New York, NY, USA, 98:1–98:10.

LI, X.-Y., SHEN, C.-H., HUANG, S.-S., JU, T., AND HU, S.-M. 2010. Popup: automatic paper architectures from 3d models. In *ACM SIGGRAPH 2010 papers*. SIGGRAPH '10. ACM, New York, NY, USA, 111:1–111:9.

LO, K.-Y., FU, C.-W., AND LI, H. 2009. 3d polyomino puzzle. In *ACM SIGGRAPH Asia 2009 papers*. SIGGRAPH Asia '09. ACM, New York, NY, USA, 157:1–157:8.

MASSARWI, F., GOTSMAN, C., AND ELBER, G. 2007. Papercraft models using generalized cylinders. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*. IEEE Computer Society, Washington, DC, USA, 148–157.

MATUSIK, W., AJDIN, B., GU, J., LAWRENCE, J., LENSCH, H. P., PELLACINI, F., AND RUSINKIEWICZ, S. 2009. Printing spatially-varying reflectance. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 28,* 5 (Dec.).

MCCARTHY, J. M. 2000. *Geometric design of linkages*. Vol. 11. Springer.

MCCRAE, J., SINGH, K., AND MITRA, N. J. 2011. Slices: a shape-proxy based on planar sections. In *Proceedings of the 2011 SIGGRAPH Asia Conference*. SA '11. ACM, New York, NY, USA, 168:1–168:12.

MITANI, J. AND SUZUKI, H. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. In *ACM SIGGRAPH 2004 Papers*. ACM, New York, NY, USA, 259–263.

MORI, Y. AND IGARASHI, T. 2007. Plushie: an interactive design system for plush toys. *ACM Trans. Graph. 26*.

PANOZZO, D., LIPMAN, Y., PUPPO, E., AND ZORIN, D. 2012. Fields on symmetric surfaces. *ACM Trans. Graph. 31,* 4 (July), 111:1–111:12.

PIETRONI, N., TARINI, M., SORKINE, O., AND ZORIN, D. 2011. Global parametrization of range image sets. *ACM Trans. Graph. 30,* 6 (Dec.), 149:1–149:10.

POTTMANN, H., HUANG, Q., DENG, B., SCHIFTNER, A., KILIAN, M., GUIBAS, L., AND WALLNER, J. 2010. Geodesic patterns. In *ACM SIGGRAPH 2010 papers*. SIGGRAPH '10. ACM, New York, NY, USA, 43:1–43:10.

RAY, N., LI, W. C., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph. 25*, 1460–1485.

RAY, N., VALLET, B., ALONSO, L., AND LVY, B. 2009. Geometry aware direction field processing. *ACM Trans. on Graphics 29*, 1:1–1:11.

SCHWARTZBURG, Y. AND PAULY, M. 2011. Design and optimization of orthogonally intersecting planar surfaces. In *Computational Design Modelling 2012*.

SCHWARTZBURG, Y. AND PAULY, M. 2013. Fabrication-aware design with intersecting planar pieces. *Computer Graphics Forum (Proceedings of Eurographics 2013) 32,* 2. to appear.

SÉQUIN, C. H. Prototyping dissection puzzles with layered manufacturing.

SHATZ, I., TAL, A., AND LEIFMAN, G. 2006. Paper craft models from meshes. *Vis. Comput. 22*, 825–834.

SINGH, M. AND SCHAEFER, S. 2010. Triangle surfaces with discrete equivalence classes. In *ACM SIGGRAPH 2010 papers*. SIGGRAPH '10. ACM, New York, NY, USA, 46:1–46:7.

WEYRICH, T., PEERS, P., MATUSIK, W., AND RUSINKIEWICZ, S. 2009. Fabricating microgeometry for custom surface reflectance. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28,* 3 (Aug.).

XIN, S., LAI, C.-F., FU, C.-W., WONG, T.-T., HE, Y., AND COHEN-OR, D. 2011. Making burr puzzles from 3d models. In *ACM SIGGRAPH 2011 papers*. SIGGRAPH '11. ACM, New York, NY, USA, 97:1–97:8.