Università di Pisa
Settore Scientifico Disciplinare: INF/01

Ph.D. Thesis: XXX cycle

Enhancing Digital Fabrication with Advanced Modeling Techniques

Luigi Malomo

Supervisor
Dr. Paolo Cignoni

Supervisor
Dr. Nico Pietroni

Referee
Dr. Antonio Cisternino

Referee
Prof. Antonio Frangioni

Chair
Prof. Pierpaolo Degano

March, 2017

Ai miei genitori.
Ad Angela.

Abstract

A few years ago there were only expensive machineries dedicated to rapid prototyping for professionals or industrial application, while nowadays very affordable solutions are on the market and have become useful tools for experimenting, providing access to final users.

Given the digital nature of these machine-controlled manufacturing processes, a clear need exists for computational tools that support this new way of productional thinking. For this reason the ultimate target of this research is to improve the easiness of use of such technologies, providing novel supporting tools and methods to ultimately sustain the concept of democratized design ("fabrication for the masses"). In this thesis we present a novel set of methods to enable, with the available manufacturing devices, new cost-effective and powerful ways of producing objects. The contributions of the thesis are three. The first one is a technique that allows to automatically create a tangible illustrative representation of a 3D model by interlocking together a set of planar pieces, which can be fabricated using a 2D laser cutter. The second method allows to automatically design flexible reusable molds to produce many copies of an input digital object. The designs produced by this method can be directly sent to a 3D printer and used to liquid-cast multiple replicas using a wide variety of materials. The last technique is a method to fabricate, using a single-material 3D printer, objects with custom elasticity, and an optimization strategy that, varying the elastic properties inside the volume, is able to design printable objects with a prescribed mechanical behavior.

Acknowledgments

As known, research work is something that simply cannot be performed alone. For this reason I am extremely thankful to all the people I have met during my Ph.D. course.

First and foremost I would like to thank my supervisors at the Visual Computing Lab of ISTI - CNR, Paolo Cignoni and Nico Pietroni, for guiding me during these years and teaching me how to do research. Among all the people in the group a special thanks goes to the head of the Lab Roberto Scopigno; it's thanks to his constant and hard work that his Lab has been and continue to be a wonderful and stimulant working environment. Without distinction I would like to thank all the people in the group, from senior researchers to master students.

Also, among everyone I have met and interacted with, particular gratitude goes to the persons that worked with me in my research and the ones that gave a significant contribution to my education: Denis Zorin, Julian Panetta, Qingnan Zhou, Bernd Bickel, Eder Miguel, Marco Tarini, Marco Di Benedetto, Andrea Baldacci, Giorgio Marcias, Luca Benedetti and Francesco Banterle.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This Ph.D Thesis is focused to *Digital Fabrication*, an active research field that has rapidly gained interest across multiple domains.

Fabrication *per se* is a very old discipline: way before the existence of digital technologies men have always tried and accomplished to manufacture objects and create buildings starting from ideas and designs. Over the centuries people had to overcome many challenges related to the laws of physics, materials employed, and the availability of manufacturing techniques.

In modern times, it is clear the advantage of digital technologies as powerful instruments to assist the realization of tangible objects and therefore they are constantly used to support the work in many fields. From the engineering point of view a vast knowledge already exists but has been always confined to very specific applications (e.g., buildings, mechanics, architecture). For these purposes, digital tools that assist in the design and realization of physical structures/objects are now pervasive but they ultimately require a high technical expertise and knowledge of each specific field.

The advent of rapid prototyping machines had radically changed the scenario. While a few years ago there were only expensive machineries dedicated to rapid prototyping for professionals or industrial application, nowadays very affordable solutions are on the market and for this reason we saw an incredible diffusion in the last years. In particular, additive manufacturing technology (3D printing) and in general CNC (computer-numerical control) machines have been available for many years, however only in the recent past have become cheaply available, providing access to consumer-level users.

The novelty represented by affordable and accessible digital manufacturing technologies have become a useful tool for experimenting and has created new needs for everyone that requires to use them, from hobbyists to professionals. In the light of such developments, many industrial and commercial companies flourished: from the companies that produce 3D printers (e.g., 3D Systems [3D 16]), to online 3D printing services like Shapeways [Sha16], along with custom

laser cutting services (e.g., Ponoko [Pon16]). The related industry growth happened thanks to the different application domains that can benefit by these novel manufacturing technologies along with the Fabrication research field (e.g. toys, medical prosthetics, architecture, manufacturing, art, etc.). As a consequence of recent discoveries, many application software were developed, for both serious applications (e.g., dental prosthetics CADs for 3D printing) and illustrative purposes (e.g., the Autodesk software 123D Make [Aut13], that allows to interactively design artifacts made of interlocking pieces).

Given the digital nature of novel machine-controlled manufacturing processes, a clear need exists for computational tools supporting their usage, especially for consumer-level users. For this reason the effort of the research community is targeted to improve the easiness of use of such technologies, providing supporting tools and computational methods that render fabrication *accessible* to consumer-level users, while at the same time push the boundaries of such technologies that could have a strong impact on industrial applications.

This topic has been endorsed by the computer graphics community. Simulation of physical (real) properties of the world around us has always been a main objective of the community. This is mainly true because of the mainstream applications that are commonly associated to the CG domain: visual effects and videogames. For both of them, and in particular for the former, the accurate simulation of reality has always been a must (e.g., the appearance of objects, their mechanical behavior, etc.). Commonly the knowledge in the field has been directly or indirectly related to the process of *rendering*, which objective is to display realistic images or movies on a 2D screen, encompassing all the physical phenomena involved in its realization. The competencies and knowledge required to obtain these results are incredibly valuable and can have a variety of applications in unexpected domains. In particular, the problems that have been solved in CG embraces physics, mathematics, engineering and computer science and they are not confined to synthetic 3D models and their visualization on a computer screen. All this expertise have been successfully employed for *tangible* application, therefore, to some extent, the research oriented to digital fabrication could be related to the concept of rendering, not for the screen, but for the real world.

1.1 Research directions

In Computer Graphics the efforts of the research on digital fabrication have been heterogeneous but the common objectives can be synthesized into three categories that not necessarily exclude each other.

Direct usage improvement The first objective is to simplify the usage of digitally controlled machines. In most cases, the 3D model reproduction or the creation of freshly designed objects cannot be achieved without an expert knowledge of the machine being used. As all technologies, these machines have technical limitations and are not always able to directly produce the intended shape. For example, for 3D printers there is number of factors that influence the outcome of a print and many constraints that reduce the variety of shapes that can be produced. Some limitations are, for example, the maximum printable volume, the printing speed, the minimum thickness required, the eventual requirements of supports, the physical property of the base material employed, etc. This research direction focuses on creating methods and algorithms that, automatically or interactively, are able to optimize the use of these technologies, lifting the burden of technical analysis for each specific need. In particular, there are two main approaches to achieve this goal, depending on the application. One set of techniques aims to obtain reproduction of an object without modifying its design. For example, this is useful for objects that have a specific functionality for which it is mandatory to preserve their intended shape. On the other hand, another class of techniques modifies the shape of the original design to overcome manufacturing limitations. In general, the result of such efforts can optimize the cost of production, its speed, reveal design flaws and improve the quality of the produced artifacts.

Innovative design The usage of digital fabrication technologies empowers users to realize original and complex creations. Despite these machines are considered *enabling technologies*, which offer great freedom of expression, very often the designs people want to achieve require an extensive effort and a trial and error process in order to be successful. Many times, the complexity related to some intended designs is so high that some ideas are impossible to realize without the help of computational tools. The general approach consists of modeling the set of constraints for the specific machinery, the manufacturing process and the materials employed, and formulate an objective function taking into account the specific constraints of the desired design. Then, usually, the computational power is exploited to explore the space of achievable design and reach optima or sub-optima results. Both completely automatic and assisted and/or interactive techniques were created to achieve these goals, providing, when needed, control over the computer design process and thus avoid the expensive feedback loop of producing many physical prototypes that are not satisfactory.

Innovative usage A last and important part of the research work aims to push the boundaries of digital fabrication technology. In particular, the scope is not focused on improving the way these digitally controlled machines are traditionally used, but the target is to go beyond their intended usage. This research strand,

while still coping with the physical limitations of the available hardware, focuses on novel strategies of producing objects enabling new possibilities that could have impact on both the consumers and the manufacturing industry.

The scope of this thesis is mainly related, but not limited, to the last objective. In particular we provide new tools and computational approaches that, exploiting the available hardware and technologies, allow designs and production methods that were not possible before.

1.2 Contributions

Given the context, in this thesis we take advantage of geometry processing techniques, shape analysis and physical simulation to propose three techniques that improve the current state of the art on Digital Fabrication, by enabling novel ways of producing objects. The ultimate goal of what we propose is to exploit current technology at its best and create cost-effective digitally assisted techniques for: i) the realization of medium/large objects; ii) the production of multiple object replicas; iii) the creation of soft objects.

In Chapter 2 we will review the state of the art on Digital Fabrication providing an overview of some of the recent and most relevant results obtained mainly within the Computer Graphics community, while the following chapters contain the technical contribution of this thesis.

In Chapter 3 we investigate a cost-effective method to reproduce objects at a medium-large scale (in the order of the meter), which is not possible with commonly available 3D printers. This is due to two constraints: i) the printing volume of 3D printers is usually limited and ii) although the objects can be split in multiple pieces, the cost of the base material (generally ABS or PLA) for producing such objects is very high. In particular, we will show a method that allows the manual construction of medium- to large-scale objects that reproduce some given target 3D shapes. The idea is to provide a simple way to produce a structure that, sacrificing the reproduction accuracy, approximates the original model. The approximation still provides a reasonable perception of the object but is very cheap if compared to the full object reproduction. This is performed by automatically generating a set of planar pieces that can be fabricated by laser cutting sheets of rigid material (e.g., cardboard, wood, acrylic). The produced pieces can be then interlocked together to obtain an approximated, structurally sound and visually pleasing reproduction of the input shape.

The output of the method consists of the 2D design of the planar pieces and the assembly instructions required to properly interlock them in the final shape. The

technique is a work made within the Visual Computing Lab (ISTI - CNR) in Pisa and it is published here:

- Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. Field-aligned mesh joinery. *ACM Trans. Graph.*, 33(1):11:1–11:12, February 2014.

In Chapter 4 we present a novel production pipeline to fabricate multiple copies of medium- to small-sized objects. In general additive manufacturing technology is the perfect choice to produce single, unique designs because the cost of producing one object is justified by its uniqueness. On the other hand, when we think of mass-produced objects, the cost of 3D printing are prohibitive and usually industrial pipelines are setup, as these allow to scale down the cost per produced unit. In between these extremes there is limited scale production, which is very often performed as a manual process. If we consider the simple use case of an artist that needs to produce a limited amount of copies, the production is carried on by creating one master copy of the desired object, which is then used to produce a cast (mold). The latter is ultimately used to fabricate multiple replicas using liquid casting or similar techniques. This process is usually difficult and time consuming as molding is a manual process that requires expertise in order to produce a valid design, especially for topologically complex objects. In practice this technique very often requires to split the master object into multiple pieces such that the resulting molds can be easily cut to extract the casted objects. In our work we provide a new method that allows to produce artifacts in limited series, combining both additive manufacturing and traditional mold casting techniques. In particular, we devise a method to automatically produce, starting from a digital 3D model, a flexible shell design that can be used multiple times to physically fabricate, via liquid casting, many replicas of the provided model. The flexible shell (the mold) is a single piece and can be directly produced with a 3D printer.

This is a joint work between the Visual Computing Lab (ISTI - CNR) and the Institute of Science and Technology - Austria and it is published in:

- Luigi Malomo, Nico Pietroni, Bernd Bickel, and Paolo Cignoni. Flexmolds: Automatic design of flexible shells for molding. *ACM Trans. Graph.*, 35(6):223:1–223:12, November 2016.

In Chapter 5 we illustrate a work that deals with the mesoscale structure of artifacts. We devised a technique that is able to produce, using a single-material 3D printer, tileable microstructure elements that exhibit a wide range of elastic properties. Varying the topology and the geometry of these structures we are able to control the elastic response of such elements. These microstructures can be used to fill the volume of a voxelized object, allowing spatially varying elastic properties. We also provide an optimization strategy that allows to define, for an object, a target deformation behavior for a prescribed set of loads and automatically compute the

distribution of microstructure elements that is able to achieve it. Compared to the other contributions, in this work we take full advantage of additive manufacturing processes as structures at this scale can be cheaply produced with 3D printers only.

This work has been made as a collaboration of the Visual Computing Lab (ISTI - CNR) and the Courant Institute of Applied Mathematics (NYU). The results are published here:

- Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. Elastic textures for additive fabrication. *ACM Trans. Graph.*, 34(4):135:1–135:12, July 2015.

Chapter 2

State of The Art

In this Chapter we will show an overview of the most recent and relevant work done in computer graphics relative to Digital Fabrication. In the following we provide a general classification abstracting from the specific technical domain (e.g., geometry processing, physically based simulation, shape analysis) involved in each work. We rather take into account the objective of the research itself and its main usage, making no distinction on the technical aspects involved. In order to offer a wide overview of the state of the art this subdivision is intentionally orthogonal to the technical contribution of the thesis.

For more information about the technical aspect of the different digital production technologies, in particular Additive Manufacturing (3D printing), we refer to the book [GRS14]. For a review of established methods and recent advances in computational design and fabrication please refer to the Siggraph courses [LWSW14, UBM15, SBM16].

2.1 Classification

On the basis of accuracy and characteristics of the produced artifacts we can classify the various methods into two broad categories: accurate and illustrative fabrication techniques.

Accurate Fabrication Methods Industrial prototyping aims to create a tangible representation of an arbitrary complex object which is usually modeled using Computer Aided Design tools. Unfortunately, most of the traditional industrial fabrication techniques reach affordability for medium or large scale productions: implementing such a process to create a single (or a few) prototype(s) is not efficient. In general, the more the object has a complex shape, the more the manufacturing process will result complex. To cope with this specific industrial need, an emerging class of digital fabrication devices and technologies has been developed for the small scale production of arbitrary shapes. Most of this

techniques have been popularly known as 3D printing. These modern devices enable almost exact copies of a given shape to be reproduced and accurate methods are aimed to obtain high quality reproduction or to simulate physical or functional characteristics.

While 3D printing technologies can perform surprisingly well in terms of accuracy and geometric complexity, these techniques still have several restrictions. The workspace is usually very small, the printing process is time consuming, and, in order to produce a high quality reproduction, the input geometry has to satisfy both geometric and static constraints. Many approaches have been proposed in literature to overcome these limitations for example by splitting the objects into several parts for increasing the size of the printed object, or changing the basic printing technology to increase geometric accuracy and visual appearance, or even to employ different sets of materials. Along this line of research, another class of accurate methods exploits the advantages of additive manufacturing to fulfill specific constraints, such as structural stability or functional characteristics of 3D printed objects. These methods make traditional use of digital fabrication technologies and try to improve their performance when dealing with problems related to their use.

Illustrative Fabrication methods These methodologies generate illustrative representation of a given object or make creative use of digital fabrication technologies to generate original and artistic designs.

As explained above, most available printing technologies aim at producing an accurate copy of a tridimensional shape. However, fabrication technologies can be also used as a mean to create a stylistic representation of a digital shape, to simulate a physical effect or to provide a particular interaction feeling with the fabricated model. We refer to this class of methods as Illustrative Fabrication methods.

The needs to overcome printing limitations resulted in new creative and practical uses of these technologies which led to radically new approaches for the fabrication of digital shapes. So, for example, instead of fabricating an exact copy of the digital shape, many approaches aim at fabricating a plausible simpler representation, or exploit fabrication to reproduce intended shapes into a completely novel medium like light and shadow, or with a different artistic expressive language like bas-relief. All these novel sets of methods represent a consistent part of the research effort within the CG community. Their scope is to abstract the shape of a digital object and create an artistic, yet often practically useful, real-world illustration. In this sense, their performance cannot be simply quantified by the accuracy of the representation but rather in the capacity of offering an original, interesting geometric performance. While such illustrative representations may be often motivated by reducing costs and printing time or to overcome conventional 3D printing limitations, they can be also intentionally used by artists to design particular optical or stylistic effect.

The overall process of sacrificing reproduction accuracy to advantage style is a

common path for many artistic contexts. In general, artists tend to emphasize and personalize their own style rather than strive for perfection and pure mimesis of reality. Moreover, painters and sculptors are in the continuous search for innovative materials and modeling techniques. In computer graphics, non-photorealistic rendering techniques can be preferred to realistic ones for some application domains. This occurs because the style of a geometric representation could be very useful to communicate a certain "shape mood" to the observer. We also believe that some of these ideas can be exploited in the fabrication community to create tangible representation of visual data.

Please consider that due to the heterogeneity of the methods shown in the following, the categorization is not stringent, as multiple techniques could be ambiguously classified.

2.2 Accurate Fabrication Methods

2.2.1 3D Printing optimizations

Additive manufacturing, better known as 3D Printing, is not a new technology and has been used for over twenty years in the context of rapid prototyping. Specifically, algorithms and methods for producing objects with technologies such as Stereolithograhy and Fused Deposition Modeling have been studied in detail in the past and a vast literature exists in the rapid prototyping and CAD community (e.g., concerning slicing [MKD99, MPVRD03, DPRW00] and supports generation [CJR95, YCS02]). In the following we will briefly overview some representative techniques recently developed within the Computer Graphics domain. For a comprehensive overview of technical characteristics, limitations, modeling and toolpath generation methods, related in particular to Fused Deposition Modeling (FDM) technology, we refer to the Siggraph course in [DGLC15].

Toolpaths and Slicing

In this context, toolpath generation for each layer of the 3D printed design is fundamental to produce a correct print. A novel technique that is able to perform this step in a clever way is [ZGH$^+$16]. The method is targeted to extrusion type printers and proposes an infill technique that optimizes the travel times of the extrusion nozzle while creating toolpaths for each layer. The approach is inspired by the fermat spirals, from which a new set of space-filling paths is derived called *connected fermat spirals*. One example of this layering technique is shown in Figure 2.1. The advantage of this method is that the outer shell of each layer (the perimeter) is aimed to be made of a continuous line so the final look of the object is also improved, with respect to standard techniques. Also, the paper shows that

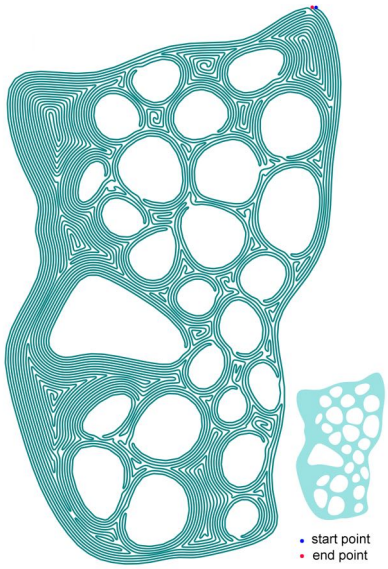


Figure 2.1: One example of space-filling layering obtained with connected fermat spirals [ZGH+16].

this way of (in)filling optimizes the robustness of the printed model. Moreover, from a technical standpoint, this method helps to minimize the number of distinct extruding paths and, as a consequence, the points where the machines start and stop extruding, which are the ones where artifacts generally occur. Another work, also targeting slicing, is presented by Wang et al. [WCT+15] and optimizes the slices (layers) thickness in order to save printing time while at the same time preserving the quality of the print. Time and visual quality optimizations are performed as two sequential steps. Additionally, the method proposes a saliency-based segmentation technique to divide an object in different portions that would have different slicing optimizations. Instead of printing individual parts and then assembling them together, they devise a method that allows to print all parts of the input model as a whole. The results obtained show that this approach can effectively save printing time without affecting the visual appearance with respect to uniform slicing.

Another technique that deals with toolpath optimization is proposed by Hergel and Lefebvre [HL14]. The method is able to improve the performance of dual extruder 3D printers to reduce the interference between the currently unused extruder and the partially printed portion of the object made with the other one (with different material). For this purpose the method performs three type of optimization: a) azimuthal optimization changes the Z orientation of the printing; b) the traveling path (toolpath) of the nozzles is optimized to avoid, when possible, the interference; c) double-walled *rampart* structures are added to the print to explicitly catch unwanted oozing material from the unused extruder.

Support Generation

Another important step for producing a correct print for stereolithography (SLA) and extrusion type 3D printers (FDM) is the generation of support structures, that prevent overhanging parts to fall down during the printing process. A number of automatic techniques is available on the market as most of companies that produce 3D printers, as well as other software companies, provide software that is able to performs this step. This is usually done in trivial ways and so there has been a number of works that tried to optimize this process. In order to reduce the material waste of overly dense support structures, Vanek et al. [VGB14a] proposed a generation strategy to produce minimal supports. Their algorithm starts by finding, for a provided 3D model, the orientation that minimizes the overhanging area that requires supports. Then they proceed to generate a tree-like support structure that minimizes the material used. For the same purpose, another approach [DHL14] uses scaffoldings similar to the ones used in construction sites, which are made of pillars and bridges. The algorithm selects support points among overhanging surfaces and builds scaffoldings to hold them. Differently from the method previously shown, the method also accounts for the physical stability of the printed model throughout the build process.

Decomposition

As described earlier, decomposition methods are a useful method to overcome 3D printers volume limitations, and many techniques have been proposed to automatically solve this and other issues.

Given an input mesh, Chopper [LBRM12] focuses on finding structurally sound and aesthetically pleasing cutlines to decompose a 3D printed object in a set of smaller pieces that fit the maximum printing volume, which can then be assembled together using connectors (plugs and holes). The algorithm uses BSP trees combined with beam search to explore the space of possible partitioning of the model, where each plane of the tree represents a cut on the model. The search combines different objective functions to optimize the number of resulting parts, connector feasibility, structural soundness, and seam placement. Once the optimized BSP (and the resulting object partioning) is obtained, plugs and holes are placed on the pieces, which can be individually printed and interlocked to each other to form the given input shape with a provided assembling order (Figure 2.2a).

Another method [HLZCO14] explored the decomposition of shapes into pyramidal pieces. This characteristic is mainly aimed to reduce support material and allows to print decomposed shapes that do not need supports an can be assembled together in the final shape.

The Dapper method [CZL+15], instead, aims at solving together decomposition

and packing problem for arbitrary 3D shapes. Decomposition is performed into a usually small number of pyramidal parts, which are pack-friendly. Each part is then voxelized and a bounded beam search is used to generate the packing configuration. While trying to fit parts into the target volume often thin and fragile parts usually are cut away, but are reincorporated into the optimization process. Additionally, the packaging of the parts is done by creating an additional structure that holds the pieces in the desired packing configuration.

(a) (b)

Figure 2.2: (a) A model partitioned by Chopper [LBRM12], then 3D-printed and assembled; (b) an assembled horse model produced with CofiFab [SDW+16].

Yao et al. [YCL+15], proposed instead a different technique in which both decomposition and packing are solved using a volumetric level-set representation. They perform interleaved solving of both problems. In the first case, they optimize the structurally robustness of rearranged object, the length of parting lines and the contact area between pieces. On the other hand the objective is to minimize the total volume of the packed print, to prevent interlocking of packed pieces and to guarantee the feasibility of the assembly.

Another technique is CofiFab [SDW+16], which aims to reproduce objects mixing 2D laser cutting and 3D printing techniques to reduce the production cost. It allows to produce big objects by decomposing them into two components: a simple inner structure made by interlocking 2D laser-cut elements, and a set of 3D printed pieces that will be attached to the inner structure to compose the final shape (Figure 2.2b). This allows to reduce the cost with respect to a full 3D print of a big object while maintaining a good quality of the reproduction. This is performed with a dual objective optimization: create a stable inner structure that can be assembled from 2D laser cut pieces while minimizing the total volume of the 3D printed parts (which dominates the overall cost).

2.2.2 Appearance

For both FDM and SLA technology, the support structures that holds the overhanging regions of the printed model generate artifacts when removed. Automatic support generation is usually targeted to reduce material usage and printing time, but the appearance aspect is also to be considered.

This problem is accounted in the approach [ZLP+15], where data collected via a user study and a machine learning technique is used to automatically understand the best printing direction of an object. The scope is to avoid supports structure artifacts in the most (visually) interesting parts of the object. For each candidate printing direction many optimizing criteria are available: mesh saliency, viewpoint preference, smoothness, and the size of the contact area for the required supports. A user study was performed to feed a machine learning algorithm that is able to automatically associate weights to each criteria, so that for any given object the technique is able automatically choose a set of candidate printing directions that reduce the impact of artifacts induced by support structures.

Additive manufacturing is known to produce models that have different precision depending on the slicing direction used in the process. This results in aliasing artifacts on the printed model because of the anisotropic precision of 3D printers: more precision on the XY plane with respect to the Z axis (due to the discretization imposed by the layer thickness). Hildebrand et al. [HBA13] introduced a technique to better decide the slicing direction for the additive manufacturing process, aimed to minimize the volume discrepancy introduced in the printed model. Initially, a normal-clustering algorithm is employed to find the three orthogonal directions that best fit the normals of the model surface. Subsequently, the model is voxelized along this reference system and the error per voxel is computed considering all three directions of printing. In the final step voxel-clustering is iterated over half-spaces, choosing among them the ones with optimal printing direction and that can minimize the error produced. The final result is a volume subdivision of the model with printing direction association.

Another set of works, rather than minimize the artifacts produced with 3D printing, aims to reproduce visual properties of the models.
Cignoni et al. [CGPS08] proposed a technique to enhance the visually perceivable details of 3D printed objects. The polymers commonly used for additive manufacturing have the undesired effect of decreasing the perception of geometric details. This is due to the sub-surface scattering (SSS) effect of the printing material and the impact becomes more evident when decreasing the scale of the produced object. The authors, thanks to the modern multicolor 3D printers, counterbalance this effect by coloring the model with an appropriate texture. The texture embeds the difference between the desired shading appearence (of the purely opaque object) and the undesired shading behavior.
In [HFM+10], vice versa, a technique is proposed to print objects simulating the sub-surface scattering effect. The authors proposed a framework to produce models with a desired sub-surface scattering (SSS) response taking advantage of multimaterial 3D printers. Using a goal-driven optimization function they find an arrangement of the base materials in the volumetric space of the model that is capable of approximating an input SSS function. A similar work is presented in [DWP+10].

To provide color shading to prints produced with dual extruder FDM technology

Reiner et al. [RCM+14] proposed a method to use dual color mixing for simulating this effect. The core observation is that discreetly mixing two colors on the surface of a printed object it is possible to obtain a dithering effect that simulates color intensity. In order to obtain this, they optimize the toolpaths of the printing heads (nozzles) while maximizing the length of continuous spans of filament emitted to compose the outer shell. The solution they provided alternates layered printing between two differently colored material, perturbing the outer shell path of the model with a sinusoidal function and, by phase-shifting this perturbation, they are able to achieve a dithering effect (Figure 2.3a).

The work in [BAU15] aims to accurately reproduce full color appearance of 3D printed model. Instead of using common technology that is able to achieve these kind of prints, like powder-binding o layer laminate technology, they focus on a technique that employs polyjet printing [Str14] and transplants the methods already used for 2D color printers to 3D printing, using a limited set of colors (CMYW in this case). The method models the diffusion of color into translucent material and, taking advantage from half toning, optimizes, on a voxel basis, the distribution of colors, achieving accurate color reproduction (see Figure 2.3b).

(a) (b)

Figure 2.3: (a) The dithering effect obtained with the method in [RCM+14]; (b) printed object with the color reproduction approach of [BAU15].

More recently, new fabrication technologies have been studied in order to reproduce the color appearance of objects. One technology is hydrographic printing, which allows to transfer color images onto the surface of a 3D object. An image printed on a film is laid out onto the surface of water and by dipping an object into the liquid and through the film, a chemical process sticks the film onto the object surface. The works in [ZYZZ15, PDP+15] take both advantage of this technology in order to create, on the surface of a 3D printed object, a custom

(a) (b)

Figure 2.4: (a) 3D printed model colored with the hydrographic printing technique in [ZYZZ15]; (b) colored miniature stomp obtained with the thermoforming method in [SPG+16].

colored texture perfectly aligned with the model. For both methods the idea is to simulate the viscoelastic behavior of the printed film while the object is immersed into the liquid. By knowing the deformation occurred within the film, it is possible to calculate a deformed version of the texture to be printed so that perfectly maps onto the object surface (Figure 2.4a). This is made possible by precisely controlling the dipping process, tracking the exact position and orientation of the model during this step. A similar technology, called thermoforming (or vacuum forming), was employed for the same purpose. This technology is widely used in industrial applications, especially for mass producing packaging. The production is performed by laying out a heated sheet of plastic material over a 3D design support, and then vacuuming the air between the two, so that the sheet takes the shape of the underlying support. In [ZTZ16, SPG+16] this technique is exploited in order to create 3D plastic designs with a high quality colored texture (Figure 2.4b). Similarly to hydrographic techniques, the approaches work by simulating the entire thermoforming process so that, by inverting the distortion of the plastic sheet after being pressed against the support design and vacuumed, it is possible to print a deformed version of the texture image on the flat plastic sheet.

For a detailed overview of appearance-targeting techniques we refer to [HIH+13].

2.2.3 Robustness

3D printed designs are easy to produce but there is no guarantee on the robustness of the resulting objects. Especially for objects that have a physical functional purpose or designs that are often solicited with external loads (e.g., toys), it is very difficult to predict, without knowledge of mechanics or industrial design experience, if the desired object possesses structural flaws. As an extreme example, it is very possible

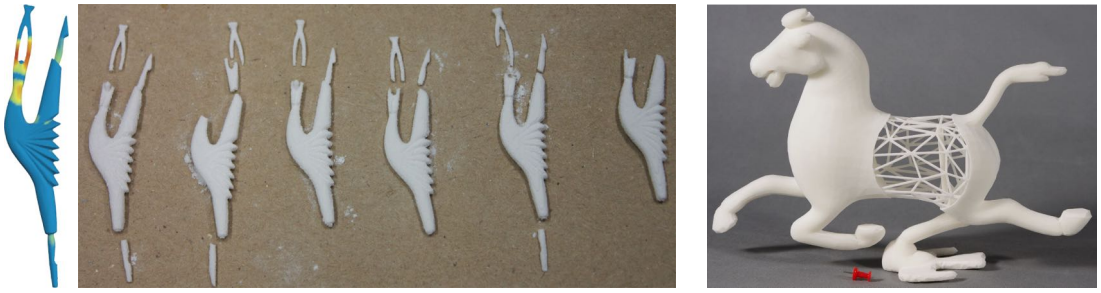
that even simple 3D printed designs are not stable while withstanding their own weight and this must be accounted for their fabrication [CZXZ14].

Stava et al. [SVB+12] proposed a technique to strengthen a 3D printed object removing the weakness that may occur in a model to be fabricated. The method performs structural analysis for a set of stable configurations, i.e. the model lying on a flat surface and sustaining its weight. Furthermore, a set of plausible *pinch* configuration is computed and analyzed: this is because the authors considered, as a worst case scenario, that the object is grasped with two fingers, with maximum force applied in two locations only. To improve structural stability three strategies are employed: thickening the thin and problematic regions of the object, adding struts in the occluded regions or in less visible parts and hollowing the model to relief the stress due to the weight force exerted by the object mass. The proposed algorithm follows an iterative scheme: initially it evaluates the benefit of the possible corrections using a linear combination of their efficiency and the visual impact derived from them, and then the best correction is chosen and performed. The algorithm converges when all the problematic configurations are eliminated.

Zhou et al. [ZPZ13] improved the work in [SVB+12] by proposing a worst-case analysis in which the aim is to automatically individuate all the possible problematic regions of the object, without a priori computation of applied forces. In the specific case modal analysis is employed and the method, given the base material properties, can effectively predict in which part a fabricated model is more likely to break after a drop test (Figure 2.5a). This kind of evaluation normally requires an extremely expensive computation because a FEM (Finite Element Method) analysis should be performed for every possible configuration of forces applied with every possible angle to the model surface. The proposed solution, instead, reduces the problem by considering only the weak regions detected with modal analysis. This approach yields a speed-up that is dramatic with respect to the brute-force solution. The algorithm is extremely robust and, thanks to its speed, can be inserted in a 3D printing pipeline in order to show to the user the possible issues of the desired model.

Another approach [XXY+15] presents an interactive tool to efficiently test potential designs flaw of the shapes to be printed. Selecting a set of constrains for an object (e.g., fixing a region and applying loads on another region), the system performs an optimized FEM analysis and highlights weak parts that could break in the final design, offering to the user candidate modifications for the problematic shape regions and visualizes the outcome of the performed changes.

A common solution for improving the strength of a printed object, although costly, is to allow solid filling of the interior. To reduce material waste, instead, an established approach consists of hollowing the model, i.e. printing only the external shell of the object using adequate thickness. A middle ground solution between these two extremes (to increase the robustness while keeping the cost down) is to fill the empty space inside the object with a lattice surface that follows a predetermined pattern. This approach is implemented in many 3D printing softwares



(a) (b)

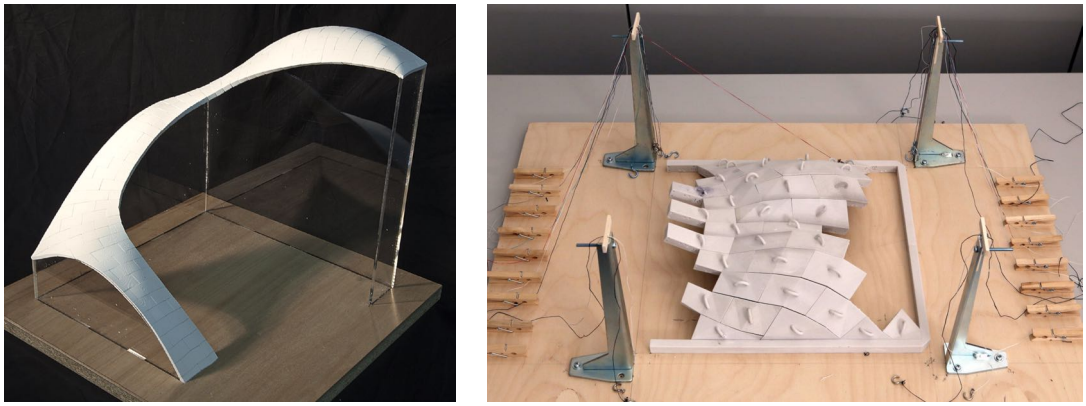
Figure 2.5: (a) A 3D model showing weak regions found with the method in [ZPZ13] and several 3D printed instances after a drop test; (b) a 3D printed model exposing the inner skin-frame structure generated with [WWY+13].

but alternative solutions were proposed in research. The work of [WWY+13] aims to efficiently reduce the material waste as well as improving the structural stability of the object. Instead of producing a uniform lattice inside the object volume, the proposed approach generates an inner frame structure, similar to the one used for bridges and buildings (Figure 2.5b). The realized frame would then sustain the outer surface of the object. This structure is made of struts connected to each other at node locations. The technique works in two steps: i) an internal frame structure is built, placing more nodes on more stressed regions of the outer shell of the model; ii) alternating phases of topology and geometry optimization are performed on the inner structure, having as final objective the volume reduction. In the topology optimizing step the radii of the frame struts are reduced. In the other step the geometry of internal node is changed, reducing the length of the struts, their radii and, limitedly, the position of internal frame nodes. The iteration is repeated until the volume of the structure stops decreasing. In addition, the authors proposed an algorithm to guarantee printability for extrusion type 3D printers: the solution adds extra supporting struts external to the object volume, which compensate for the lack of support material during the printing process.

The same objective is achieved with a different strategy: the method in [LSZ+14] provides a solution that performs selective hollowing of the printed object while maintaining a high robustness. The authors used a hollowing optimization strategy that generates, as inner structure, a honeycomb lattice that possess high strength-to-weight ratio. The lattice is generated from voids that correspond to cells of a Voronoi partitioning of the volume, and the optimization finds, considering a prescribed load applied, the void distribution with maximum strength-to-weight ratio.

2.2.4 Architectural modeling

The results obtained exploring digital fabrication can have unexpected uses if considering big architectural models. One example is masonry structures, which



(a) (b)

Figure 2.6: (a) One small-scale example of self-supporting masonry structure designed with [PBSH13]; (b) a partially assembled masonry structure supported by a single chain only [DPW+14].

are constructions made of bricks (blocks) that are able to withstand their own weight relying only on the compression forces between the elemental blocks (plus, sometimes, additional tension provided by cables). To support the design of such structures Whiting et al. [WSW+12] proposed a technique that, starting from an initial masonry design, is able to suggest and perform modifications to improve the overall "structural soundness" of the model. This is performed on masonry buildings blocks as well as cables that compose the model. The method provides a closed form for the soundness evaluation of the structure and optimizes the displacement of block vertices following a gradient-descent procedure. In addition, this method can also cope with constraints such as the thickness of the masonry blocks or add an additional objective function like volume minimization.

The work in [PBSH13], instead, provides a method that allows to automatically produce a masonry structure starting from an height field that defines its shape. This method is another good example of CG techniques applied to the fabrication domain. Initially, the height field domain is tessellated with triangles. The idea is to approximate the flow of forces along the surface using a cross-field, which is computed on the triangular geometry using a modified mixed integer quadrangulation algorithm [PLPZ12]. The algorithm mixes soft and hard constraints accounting for different properties of the model surface in order to plausibly align the resulting field with the force flow. The model domain is then quadrangulated; the dual connectivity graph with respect to this meshing is computed and used as a force diagram. Using both the primal and dual diagram a stable configuration is found by minimizing the distance of the stable design to the input provided. As a last step the method uses a structurally informed algorithm to generate hexagonal blocks for the final model (Figure 2.6a). Deuss et al.

[DPW+14] proposed an algorithm to generate a work-minimizing construction sequence for a given self-supporting structure like the ones generated in the method previously shown [PBSH13]. The algorithm decomposes the masonry structure into arcs and stable regions by analyzing the static equilibrium, and produces a construction sequence that starts from the arcs. The goal is to reduce time and cost by minimizing the number of addition and removal of support chains, which are normally used when building masonry structures, and represents the variable part of the total effort for the building process (see Figure 2.6b).

Concerning freeform designing, it is common practice to divide up the original shape into different components, which are fabricated separately but assembled together to produce the desired shape. Like for blocks in masonry structures, in architectural modeling, the original shape can be also subdivided into a finite set of triangular [SS10a] or quadrilateral [FLHCO10, EKS+10] basic panels. A method to fit a freeform shape with a set of single direction bendable panels (like wooden panels) is proposed in [PHD+10]. To further improve the smoothness of freeform surfaces, [BPK+11] introduced the so-called Circular Arc structures.

2.2.5 Articulated

Other techniques are conceived to fully exploit the power of additive manufacturing technology. These techniques are able to provide additional value to the 3D printing usage by directly printing articulated objects that can have different configurations. For example, in [CCA+12] the authors provide a pipeline that starts from a 3D mesh and its rigging (skeleton-like structure) and produces 3D-printed articulated models that require no assembling. One of the contribution of the paper is a physical cage ball-joint design that is directly printable with current additive manufacturing processes and, at the same time, is suitable for articulating models. The pipeline, starting from the 3D object, fits on the mesh the spherical joints corresponding to the rig articulations and segments the model in pieces that are connected by them. Using the additional rotational constraints provided with the rig, a set of automatisms are able to modify the geometry around the articulations to allow the desired rotational degrees of freedom for articulating the pieces.

Similarly to the work just mentioned, Bächer et al. [BBJP12] provided a similar pipeline that allows to produce models starting from a completely skinned mesh using both hinges and ball-and-socket joints. In addition they implemented an algorithm to avoid joint-to-joint collision; for this reason they are able to use completely skinned mesh to produce models with much more joints and that enable a wider range of poses with respect to [CCA+12].

The work of [STC+13] is instead tailored to produce physically deformable models using multimaterial 3D printing, starting from a 3D mesh and a set of deformed poses. The method uses an elastic model for the finite element approximation of the volume to measure the elastic energy with respect to its neutral pose. The framework allows the user to set “actuators” on the model,

which will be used to provide the external forces to achieve deformation. Additionally, the system is able to automatically produce a set of optimized actuators. The result of the analysis yields a continuous distribution of two base materials (one soft and one stiff) for the model volume, that is then discretized in a successive step to allow a feasible 3D print. With this method the produced model possesses the deformation properties needed to reproduce the input poses by applying forces in the actuators location.

The concept shown above is extended in [ZLW+16], which allows printing articulated models that can deform by just varying the thickness of the object shell. By using of a data-driven model, the method can optimize the shell thickness of the printed object to allow a prescribed deformation when the artifact is subject to specific loads.

2.2.6 Elastic Printing

Nowadays 3D printers are able to print using many materials: PLA and ABS are the most popular ones but are also very hard; there exist softer materials that can be employed with 3D printing but it is still difficult to control mechanical properties of objects and obtain user-desired characteristics. In [BBO+10] the authors devise a pipeline that is able to actively reproduce a target material behavior when subject to forces. The work employs a non-linear model for a set of known materials that can be used with multimaterial 3D printers. To capture the physical characteristic of the target material and to tune the underlying model for the set of base materials employed, they realized a setup to measure the response of a given material to forces. They used an actuator with a force sensor to probe the material, and a set of cameras to retrieve, with optical methods, the physical deformation of the material sampled. The objective is to reproduce a given target material by building a multilayer reproduction made with a combination of the base materials. This is achieved using a particular branch-and-bound algorithm in which different combinations of the base materials are explored. The results are satisying as the 3D-printed multilayer object reacts to forces similarly to the simulated model and with minimal deviation with respect to the real material to be reproduced.

Alternatively, to reproduce soft elastic objects with standard single-material 3D printers, the works in [SBR+15, MDL16] suggest printable microstructures that result in several orders of magnitude softer elastic responses than the original printing material. The first work uses topology optimization to design voxel units that behaves with user-specified elastic behavior. Martinez et al. [MDL16], instead, achieves soft elastic response by filling the object volume with a procedurally generated Voronoi foam, whose density determines the stiffness of the resulting object region. These results are only possible by using additive manufacturing techniques, since other manufacturing process are not able to produce these kind of designs. Another contribution on the same topic is provided

by our technique, which is explained in details in Chapter 5.

2.2.7 Optimizing Criteria

Many times, the design of an object is not purely related to its appearance or its robustness, but it is targeted to optimize a specific functional requirement of the object. In the following a few results on the topic are shown.

The work in [PWLSH13] shows a method that tackles the problem of balancing 3D objects. Given as input a 3D model to be printed and a desired up-vector orientation the proposed framework is able to rearrange the mass distribution of the object to guarantee that it will stand according to the provided input. To achieve this goal the technique uses two strategies: carving inner voids inside the object and modifying its overall shape (with bounds on deformation). The two strategies are put together into an energy minimization problem that tries to move the center of mass of the object such that its projection falls within the object supporting polygon, the portion of the object that lies on the ground plane. Since the formulation into a single problem is too complex to be tractable, the optimization algorithm is split in two alternating phases of carving and deformation that ultimately converges to a feasible solution.

Another example of such optimizing criteria is present in [BWBSH14]. Given as input a 3D mesh the goal is to convert it into a spinning design (spinning top or yo-yo). In this case the mass distribution is optimized to ensure that the primary axis for the moment of inertia aligns with the desired axis of rotation. This is performed, similarly to the previous work, by creating voids into the solid model. For more challenging cases, in order to fulfill the spin stability requirements, the author also provide an optimization strategy that uses a cage-based deformation of the original design and, when this is not sufficient, they optimize using dual material distribution combining materials with different density.

Wang and Whiting [WW16] tackle instead the problem of buoyancy optimization. The objective of this approach is obtain floating objects with prescribed orientation and floating level. This problem can be reduced to reaching the mechanical equilibrium of the object, by ensuring that the total force and total moment acting on it are zero. In the solution an external shell is always present on the resulting designs to make the printed object watertight. Similarly to previous approaches, in order to achieve the desired result, an optimization strategy is employed to create a per-voxel fill distribution inside the object.

The work in [CLD+13] is more ambitious and aims to integrate in a single framework the ability to fabricate objects with any desired physical properties. Usually users work with 3D-printing materials piloting directly the printer they have at disposal (i.e. indirectly, for each voxel of the produced model the material to be employed must be specified). Often this is a struggling process that lasts until the desired physical behavior is obtained. Inspired by the works in [BBO+10, HFM+10], the proposed framework starts from the base specification of

the materials and the machine employed to obtain a fabricable object with the user specified characteristics. The approach proposes a reducer-tree data structure and a tuner component that work on common reusable blocks in order to obtain desired specifications in a common way. This allows to combine different specification for the same 3D-printed object (e.g., albedo and subsurface scattering, elasticity and color, etc.).

2.3 Illustrative Fabrication Methods

2.3.1 Alternative Fabrication Processes

In this section, rather than overviewing ordinary usage of fabrication technology for illustrative purpose (e.g, [DLL+15, CZX+16]), we focus on methods that are designed to employ materials and devices that are very popular and inexpensive, such as paper, wooden flat lists or iron wires. This is particularly true for shape approximations, that can be fabricated by using simpler processes than the classical 3D printing. This class of methods does not require sophisticated fabrication devices, and due to the simpler fabrication processes, they scale relatively well with the overall size of the reproduction.

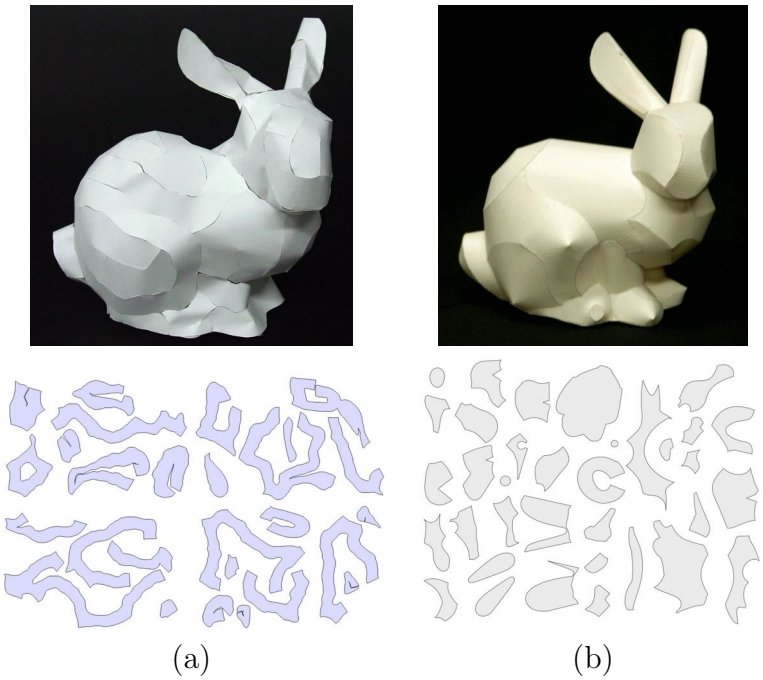


Figure 2.7: Assembled papercraft (top) and planar developable strips (bottom) produced with the approaches of (a) [MS04b] and (b) [STL06].

Papercrafts

Following this idea, several methods have been proposed to generate papercrafts from digital shapes. Papercrafts are sculptures that can be fabricated by cutting, bending and glueing paper strips. Most of the methods for papercraft design aims at deriving the best way to map a 3D shape onto a 2D flat pieces of paper. This problem has some similarities with the problem of deriving an optimal parametrization, however papercraft design requires low or no distortion in the mapping process (as paper is mostly inextensible).

One commercially available technique [Tam], consists of unfolding an input 3D mesh into a planar domain, eventually introducing cuts. The foldability, i.e. the possibility of flattening the object introducing a little distortion, is guaranteed only when the input model is triangulated at a very low resolution. While this method is really intuitive and simple to implement, the resulting objects inherit all the geometric artifacts which may derive from the low resolution of the input mesh; this is particularly evident in the case of smooth surfaces. This method has been extended to mitigate this approximation problem. Mitani et al. [MS04b] proposed a method to split the input mesh into a set of smooth, developable, paper strips that can be easily assembled together (Figure 2.7a). The technique uses least squares conformal maps [LPRM02] to derive an initial segmentation of the model surface. Boundary lines are then added to each segment to preserve concave or convex features. Triangular strips approximating the model surface are then generated from inner and inter-segment boundaries. Once unfolded, the different pieces correspond to the paper cut-outs that can be used to craft the paper model. The approach proposed by Shatz et al. [STL06] extended the previous methods to derive a more compact, structurally sound patch layout rather than simple developable strips (Figure 2.7b). Since the cut layout is simpler also the glueing process is significantly simplified. This technique starts from a very fine segmentation of the mesh and for each patch it associates a parametric unfolded surface (either a plane or a conic surface) which defines the approximation error. Adjacent patches are iteratively merged until the approximation error remains below a certain threshold. The cut layout is improved during the process to facilitate the assembling procedure.

Massarwi et al. [MGE07] improved the method of [STL06] by measuring the error with the Hausdorff distance and employing generalized cylinder components as developable surfaces (rather than restricting to planes and cones). This leads to much better results, as the paper cut-outs obtained cover a wider range of cases and thus better represent the original object.

Computer-generated 2D papercuts [XKM07] reproduce images by generating a single connected component cut out of a sheet of paper.

Other works, instead, take inspiration from paper pop-up illustrations. Pop-up illustrations are 3D figures that arise when a flat arrangement of paper sheets is

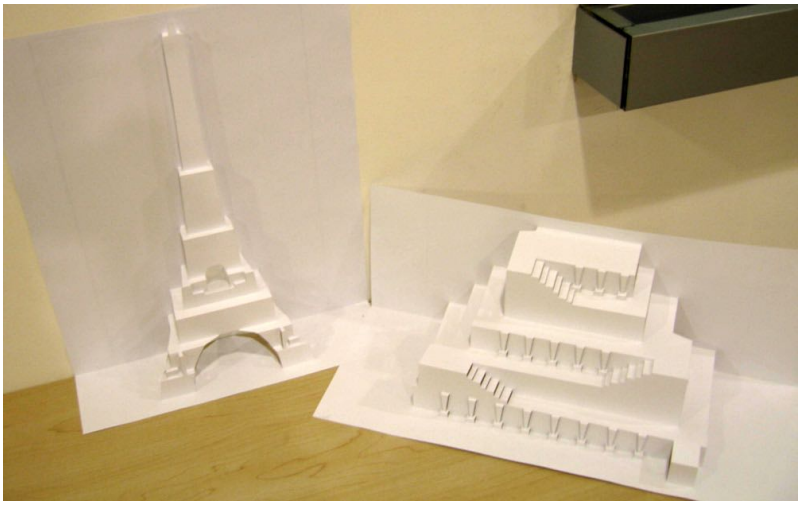
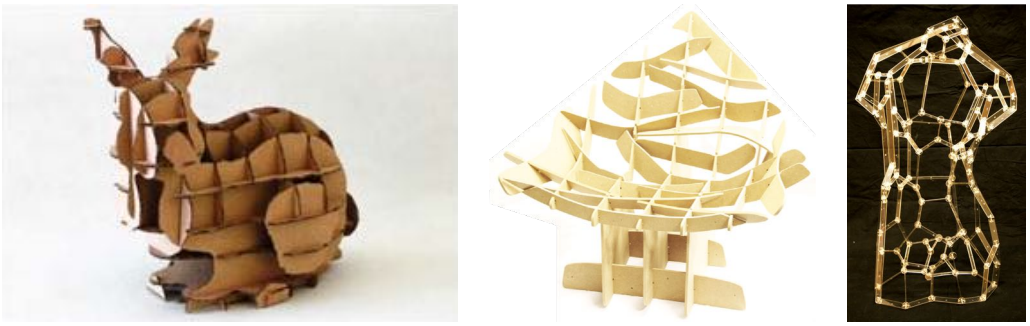


Figure 2.8: An example of pop-up architecture generated with the technique in [LSH+10].

opened. Pop-ups have two states: open (showing a 3D figure) and closed (reduced to a simple sheet of paper). Mitani and Suzuki [MS04a] proposed an interactive tool to design *origami architectures*, a particular type of pop-up designs made from a single sheet of paper, without any additional parts pasted on. This method produces a completely planar configuration of folds and cuts starting from a user-designed set of horizontal and vertical polygons that moves to compose the target shape, once the sheet is unfolded.

The technique shown in [LSH+10] improves the previous work providing a method to automatically produce an origami architecture starting from an architectural 3D model. The output is a set of cuts and folds to be applied on a paper sheet. An approximated reproduction of the input model appears when the sheet is unfolded (Figure 2.8). Additionally, the algorithm guarantees that the pop-up model is stable, i.e. the popping model has a unique unfolded configuration. A followup article [LJGH11] tackles the problem of computer assisted design and automatic generation of traditional V-Style pop-ups, commonly found in children books. The method guarantees the feasibility of the produced pop-ups, which can be embedded in an actual book.

Origami is a papercrafting art that has been know for centuries. Traditionally origami are manually designed, however a generation of CG tools has been developed to automatize their design. Tachi [Tac10] proposed a novel method to create an origami craft depicting an input polyhedral 3D surface. Using tuck folding [Tac09] the method allows to automatically reproduce an input surface using a single sheet folding pattern.



(a) (b) (c)

Figure 2.9: Interlocking planar slice designs obtained with (a) [HBA12], (b) [SP13], (c) Beam Meshes [RA15].

Interlocking

Planar slices can be easily manufactured using laser cutting and interlocked to create effective illustrative shape representations. Compared to traditional 3D printing, manufacturing planar cut-out is much cheaper and allows the use of many different material (e.g., cardboard, wood, acrylic, metal, etc.). The usage of these technology for shape approximation has recently grown interests.

In [MSM11] the authors proposed an algorithm to create minimalistic representations of 3D objects using a limited set of planar 2D cross-sections. A user study was performed to understand how humans approximate a given object using cross-sectioning planes. The position of the planar slices has been correlated with the relevant geometric features (ridges, valleys, symmetries, etc.) of the input shapes to derive the best weighting of each geometric feature and reproduce the user slice placement. Combining a plane-space exploration and the above weights, a greedy iterative algorithm selects planes that capture as many important features as possible and converges to a set of planar proxies that approximate the object in a similar way as an user would do. In addition, the authors propose to use the results to physically fabricate paper puppets using the planar proxies obtained as cut-outs, interlocked by using colored guides.

This approach has been extended by Hildebrand et al. [HBA12] to create constructible sculptures made of cardboard cut-outs (Figure 2.9a). The pieces are planar cross-section of the input model that are interlocked together through a simple slit mechanism. Using a modified BSP-tree, the proposed algorithm builds an approximated model by selecting and iteratively adding planar cut-outs with a specified inserting direction, which guarantee the feasibility of the insertion operation. To this purpose, the authors restricted the algorithm to use orthogonal or quasi-orthogonal planes. This choice was done because the orthogonal slit interlocking mechanism provides a tight grip between the planes. The final result consists of the sliding pieces and assembling instructions which guarantee that the

model approximation is constructible according to the provided insertion directions.

This method has been extended by [SP13] to allow for generic non-orthogonal planes configurations. The final configuration is designed to fulfill the requirements of fabrication, stability, and assembly [SP12]. Starting from a set of user-defined or automatically generated planar sections, the method automatically extracts from the input model the stencils for the resulting planar pieces and their insertion direction (Figure 2.9b). During the optimization it is ensured that two types of constraint are satisfied: (*i*) rigidity constraints, derived from the intersection angle and slit width of each interlock location, and (*ii*) slit constraints, guaranteeing that for each planar piece it exists an insertion direction and an ordering sequence to actually assemble them.

Beam Meshes [RA15] approximates a 3D model with a set of interconnected beams with rectangular cross-section (Figure 2.9c). The technique starts by approximating an input free-form mesh using a variant of the anisotropic centroidal voronoi tessellation; then, it uses the dual mesh to define the initial beam structure. Then a local-global optimization is used to planarize each beam (torsion-free). For fabrication purposes, planar joints are generated to hold collimating beams extremities, and, along with beams designs, they can be produced using laser cutting.

While the methods shown above offer automatic generation, FlatFitFab [MUS14] adopts an interactive approach. Specifically, it describes an interface to design objects composed of planar pieces that interlocks orthogonally. By using an interactive interface, the user can design the main structures from scratch, while the system provides physical feedback.

Chen et al. [CSaLM13] provide another interesting example of fabrication technique to closely approximate an input 3D object. Given a target number of faces and an optional user-specified saliency map over the model surface, the method iteratively clusters mesh primitives to produce an arrangement of polygonal planar faces that approximates the input shape. The method ultimately produces a set planar cut-outs that compose the approximated object, along with a series of (planar) joints that are used to connect them together. The output shapes can be fabricated with a 2D laser cutter and then assembled. Moreover, each face of the resulting object can be easily textured accordingly to the diffuse appearance of the original model. To provide color to the fabricated model, it is sufficient to print the colored cut-out and attach it on the external side of each corresponding piece.

Taking inspiration from IQlights lamps, the method proposed in [SCGT15] produces suggestive objects made of flexible rhomboidal planar elements. These elements can be combinatorially interlocked to produce different configurations. The paper presents a tool to interactively design such objects: starting from one base shape (a configuration obtained from a limited set of elements) the tool

Figure 2.10: One example of design obtained with the approach of [SCGT15] (right), composed by interlocking multiple elements (left).

allows the user to explore different options by modifying the object using two operations: extrusion, that allows to select the extrusion profile selecting among some base shapes, and merging, to combine distinct pieces by providing alternatives. The tool ultimately yields a physically realizable design that allows to produce a surface by assembling elements that have the same shape (Figure 2.10).

Wires & Rods

Another class of techniques uses rigid rods or more generic flexible wires. This class of approaches gives to the objects a distinctive style. Unfortunately wire meshes are in general very difficult to design manually. In the following we briefly overview the computation tools to allow automatic or semi-automatic design.

The approach of [GSFD+14] defines an interactive system to generate wire meshes (Figure 2.11). Wire meshes are composed by a single sheet wire mesh (metallic woven wires arranged in a regular grid). This type of structures is quite popular among artists that, for creating their designs, usually employ a trial and error approach by manually bend a piece of wire mesh to adapt it to a desired solid surface. Similarly to the manual approach, this tool allows the user to place a set of constraints and define the regions where the wire mesh should better approximate the input geometry. A coarse to fine solver is used to simulate the effect of the wire mesh once attached to the solid object.

Another use of wires is tailored to rapid prototyping. While originally 3D printing was conceived with this objective in mind, nowadays the printing process is still relatively slow because printing medium sized models can take tens of hours. The approach presented in [MIG+14] solves this problem by producing a wireframe preview for an input 3D model and generating printing instructions that are suitable for any FDM 3D printer. While this approach is created to speed up

Figure 2.11: Fabricated wire mesh design [GSFD+14] (center and right) of a 3D model (left).

Figure 2.12: Starting from the 2D input image on the left the approach in [ILB15] allows to produce the design shown on the right.

the design process of objects, it also offers an alternative to produce tangible illustrative representations that allows people to perceive the original object shape.

Metal wires are also employed to produce jewelry artifacts. Iarussi et al. [ILB15] provide an interactive tool for designing wire wrapped jewelry (Figure 2.12). The tool assists the user addressing the two main challenges of designing such pieces: (a) decomposing an input drawing into a set of wires based on artistic and fabrication constraints and (b) generating an ad-hoc 3D-printable support structure to guide the manual wire bending process.

The work in [MLB16] is aimed to create illustrative shape reproduction by planar bending of metallic rods, achievable with a 2D wire-bending machine. Starting from an input 3D model and a set of planar input contours, the system automatically computes a stable, self-supporting wire sculpture. The output of the system provides instructions for a 2D wire-bending machine and the obtained bent wires can be easily

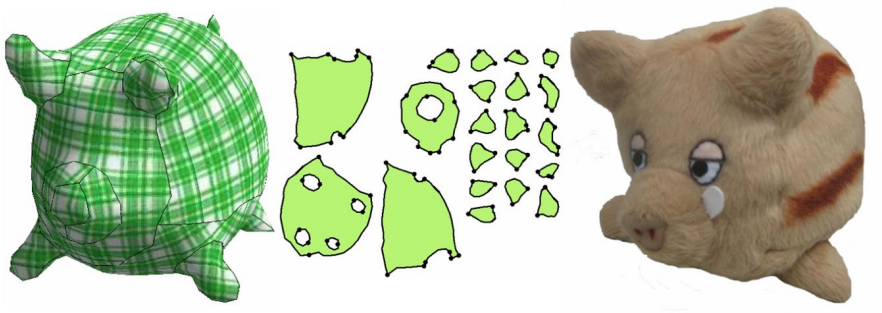


Figure 2.13: An example of plush Toy modelled with Plushie [MI07]

assembled without the need of connectors at contact points.

Combining solid objects and wires is also a very popular technique to produce small artifacts. In this context Beady [IIM12] is a method for assisting the design and manufacturing of beadwork, which is the art of connecting beads together using wires. The design process of these objects is really hard and the assembly requires hours or even days. Beady offers both an interactive tool to design such objects, but also allows to automatically produce a beadwork from an input 3D model. For produced designs the system offers step-by-step instructions to guide the wire path and beads insertion in order to ease the manual construction.

2.3.2 Soft Materials

Soft materials allow the production of objects with a very particular aesthetics and feel. In recent years, computer graphics research has developed numerous tools that allow users to design soft objects, which can be classified by the lower-level primitives the object is assembled off. Popular primitives are flat patches, which are sewn together; yarn, which is woven or knitted; or resin, which is either cast or 3D printed. Due to the distinct physical properties and fabrication constraints of these primitives, each of them requires specifically tailored design approaches.

Often, the mapping between low-level primitives and the final resemblance of the object is highly non-trivial. Addressing this challenge, Sensitive Couture, an interactive tool for garment design [UKIG11], allows to manipulate the patch geometry in 2D and provides an interactive visualization of the draped dress in 3D.

Plushie [MI07] is an interactive modeling system that supports users to design their own original plush toys from scratch, proposing a sketching interface for 3D modeling and editing (see Figure 2.13). Internally, the system constructs a 2D cloth pattern such that the simulation result matches the user's input stroke. Alternatively, assuming that the designer already has a target shape at hand, pillow [MI06] presents a workflow for interactive flattening of patches starting from a given 3D shape.

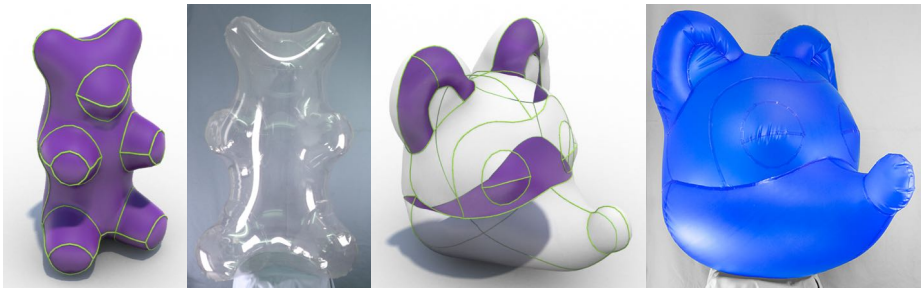


Figure 2.14: Examples of inflatable shapes produced by [STK+14], with the relative patch decompositions.

Inspired by this workflow, Skouras et al. [STK+14] present an interactive, optimization-in-the-loop tool for designing inflatable structures (see Figure 2.14). As the designer sketches the proposed placement of seams, the underlying optimizer supports reverse-engineering the physics of inflation, proposing a set of panels that best accommodate the desired seams and target shape. In all of these approaches, interactivity plays a key role in keeping the user in the loop for providing control over aesthetic considerations when finding a desired trade-off between the number and complexity of patches and the approximation quality of the 3D shape.

Alternatives to patch-based objects are inflatable membranes that can for example resemble complex shapes as balloons [STBG12], or knit fabric, which does not need to be locally flat. While machine knitting is a mature technology, finding low-level machine instructions for non-trivial shapes requires a deep understanding of low-level knitting operations. Making this technology accessible to a wider audience, McCann et al. [MAN+16] propose a compiler for 3D machine knitting that translate high-level shapes to machine instructions.

Recently, the development of new 3D printing technology and materials has created an interesting avenue for reproducing objects with soft materials. Hudson [Hud14] presents a new type of 3D printer which fabricates three-dimensional objects from soft fibers and was demonstrated for printing teddy bears.

2.3.3 Low Dimensional Representations of 3D Scenes and Viceversa

A *low-relief* (or bas-relief) is a way to represent a fully 3D shape using a thin layer of material. It can be considered a strictly 2.5D geometric representation of a more complex full 3D geometry. Similarly, an *high-relief* projects a 3D geometry onto a thin layer of material; but, in this case, it keeps part of the original "tutto-tondo" sculpting volume.

Since antiquity, low- and high-reliefs have been widely used in arts to

approximate three-dimensional shapes, to decorate cameos, sarcophagi or architectural elements, or to create the engravings minted on coins.

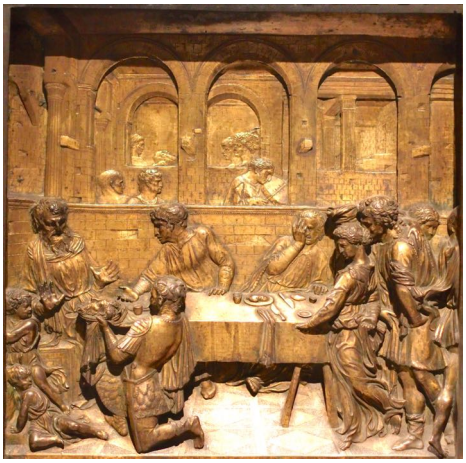


Figure 2.15: Herod's Banquet by Donatello (Siena, circa 1427), one of Donatello's earliest relief sculptures.

Low-reliefs represent a particularly reliable and lasting approach to represent a 3D shape onto an almost two dimensional space. The final perception of the representation is also greatly affected by self-shadowing. From a fabrication point of view, low-reliefs reduce significantly the production costs since they can be fabricated using 2D milling subtractive techniques. Due to the reduced amount of material used, this reproduction technique scales very well to cover large surface areas. A survey on the specific problem of digitally generating and modeling digital bas- and high- relief can be found in [KWC+12].

Cignoni et al. [CMS97] proposed the first method for the automatic generation of a digital low-relief starting from a 3D model. The main issue in this class of problems is to find an efficient way to compress the depth range of an arbitrary 3D scene into the very limited thickness of a bas-relief. A number of perceptive effects has been taken in account to solve this problem, like the fact that there is a relation between actual prominence of the shapes and their distance from the observer (farther objects are usually flatter) and that the most *important* parts and details should be marked with higher reliefs. The approach initially proposed in [CMS97] used a simple $1/z$. While this method is able to reduce the elevation of farther objects (see Figure 2.16), it had serious limitations in ensuring the visibility of all the important details of the scene. This technique has been extended to use histogram equalization to the involved depth ranges [SRML09] in order to better distribute the height in the bas-relief and preserve the details.

Another class of methods for generating bas-reliefs [WDB+07, SBS07, KTB+09, BH11, ZZZY13, JMS14] achieves better results by considering that details are characterized by normal variations and therefore they work in the gradient domain instead of using the depth values of the scene. In

Figure 2.16: A bas-relief of a cloister generated with the approach proposed in [CMS97].

these approaches the initial 3D model is used to generate a gradient field that is processed (eventually exploiting user input) in order to identify the most significant regions. The gradient field is finally integrated to recover the height field of the bas-relief. Other recent approaches [SPSH14, ASH15] try to pose the problem as a more general deformation problem. They work directly on the input 3D mesh rather than a depth map. Such approaches are able to manage the problem of creating high-relief in a direct manner, as illustrated in Figure 2.17, where the objective is not creation of simple height field.

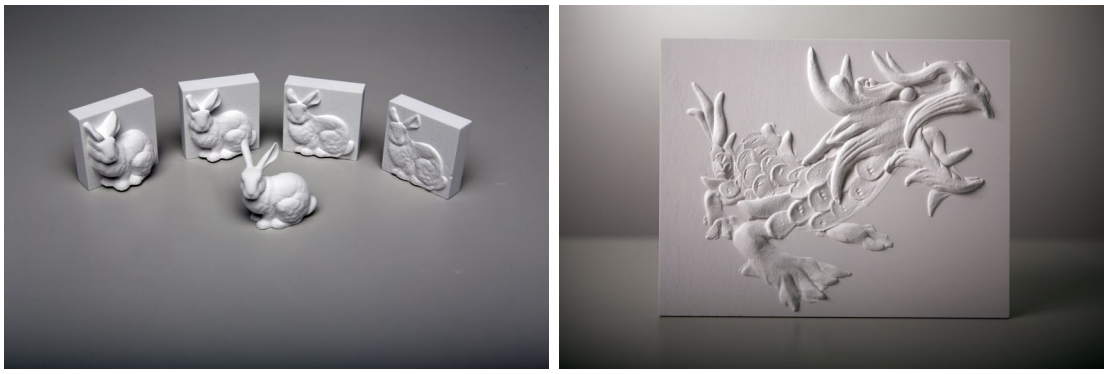
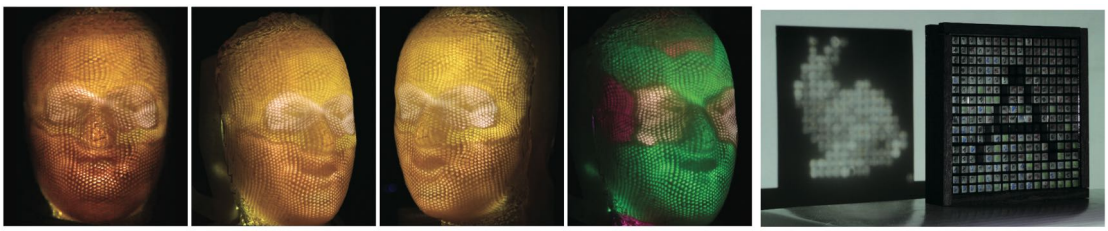


Figure 2.17: Two examples of high- and bas-relieves automatically generated with the approach proposed in [SPSH14]

.

It is also worth noting that bas-relief can be created starting directly from a simple 2D painting or a picture, translating them into a tactile 3D reproduction [RMP11, NR13], a task that has significant implications for visually impaired people as it offers a direct way of accessing purely 2D information by mean of 3D medium.



(a) (b)

Figure 2.18: (a) A human face shaped display fabricated by using the technique proposed by [PRM14]. (b) An example of display fabricated using the approach of [YIC+12].

2.3.4 Transforming Light and Shadow

A recent research trend has focused on how to transplant optical properties to a fabricated model. The final appearance of a fabricated surface depends on a wide numbers of physical parameters that can change at different scales. They may range from molecular scale, affecting wavelengths, to micro-surface roughness, affecting reflectance or even mesoscale that results in sub-surface scattering. The recent advances in the fabrication technologies allow to control, drive and predict the reflectance behavior in a direct way. A recent survey [HIH+13] discusses all the specific aspects of the problem of creating physical workpieces with controllable appearance characteristics, including the relationship of this problem with display technologies. However most of these methods try to replicate a real-world effect, striving for accuracy. On other hand similar ideas can be used to interact with light and produce nice illustrative representations.

Playing with Transparency

It is possible to design interesting lighting effects by distributing, within the volume of an object, materials with different refraction indexes. In [PRM14], the authors proposed the use of multi-material 3D printing to fabricate arbitrary surfaces with embedded optical fibers. The fibers are printed using two materials with different refraction indices: an external and an internal which exhibits a higher refraction index. The light propagates within the internal material and is trapped between the barriers provided by the external material. This allows to use the internal reflection to guide the light to follow specific paths inside the volume of an object. In this way, the light can follow complex curved paths and create nice illustrative effects, like displays of arbitrary shapes (see Figure 2.18a). The algorithm takes as input two meshes with a cross parametrization which indicates entry and the exit point of each possible light beam. The algorithm computes the optimal paths considering both fabrication and the physical constraints involved in the light propagation. As final output it returns a volumetric representation of the printing volume with the optimized distribution of the two materials.

The approach proposed by [YIC+12] exploits refraction to produce Pixel Art Images. A flat panel is uniformly tessellated by using a vocabulary of different sticks made of acrylate resin, each stick refracts the light in a specific direction. An automatic optimization algorithm composes the panel such that a parallel light source is redirected in specific locations, composing the final pixel art on a viewing surface. As shown by Figure 2.18b this approach allows to produce panels that appear as opaque glass but produce complex patterns when traversed by light.

Figure 2.19: A multilayer model [HBLM11].

A completely different approach [HBLM11] creates multilayer models of an input 3D mesh. A multilayer model is composed by a set of images embedded in glass panels that are stacked together (Figure 2.19). Each image displays the model at a different depth along the stack direction and their composition results in a smooth 3D representation. The algorithm optimizes the images such that the multilayer model does not show any artifact (self shadowing or image splitting) if rotated.

Caustics Design

Another class of approach aims to optimize the shape of an object such that the reflected or refracted light paints a desired caustic image on a flat screen. The main idea of all these methods is to slightly variate the normal on the surface of the object such that photons are deviated to compose a target image. Usually the surface is fabricated by using a milling machine. The approach proposed by [WPMR09], using precision CNC milling on a metallic surface, fabricates a set of reflective micro facets to deviate photons and concentrate them to match the target distribution specified by an input image. This approach has been extended [PJJ+11] to use curved micro facets to reflect or refract light. For more information on the topic, as well as a formal classification of the state-of-the-art techniques, refer to [HIH+13].

The approaches based on micro facets do not scale well when increasing the resolution of the image and it may creates artifacts for the reproduction of smooth features. These artifacts can be reduced by solving a global optimization problem [KEN+13] to retrieve a continuous mapping between input light and the amount of photons that hits the target surface. In particular in [YIC+14] the error in the target distribution is iteratively reduced following a globally smooth gradient field computed by solving a poisson equation. The approach proposed by [STTP14]

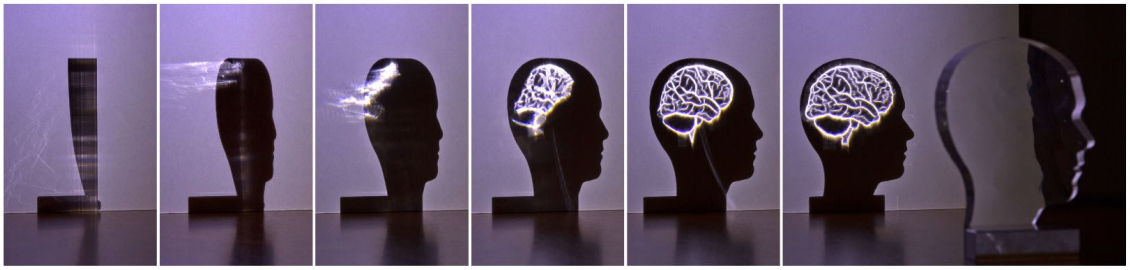


Figure 2.20: A transparent glass produced by using the approach proposed by [STTP14].

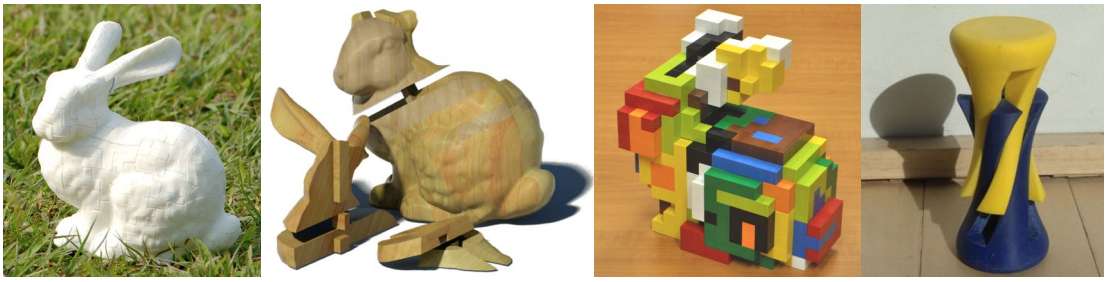
Figure 2.21: An example of Shade Art Design [MP09]

uses of piecewise-smooth surfaces to allow the reproduction of sharp features. An example of caustics produced by [STTP14] is shown in Figure 2.20.

Shadows Design

Shadows can be used as a medium to create suggestive effects. The approach proposed by [MP09] creates 3D shadow art sculptures. The user selects a set of input images that represent several target shadow silhouettes. The system retrieves a single 3D shape that is capable of projecting all the silhouettes. Obviously each silhouette appears when the object is illuminated from a specific direction. This effect is shown in Figure 2.21. This system uses a two-steps optimization; an initial step creates an initial shape by intersecting different volumetric *shadow hulls*, then the shape is optimized with as-rigid-as-possible deformations [SA07].

A completely different approach is the one proposed by [BBAM12]. ShadowPix are flat surfaces capable of showing several images when are illuminated from different directions. The effect is obtained by the self-shadowing of micro geometry that is distributed over the surface of the flat panel.



(a) (b) (c) (d)

Figure 2.22: (a) 3D Polyomino puzzles [LFL09]; (b) Burr puzzles of [XLF+11]; (c) Recursive interlocking puzzles [SFCO12]; (d) Dissection puzzles [SÍ12].

2.3.5 3D Puzzles

Digital fabrication technologies can be used to create 3D puzzles. The input shape is decomposed in a set of interlocking pieces where the assembly of the final model is the main scope of these methods. Thanks to modern digital fabrication techniques the pieces can be fabricated individually in a very efficient manner.

Generic Pieces

Most of the methods for automatic 3D puzzle generation poses great attention in ensuring that a feasible assembling sequence exists. 3D Polyomino puzzles [LFL09] is the first algorithm that generates interlocking 3D puzzles from a digital shape (see Figure 2.22a). Given as input a quadrangulated mesh, pieces are iteratively selected from a vocabulary and disposed on the surface of the object until the entire surface is covered. A valid assembly sequence is retrieved from the pieces graph and the partial ordering of inter-piece dependencies.

While this method is limited to create an external shell of the object, other methods decompose the entire volume. The Burr puzzle technique [XLF+11] distributes a set of nodes inside a given 3D shape. Each node is used to create a mechanism that collects six pieces together, guaranteeing their mutual interlock (Figure 2.22b). The approach proposed by [SFCO12] uses a more general placement of pieces. The object is split in a sets of interlocking pieces made of grouped cubic voxels (see Figure 2.22c). The peculiarity of this method is to admit a unique sequence of assembly/disassembly. In other words, for each step it is possible to remove only a single piece at the time, as the others are locked together. As a byproduct, the resulting puzzle is kept stable during the whole assembly process. Dissection Puzzles [SÍ12] decompose simple input shapes into pieces that can be interlocked by using complex movements like rotations (rather than axis-aligned translations).

Predefined pieces

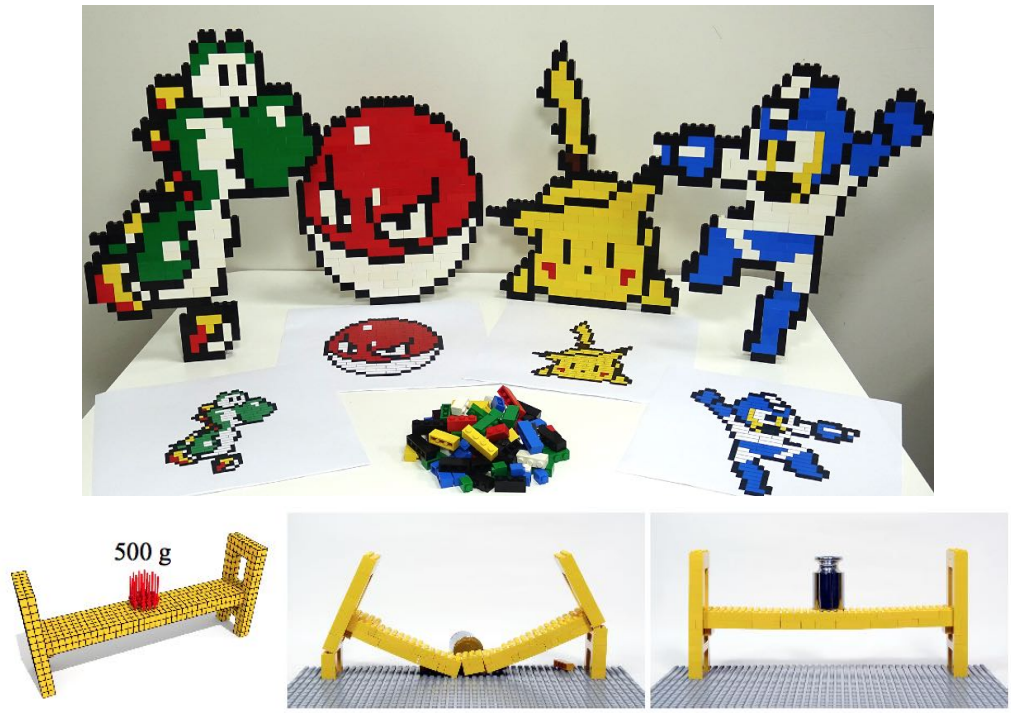


Figure 2.23: Top: an example of pixel art modeled with LEGO [KLC+15]; bottom: the approach proposed by [LYH+15] optimize lego composition to support external loads.

Other methods investigate the possibility of creating puzzles by using standard commercial piece sets, like LEGO pieces. A first method for the automatic generation of constructible models made of LEGO bricks has been proposed by [TSP13]. This algorithm generates a feasible and stable lego configuration and the relative assembly instructions. The approach proposed in [KLC+15] generates pixel-art sculptures using LEGO pieces (see Figure 2.23, top). This approach includes a simple optimization step to improve stability and to ensure that the sculpture stands in place once assembled. Other methods start from a set of input pictures to produce a LEGO model [KTM16].

Another class of methods studies how to optimize the physical properties of LEGO models. [HWS+16] proposed a method to adjust the centroid of the model by changing the distribution of the pieces within the volume. More complex methods can be used to study stability [WW12] or to optimize the design with respect to specific physical constraints [LYH+15] (see Figure 2.23, bottom).

Other methods use the zometool system to approximate polygonal meshes. Zometool is a modeling system which uses a set of different struts (3 different types each coming in 3 different lengths) that are interconnected at nodes (there is one single type of node that has 62 possible attaching directions). The approach

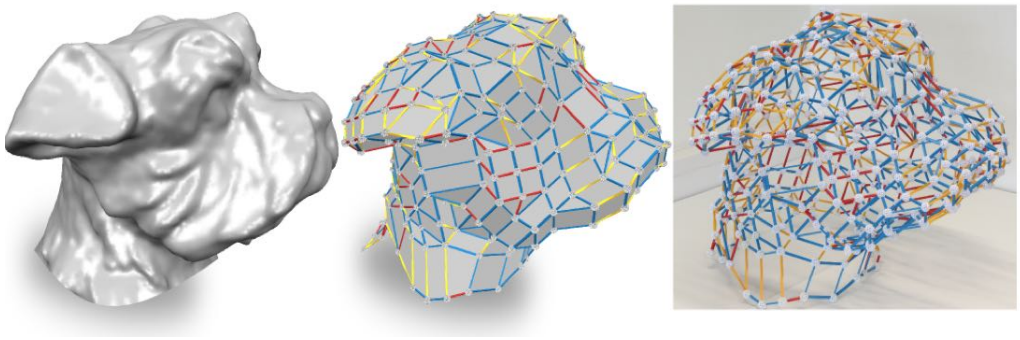


Figure 2.24: Design of a zometool obtained through the method proposed in [ZK14].

(a) (b) (c) (d)

Figure 2.25: (a) Ring parabolic cyclid and (b) Sievert surface, Intitute Henri Poincaré, (c) Clebsch diagonal surface, and (d) String model representing a ruled surface with two real double lines and four real pinch points on each line, Groeningen Museum

proposed by [ZLAK14] finds an initial zometool approximation using a voxelization of the input mesh, then a sequence of local operations is used to optimize the final result. A different approach based on advancing front is proposed by [ZK14].

2.3.6 Printing the Unprintable

Actual physical shapes allow a much more natural and direct comprehension of the shape and properties of complex 3D structures and this has been exploited for didactical purpose since last centuries. In mathematics there is a long standing tradition in using actual physical models, handcrafted using mixed techniques, to illustrate abstract concepts and geometrical structures; the Groningen Mathematical museum [Sch03] and the Institute Henri Poincaré [PB07] in Paris have large collections of mathematical objects that, using gypsum, wireframes and thread nets, are able to present complex surfaces and their properties in an accessible format. Figure 2.25 shows some of these old models from the two cited collections.

3D printing techniques have started to be used for this purpose [KS13, Gür15], but the possibilities are still largely unexplored. It has to be noted that in the classical models various solutions were adopted to portray specific mathematical

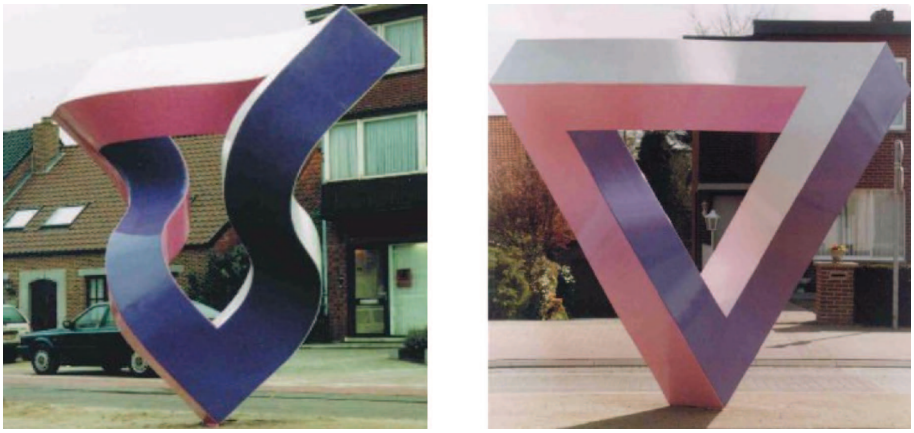


Figure 2.26: A sculpture depicting a real 3D Penrose Triangle (Mathieu Hamaekers, Ophoven, Belgium, 1997)

characteristic and properties of the surfaces, such as parametric isolines, straight lines embedded in the surface, singularities, etc. The task of generating printable shapes that automatically collect and mark such characteristics has been started to be faced in [SS10b], where the authors present a system for exploring the construction of mathematically and aesthetically interesting surfaces. Strictly connected with the use for didactical purposes are the exploitations of the connections between art and mathematical structures for artistic purposes [VC+10], where digital fabrication technologies have been used to bring to life abstract, mathematically-defined shapes that results appealing from both the aesthetic and the mathematical point of view.

Finally the possibility of actually printing shapes deemed as geometrically impossible is yet another possibility. Prominent examples of such objects are the drawings of Escher that, intuitively, do not allow a direct physical actualization. However such figures can be built as tangible 3D objects by imposing constraints on the view point like the actual creation of a Penrose triangle depicted in the sculpture of Figure 2.26. The creation of such structures that, from at least one specific viewing direction, seems in contradiction with the presented scene, has been initially faced by Elber [Elb11]. In this work he identified a class of illusions that can be realized and modeled in 3D and presented a system package that allows end-users to define such seemingly impossible object in two stages: modeling a regular 3D model and then converting it into a seemingly impossible shape using special deformations.

Wu et al [WFY+10] further explored the problem proposing an algorithm that, given a set of 3D locally possible parts of a seemingly impossible figure, automatically optimizes a view-dependent 3D model, subject to the necessary 3D constraints for rendering the impossible figure from the desired novel viewpoint.

Chapter 3

Field-aligned Mesh Joinery

This chapter is based on our publication [CPMS14], in collaboration with Paolo Cignoni, Nico Pietroni and Roberto Scopigno. For this project I have worked on the development of all steps of the following technique (in particular on the polylines sampling strategy, the formaliziation of relations between slices and the sorting of an assembling sequence).

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 [MSM11, HBA12, SP12]) we oriented the slices according to a given cross-field defined on the surface. As most of the recent quadrangulation papers have shown [RLL+06, KNP07, BZK09, PTSZ11, BLP+12], 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,

Figure 3.1: 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.

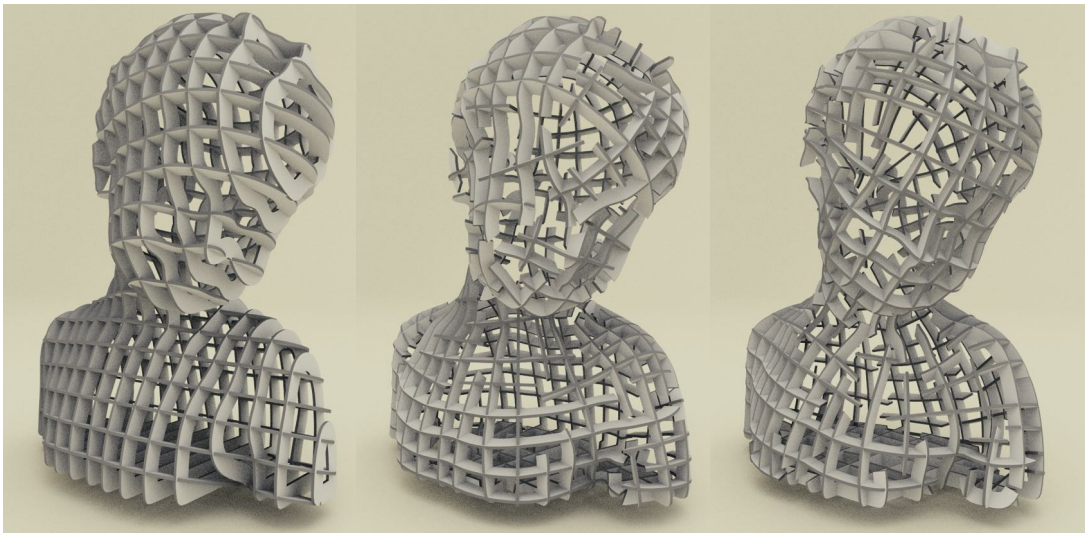
consequently, to efficiently approximate the global structure that characterizes the input shape. Additionally, we ensure a sufficient degree of physical stability of the final structure along with the sequence of manual operations required for the assembly procedure.

Our approach provides limited but low-cost solutions due to the simple cutting technologies employed and the relatively 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, such as in artistic or illustrative contexts, in puzzles or toys, and where assembly is a key part of user experience.

Motivation Rapid prototyping [DSdB06] 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 [MS04b, STL06, MGE07, MI07, LSH+10]. 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.

As for the mentioned works, in our approach, given the low cost of the technology involved, we can create approximation of large objects with a cost that is from one to two orders of magnitude less than a full object reproduction. This is especially true if we consider that the cost of producing structures like the one shown in Figure 3.1, scales linearly with increasing size. For this reason the proposed technique is particularly efficient for producing medium/large structures approximating an input 3D model.

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



(a) (b) (c)

Figure 3.2: (a) The classical waffle approach modeling technique (with axis-aligned slices); (b) our method applied to a cross-field calculated with [BZK09]; (c) field symmetrization techniques [PLPZ12] increase the visual appeal of the final result. The total length of the polylines for each method is approximately the same.

with each other using an extended slit mechanism. Specifically, our contribution for this method are as follows.

- We propose a novel strategy to fabricate illustrative shape approximations based on ribbon-shaped planar slices. Compared to classical planar sections [HBA12], ribbon-shaped slices reduce the physical constraints involved in the assembling procedure, allowing for more complex structures.

- We extend the classical slit mechanism [HBA12] by providing additional structural degrees of freedom. In particular, we consider insertion movements that are not orthogonal to slices. In addition, we formulate nonorthogonal slice placement [MSM11, SP12] in a novel, structurally sound perspective. We demonstrate 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 [HZ00, BZK09, RVAL09]). 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 [PLPZ12]

(see Figure 3.2) 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 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.

3.1 Related Work

Fabricating tangible models from a digital 3D shape is fundamental in many industrial production processes. The majority of current applications requires 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.

As highlighted in Chapter 2 there are currently two trends of research for digital fabrication: accurate and illustrative. This work is mainly related to the latter (Section 2.3) and closely related results are presented in Section 2.3.1.

To recap, McCrae et al. [MSM11] create shape abstractions arranging planar slices to optimize the perception of the original object. This method allows non orthogonal 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. [HBA12] 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 [SP12] allows nonorthogonal slices, but their method tries to retain the simplicity of orthogonally intersecting pieces. Recently Schwartzburg and Pauly [SP13] 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, they 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. Compared to the

method in [SP13], our approach also avoids the complexity of considering multiple planes interlocks inside the object volume by using ribbon-shaped slices and, as a byproduct, the final assembly is significantly simplified.

3.2 Overview of the Complete Pipeline

Our fabrication pipeline, as shown in Figure 3.3, has the following steps.

1. As input, we get a triangle mesh with a cross-field defined on its surface (see Figure 3.3a). We obtained the cross-field using the method proposed in [BZK09] with the symmetrization of [PLPZ12].

2. We sample a set of planar polylines that lies on the original surface (see Figure 3.3b). 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 3.4.

3. The polylines are transformed into a set of ribbon-shaped slices (see Figure 3.3c). These profiles are obtained through a sequence of boolean operations performed in a 2D space (using ClipperLib [Joh13]).

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 3.3d). This step is detailed in Section 3.5.

5. Each slice is then converted to a vectorial representation and organized into sheets ready for automatic laser cutting (see Figure 3.3e).

6. Finally the slices are assembled by following the sequence specified by our system (see Figure 3.3f). The derivation of the assembling sequence is detailed in Section 3.6.

3.3 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 [SÍ12].

For the sake of simplicity, consider the simple situation of two perpendicular slices fitting together (see Figure 3.4). 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

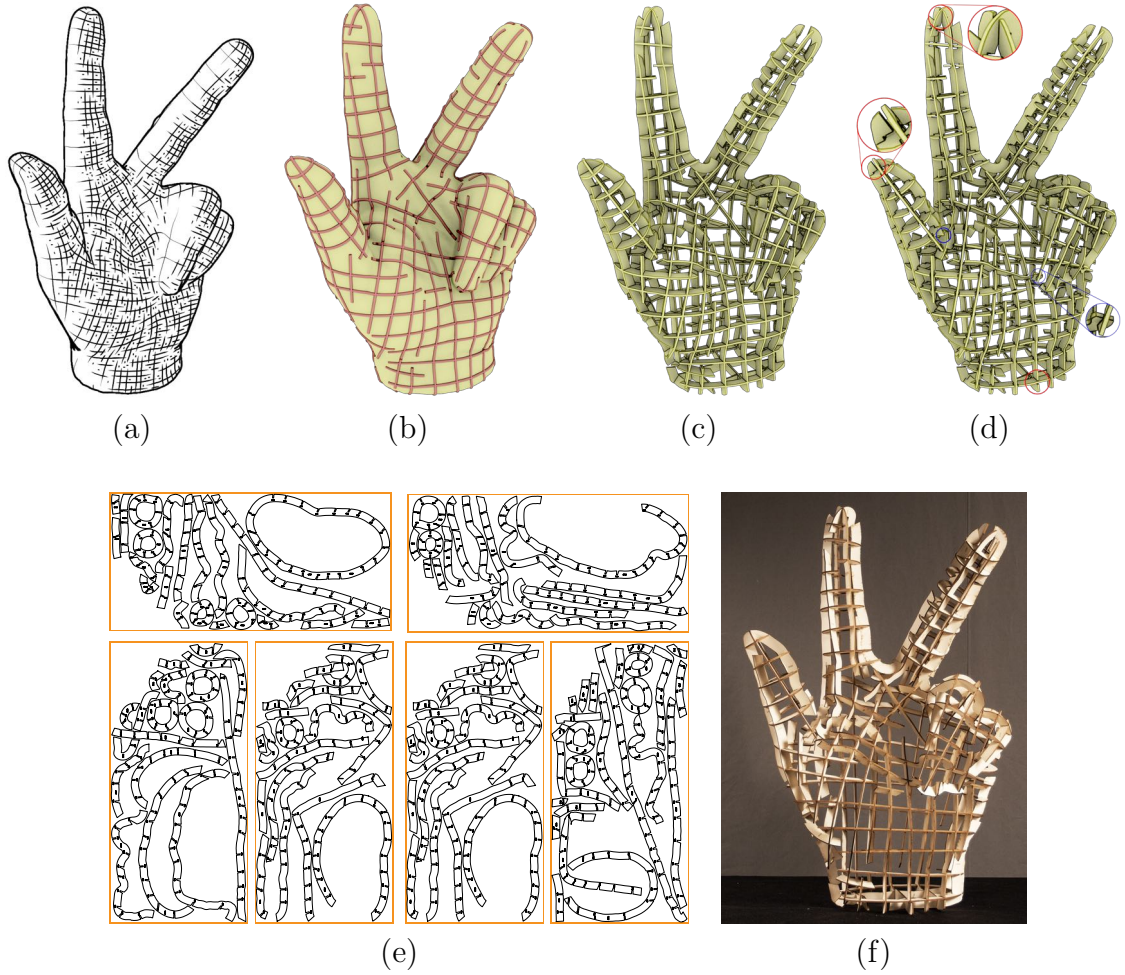


Figure 3.3: A complete overview of our fabrication pipeline: (a) we get as input a triangle mesh and an associated smooth (possibly symmetric) cross-field; (b) we sample a set of well-distributed field-oriented planar polylines; (c) the polylines are transformed into ribbon-shaped slices; (d) the slice 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.

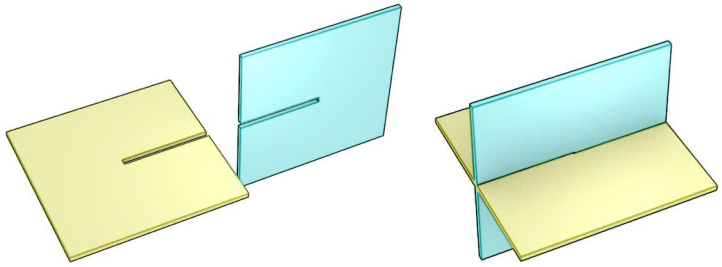


Figure 3.4: The classical situation of two connected slices: for each piece we create a rectangular slit in correspondence with the intersection line.

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 to which this method can be applied.
To overcome this problem (instead of increasing the sampling rate) we explicitly relax these two constraints.

3.3.1 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 λ:

$$\lambda = (|\tan(\pi/2 - \alpha)| + 1) \cdot \tau \tag{3.1}$$

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

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 3.5) or, alternatively, by four slices interlocked together with nonparallel intersections (see Figure 3.7). 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 [MS00]. Any spatial linkage formed by four links and four hinged joints, when in general position, is a highly constrained (rigid) mechanical system. Section 3.4 outlines how we exploit this mechanism to ensure stability in the final structure.

3.3.2 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 3.6, 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 the

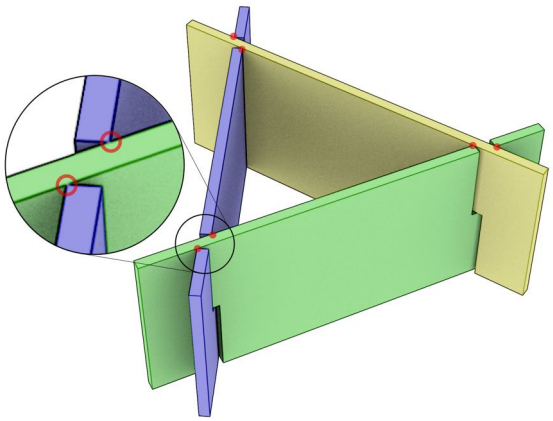


Figure 3.5: Three interlocked slices are rigid and tightly connected, although the slices are not orthogonal and the wide slits are not tightly fitted onto the surface of the other slice. The red dots denote where the slices are pressed/forced against each other, such that the resulting friction ensures the stability of the structure.

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 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 3.6, 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 3.7 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 $s_1, ...s_n$, 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:

$$\Lambda(s) = \min_i(\max_{j \neq i} \textsc{Angle}(\ell_i, \ell_j)) \tag{3.2}$$

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 3.6, the divergence of the green slice is the angle indicated in the second row of the right part of the figure.

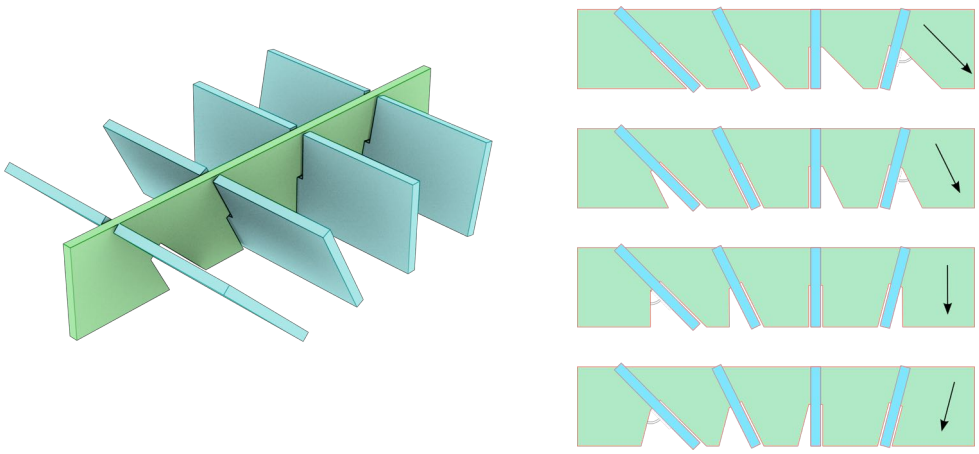


Figure 3.6: The shape of the slit widening depends on the insertion direction. The divergence of the green slice is the maximum angle between the various intersection segments when the best insertion direction is chosen. On the right we show how the slit widening varies when different insert directions are chosen.

3.3.3 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 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.

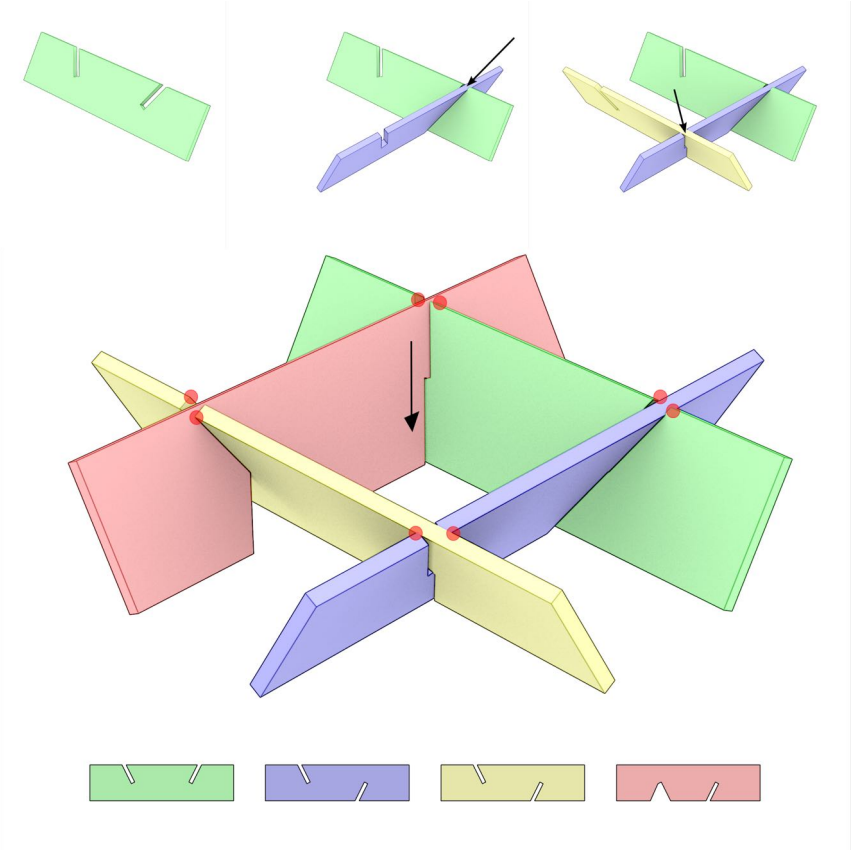


Figure 3.7: Four interlocked slices that are rigidly and tightly connected, even though the slices are neither orthogonal nor inserted along a direction parallel to the intersections. Starting from the green slice, the blue and yellow slices are inserted one by one onto the previous slice along the intersection 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.

3.3.4 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 [HBA12]). 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 [HBA12]: we remove one intersecting slice by splitting the ribbon that has the smallest area.

3.4 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 the chapter introduction 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 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 3.3.1, stability can be ensured locally by the orthogonality of the slices or, globally, by mutual interlocking.

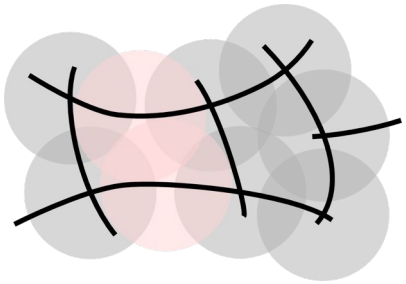


Figure 3.8: The constraint used to guarantee an even distribution of the traces. Gray disks represent intersection distances, while the red disks show the distances between points that are far from the intersections.

3.4.1 Alignment to Cross-Field

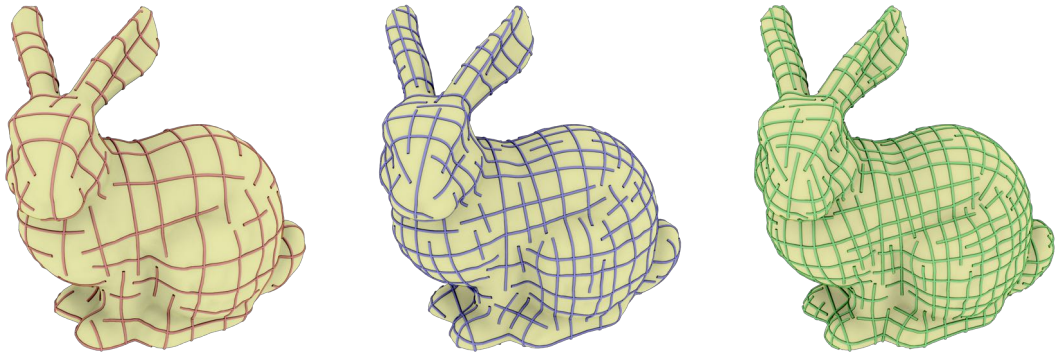
We designed a simple procedure to trace field-aligned planar polylines. For each face and for each direction, we iteratively trace a polyline, called a *separatrix*, which follows the orientation of the field. Since the cross-field is invariant to 90° rotations, at each tracing step the separatrix follows 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.

3.4.2 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 3.8.



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

Figure 3.9: 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.

Figure 3.9 shows a mesh sampled at different radius resolutions. Obviously the higher resolution (small values of r) increases the details of the final model.

3.4.3 Stability Constraints

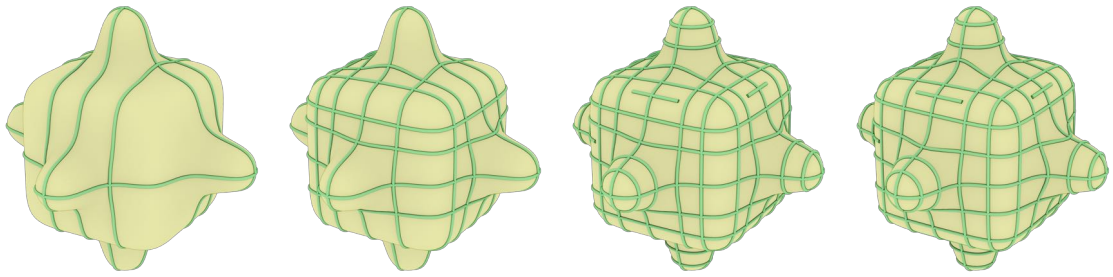
In order to keep the final structure stable, the slice arrangement must be a single connected component. Moreover, the slices should be almost orthogonal to each other. Indeed, orthogonality provides a good grip for the interlocking mechanism, by minimizing the slit widening.
We consider a slice *stable* 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 $[\pi/2 - \delta, \pi/2 + \delta]$;
- or the slice is interlocked in a rigid substructure (see Section 3.3.1, following the intuition of the triangular configuration in Figure 3.5).

3.4.4 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



(a) (b) (c) (d)

Figure 3.10: 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).

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

3.4.5 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 3.10.

3.5 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 3.11 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*.

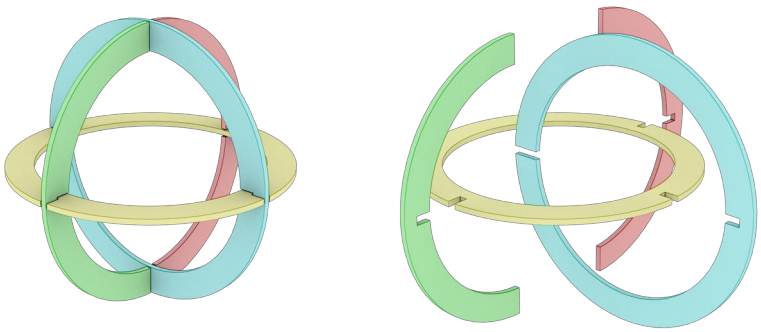


Figure 3.11: Three interlocked looping ribbons must be split into four pieces so that they can be untangled.

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 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 3.3 and, in order to fulfill them, we use the following two-step procedure which:

- 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.

3.5.1 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.

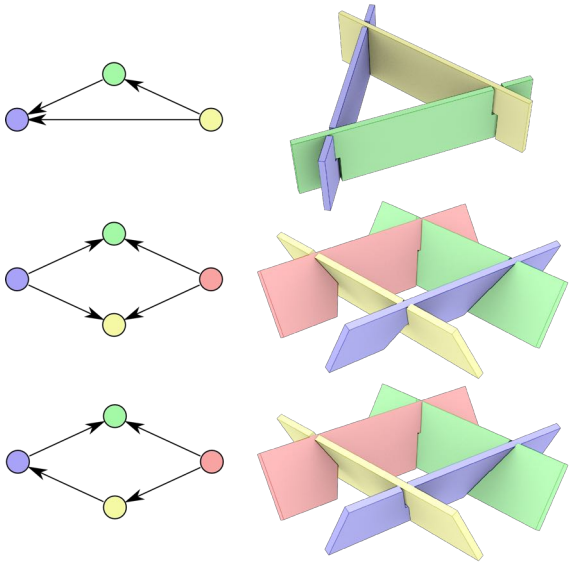


Figure 3.12: The two slice graphs corresponding to the slice arrangements shown in Figures 3.5 and 3.7. The last two rows show two different arc orientations for the same slice arrangement: the slit widenings are affected by the orientation.

Three simple examples of slice graphs with the corresponding slice arrangements are shown in Figure 3.12.

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 3.3 and shown in the last two rows of Figure 3.12 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.

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 of nodes 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

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

Table 3.1: 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.

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 msecs of computing time) are sufficient to get a stable sink set.

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 3.1, 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).

3.5.2 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. 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* [BHT00] (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:

$$h(G) = \min_{0<|G|\leq \frac{n}{2}} \frac{|\partial(U)|}{|U|}, \tag{3.3}$$

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.

3.5.3 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 3.1). This operation may split a slice into two separate components or, if the ribbon is a loop, it may open it.

3.5.4 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 ℓ 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 $Split(s_1, \ell), Split(s_2, \ell)$. 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 3.3d the two blue circles highlight the ribbons that were processed for removal of improper intersections. Figure 3.13 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.

3.5.5 Removing Double Intersections

There are two main reasons for splitting a ribbon:

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

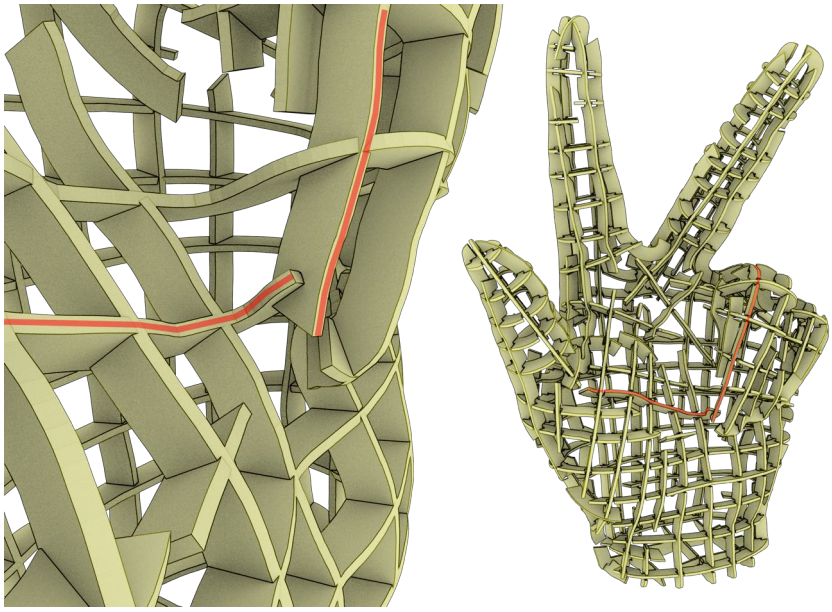


Figure 3.13: A close-up of an improper intersection in the Hand model. The two ribbons marked in red have an intersection that does not touch the original surface

First, we remove all the double intersections, that is, pairs of slices s_i, s_j whose intersection is not a single segment ℓ, but it is composed of two (or more) segments. A typical situation is depicted in Figure 3.11.

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 3.14 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 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

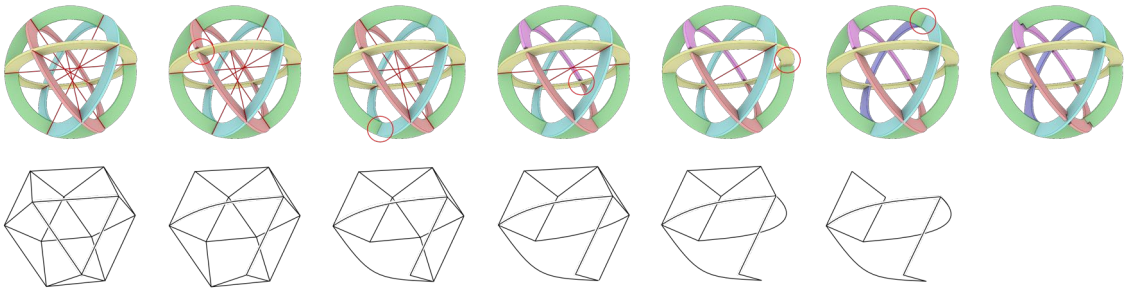


Figure 3.14: An arrangement containing multiple double intersections (indicated by red lines) is corrected by means of repeated split operations (indicated with red circles). In the bottom row we show the intersection graph at each step of the process. The top-right image shows the arrangement when all the remaining six intersections are transformed into slit mechanisms.

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 3.3.1: each slice is involved in a four-cycle of non parallel intersections).

3.5.6 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 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 3.14 there is a slice with a high divergence which causes slit widening. 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 3.3.1.

In Figure 3.3d 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).

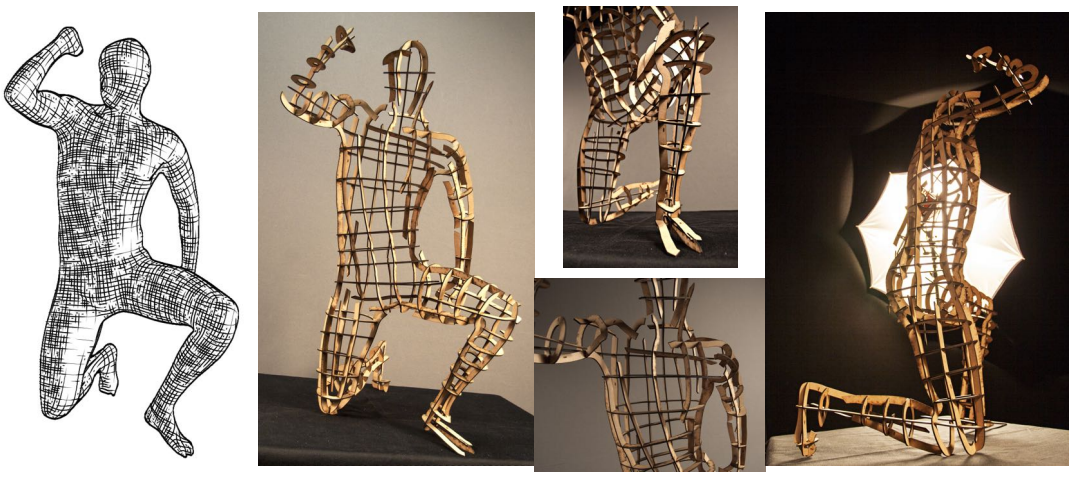


Figure 3.15: The Kneeling Human model. The model is composed of 140 slices.

3.6 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 3.5.1 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 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.

3.7 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 work 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 3.16.

We successfully applied the entire pipeline described in Section 3.2 to approximate input geometries with an associated feature-aligned cross-field as input. The optimized placement of the polylines over the surface of the input mesh is the most computationally intensive step. Using unoptimized C++ code it took from 20 to 60 minutes to compute a regular, field-aligned, arrangement of polylines on the input surface, with time depending on the desired resolution. The other steps of the pipeline, instead, were completed in about one minute. Some examples of the resulting fabricated structures are shown in Figures 3.15, 3.17 and 3.18. 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 [HBA12] (this comparison is shown in Figure 3.2). In addition, the cross-field can be further optimized in a preprocessing step to increase the quality of the results (see Figure 3.2).

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.

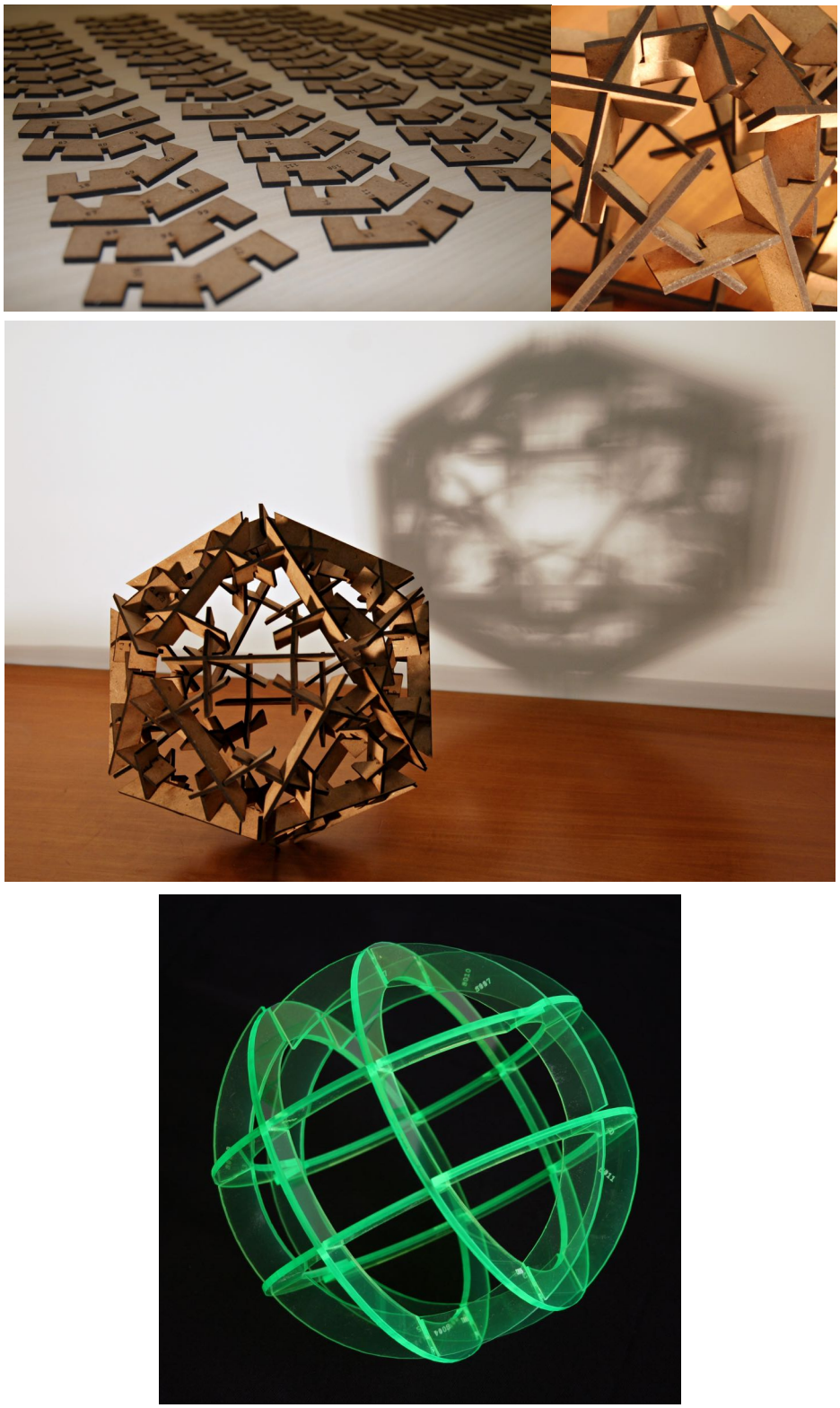


Figure 3.16: A simpler slice arrangement (rather than following a cross-field) has been tested to assemble an icosahedron and a sphere (which has been built using plexyglass).

Figure 3.17: Our algorithm applied the Hand model. The arrangement is composed of 122 pieces.

Figure 3.18: Our algorithm applied to the Bimba model. The arrangement is composed of 178 pieces.

Chapter 4

FlexMolds: Automatic Design of Flexible Shells for Molding

This chapter is based on our publication [MPBC16], in collaboration with Nico Pietroni, Bernd Bickel and Paolo Cignoni. For this work my contribution covered all the steps of the pipeline, including the development of the idea and the implementation of all the computational techniques involved.

Mold casting is a widely used manufacturing process for rapidly producing shapes. It allows the efficient shaping of a wide range of materials, such as resins, and scales well with the number of required copies. However, reusable molds come with the requirement of being detachable from the actual molded part without physically destructing the mold or the molded part. This poses hard constraints on reproducible shapes. For rigid molds, the cavity of a mold piece must resemble a height field. This results in a trade-off between reproducible shape complexity and the number of required mold pieces. Each mold piece should be designed by keeping in mind the complete movement that is needed to separate it from the cast object. Eventually, shapes with high genus or very complex geometries may require to be split into multiple pieces, which can be individually cast and assembled afterwards. In general, automatic mold design is a very complex task since it must satisfy multiple non-local constraints.

A different strategy is silicone mold casting [BOP10]. An object is surrounded by silicone, and after curing, the silicone is cut out and detached from the original object, obtaining a flexible mold. This allows the creation of stunning replicas of objects with highly complex geometry. However, so far this has been a labor-intensive manual process, requiring a positive copy of the shape, performing a silicone cast, and cutting the silicone mold. Furthermore, in practice, deciding and performing these manual cuts is challenging, especially in the presence of complex shapes and topologies.

Inspired by this approach, we propose a novel reproduction approach for highly detailed free-form shapes, based on *flexible mold shells* (FlexMolds). FlexMolds are made of a thin but still sufficiently shape-preserving shell of deformable

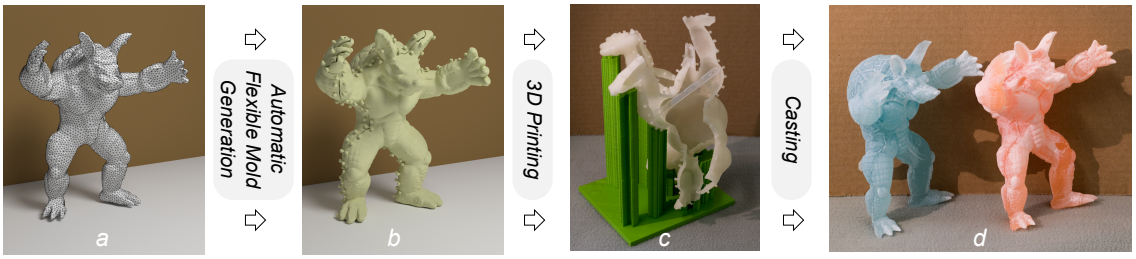


Figure 4.1: Starting from a 3D model (a), we automatically generate a set of cuts over its surface that allow the generation of a flexible mold shell (b) that can be 3D printed (c) and used for casting multiple physical copies (d) of the original model.

material, which can be printed by using a commercial 3D printing device. Thanks to their flexibility, they provide more freedom during the removal process, allowing the reproduction of complex shapes, often even with just a single mold piece. Our fabrication method inherits the main advantages of reusable mold casting manufacturing techniques: the mold shell can be reused for multiple copies, the fabrication process is fast and cheap compared with 3D printing, there is a wide range of manufacturing materials and colors available (see Figure 4.1).

FlexMolds are designed with a sufficient number of cuts that allows the extraction of the internal object once the cast liquid is solidified, without damaging the mold. Cut placement is a complex non-local problem. A proper cut layout should be designed not only to guarantee local detachment, but also to allow the mold to completely depart from the cast object. This may include complex, global, large-scale deformations that allow the flexible mold to adapt to geometric constraint during the extraction (see Figure 4.2). To guarantee the existence of a feasible extraction sequence we simulated the entire extraction process. Specifically, we are interested in ensuring the existence of a feasible extraction sequence that keeps the deformation within a certain range. Eventually, the simulated extraction sequence may be different from the one performed by the user.

Ideally, we would like to use as few and as short cuts as possible. However, optimizing cut design is not a well-posed problem, i.e., the extraction sequence does not change continuously with respect to the cut. Figures 4.2c and 4.2d show that a small change in the cut topology may cause an abrupt change in the extraction sequence.

4.1 Related Work

In recent years, the computer graphics community has contributed with many shape-processing approaches and design tools for computing physically realizable objects. While a comprehensive overview can be found in [LWSW14, UBM15], the following sections will focus on setting our work in context with closely related work.

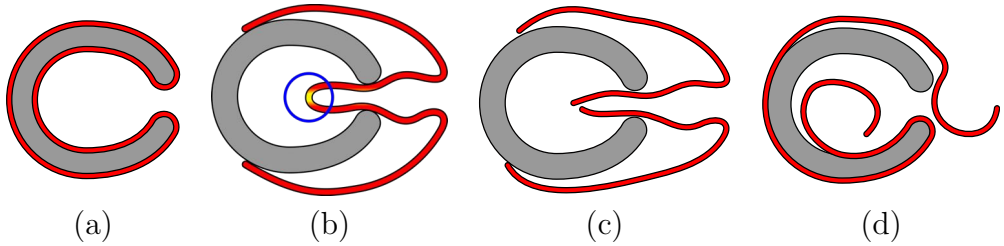


Figure 4.2: A simple 2D example: (a) the model to be cast; (b) thanks to their flexibility, FlexMolds allow a simple removal process; however, the stress can be concentrated on a point (blue circle); (c) additional cuts can be added to reduce the stress induced by the removal process; (d) the same effect may be obtained by accurately placing the cut in a different position.

Mold design Molding is a widely used manufacturing process for rapidly producing shapes. While ancient molds, for example, for creating clay figurines or spear heads, are proof that the technique has existed for thousands of years, mold design is still a highly challenging and important topic in engineering. Expendable mold casting allows the reproduction of shapes of similar complexity as 3D printing with a wide range of materials beyond readily available 3D printing materials [Sin10] and is used, for example, for wax casting of digitally designed jewelry [Wan11]. In our work we focus on non-expendable mold casting, i.e., molds that can be reused for multiple production cycles. An introduction to manufacturing processes can be found in [DGBK11].

Reusable molds are usually assumed to be made out of a rigid material, such as metal, which restricts the complexity of shapes that can be manufactured as the shape needs to be removable from the mold without damaging them. Therefore, the parting direction and parting surfaces of molds play an important role. Chakraborty and Reddy [CR09] proposed a joint optimization to minimize the area of undercuts, the flatness of the parting surface, and minimized draw depth for determining the best parting direction, parting line, and surface for a two-piece molded component. Zhang et al. [ZZL10] performed a geometric analysis of the shape to identify potential undercuts with their possible withdrawal directions; this analysis provides decision support information for designers choosing parting direction, parting lines, and surfaces. For more complex models, Lin and Quang [LQ14] proposed a heuristic to segment the surface of an object and compute feasible parting directions for automatically generating multi-piece molds, and [HLZCO14] explored the decomposition of shapes into pyramidal pieces. To address the problem that complex geometries might require an extensive number of mold parts, Herholz et al. [HMA15] proposed a method for approximating free-form geometry with height fields in which they deform the shape slightly in order to meet the height field constraints needed for rigid casting. In our work, by exploiting the properties of flexible molds, we follow a very different strategy that

does not require any changes to the original geometry and is able to manage objects with complex geometry and topology. Traditionally, flexible molds are made by surrounding an object with silicone and then, after curing, iteratively cutting the silicone until the object can be removed. While this provides the potential to reproduce highly complex shapes, so far the design of such molds relies on a manual process and well-trained users [BOP10]. Furthermore, it is a two-step process requiring a positive of the shape before the mold can be manufactured. In contrast, our method is fully automatic and allows the mold to be directly 3D printed.

Simulation and motion planning A variety of methods have been presented for simulating deformable objects and shells based, for example, on nonlinear finite elements [SB12], discrete elastic shells [GHDS03], mass-spring systems, co-rotated linear finite elements [MSJT08], or position-based dynamics [MHHR07, BMOT13]. The chosen method is usually the result of a careful balance between required accuracy, speed, and robustness.

Solving the inverse problem, i.e., what forces or displacements are required to achieve a desired goal position, is highly challenging. Methods were proposed for motion planning for robotic manipulation and the assembly planning of deformable objects, such as ropes and wires [SI06], sheets, and elastic volumes [Jim12]. These strategies are often defined for relatively simple polyhedral environments and do not extend directly to our case, in which we are facing the removal process of a tightly fitting elastic shell from an object with complex geometric features and topology.

While in general important, we consider the simulation of the flow of material during the casting process [GPT07] out of the scope of this work and restrict ourselves to low-viscous resins.

Shape decomposition and approximation for fabrication In the broader context, numerous algorithms for shape decomposition and fabrication have been proposed in computer graphics. For example, recent work suggested strategies for partitioning models into smaller parts, relevant for packing large objects into 3D print volumes [LBRM12, VGB+14b, CZL+15, YCL+15].

To approximate the surface with a small number of planar polygonal primitives, Chen et al. [CSaLM13] iteratively assigned mesh faces to planar segments. This results in a closed intersection-free mesh, which can be augmented with internal connectors and fabricated. Relaxing the constraint to developable patches, D-Charts [JKS05] provide a quasi-developable mesh segmentation, while Tang et al. [TBWP16] and Solomon et al. [SVWG12] provided interactive modeling systems for developable surfaces that can be used for model fabrication from sheets of material.

Starting from a surface model, Hildebrand et al. [HBA12] and Schwartzburg and Pauly [SP13] computed interlocking planar elements that resemble an

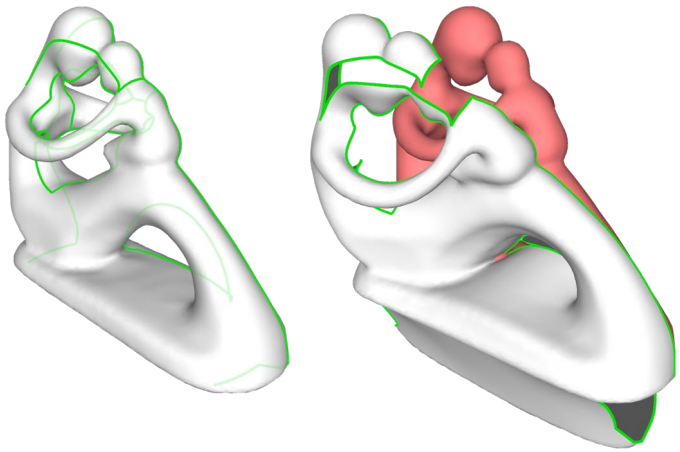


Figure 4.3: A cut layout X (in green) characterizes a thin flexible mold M (white), determining how it can be opened and detached from the object surface S (in red).

illustrative approximation of the desired shape. Also in the method of Chapter 3 we extended this concept and placed the elements driven by a feature-aligned input cross-field [RVAL09, BZK09]. Inspired by this concept, we compute potential candidates for cuts along smooth polylines that emerge from a field-aligned patch layout decomposition of the initial mesh [TPP+11, CBK12, MPZ14, RRP15].

4.2 Designing FlexMolds

Given an object represented by a manifold, watertight, triangulated surface S, we target the problem of the automatic design of a flexible thin mold M that can be used for liquid casting. With respect to the more commonly faced problem of rigid casting, allowing the deformation of the mold makes it possible to remove more complex *non-height-field* shapes without requiring intricate fine-grained mold decomposition or the modification of the original object shape. Our flexible molds are usually very thin (in the order of 2–3 mm) and can be composed of a few pieces or even a single shell that can be unwrapped/detached from the cast object, exploiting the elasticity of the mold material.

Under these assumptions, the main characterizing feature of a mold for a surface S is the cut layout X over S that determines how the mold M_X (which is the mold M, equipped with the cut X) decomposes and opens into one or more patches p_i. Figure 4.3 illustrates these concepts, showing a cut layout X (in green) of a flexible mold M (white) and shows how X rules how M can be opened and detached from the surface S of the cast object (in red). The other geometric characteristics of the mold (e.g., its thickness and the actual 3D profile of the cut) can be inferred starting from X and S and are discussed in Section 4.3.

To understand if a cut layout X can generate a feasible mold M_X, we test if M_X

can be successfully removed from the cast surface S without suffering high strains. This requires solving the following two problems:

Extraction movement Given a flexible mold M wrapped around an arbitrarily complex object S, we want to find a temporal force field $F_M(t)$ that, once applied to M, detaches it from S in a finite time and in a sound way. We postulate that the complete solution to this problem, given the similarity with the computational hardness of motion planning [HA92], probably lies in the realm of intractable problems, and we propose a practical heuristic for that (Section 4.2.1).

Strain evaluation Assuming we know how to drive the removal of a mold, we want to evaluate how this process affects the mold itself in terms of the strain suffered during the extraction: we have to ensure that for multiple uses of the same mold, it will not break when the object is extracted (Section 4.2.1).

These two interconnected problems are specific to the case of flexible molds and, to our knowledge, have never been faced before. In this work, we define that a cut layout X is *feasible* if, solving the two problems above, we find that the mold M_X can be removed from S and during this process the strain is below a certain threshold.

One very important characteristic of flexible molds is that their thickness can be considered negligible in many parts of the feasibility evaluation process. For example, if a cut layout is composed of multiple pieces (as the one shown in Figure 4.4, left), we can assume that we can evaluate the feasibility of each piece independently, expecting that, given the small thickness of the mold and the flexibility of the material, the patches do not impede/obstruct each other.

4.2.1 Searching for Good Cut Layouts

As introduced in the previous section, the shape of the cut layout X is the main feature defining a mold.

To obtain a cut layout, we initially tried to decompose the mesh in multiple pieces by using a purely geometric approach that clusters portions of the surface with similar normals [CSAD04] (see Figure 4.4, left). Unfortunately, we observed that this approach, when applied to arbitrarily complex geometry, may decompose the mold into an overly high number of disconnected pieces, making the manual assembly of the mold practically infeasible. Conversely, introducing just the minimal amount of cuts to open the mesh to a topological disk [Dey94] may result in a configuration that does not allow for extraction due to excessive deformation (see Figure 4.4, right). This motivates the need for a method to adapt an initial cut layout, either by removing or introducing more cuts.

In this chapter, we want to derive a cut that, while being sufficiently short and simple to generate a practically feasible mold, it also allows generating a mold that can be detached from the cast object without suffering excessive deformations.

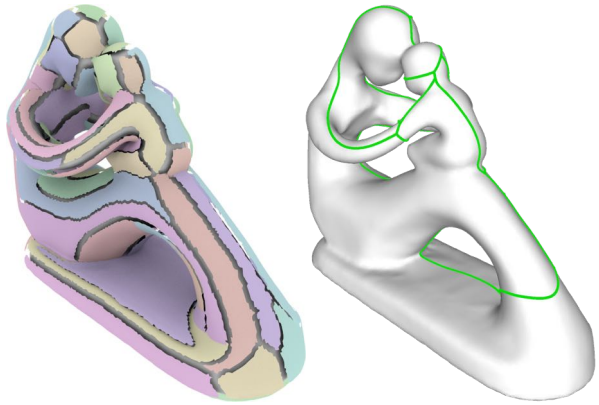


Figure 4.4: Left: a cut layout obtained using the approach in [CSAD04]. Right: a cut layout that opens the mold to a disk is not sufficient to ensure its full removal because of the high deformation induced by the extraction process.

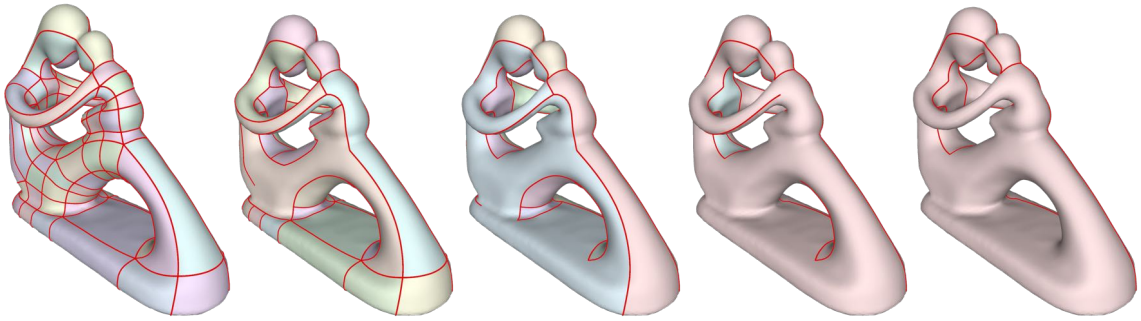


Figure 4.5: The bottom-up greedy optimization process. Starting from a dense cut layout (left), generated by a patch decomposition, we iteratively perform operations that remove segments of the initial cut layout, choosing at each step the operation that requires the minimal deformation in the extraction process and stopping when this exceeds a given threshold.

One possible way to explore the space of possible cut layouts could be a top-down approach in which cuts are grown over the surface starting from an almost closed mold. This class of approaches has been successfully used in texture parametrization to minimize the distortion [GGH02], but we found it unsuitable to bootstrap our optimization. In fact, with those approaches in the initial steps, when the cut is short, the resulting deformations and forces required to extract the mold can be arbitrarily high. This can be a significant problem for two main reasons: first, large deformations require more accurate and costly simulations, making their use in the inner loop of an optimization process impractical; second, we experimentally found that when a high deformation is allowed, our heuristic extraction strategy may generate implausible removal movements.

For these reasons, we opted for a bottom-up approach that stays in the region

of feasible cut layouts. We start, as illustrated in Figure 4.5, by partitioning the mesh into a set P of small, simple, disk-like patches p_i whose boundaries provide a dense cut layout X_0. For such a *dense* cut layout, it is easy to test feasibility; all patches are checked individually if they are removable with relatively low deformation. Assuming that this initial condition is fulfilled (each patch in the initial patch layout is removable), we consider the initial cut layout induced by P as composed of a set of curved polylines χ_{ij} over S that correspond to the portion of boundary shared by two patches p_i, p_j. The main idea is that we incrementally make X sparse in a greedy way by iteratively removing one polyline χ_{ij} at a time. Intuitively, at the beginning of the process, this means merging the two patches p_i, p_j along a common boundary.

Figure 4.6: Sculpt model. Left to right: the flexible mold, the result of the cast using neutral resin, the cast with gypsum, and the cast with colored resin.

Greedy Optimization

The optimization process is organized as a greedy loop where at each step we remove the best polyline χ from X. This removal operation can involve one or two patches p_k of the mold M_X according to whether before the removal χ connects two different connected components of M_X or not. At each step of the process, we remove the polyline χ such that the resulting mold can still be extracted with minimal deformation.

We keep all the possible removal operations arranged in a heap where the score function is the maximum deformation found during the feasibility evaluation process described in Section 4.2.1. We continue this greedy process until there is no more feasible operation, or in other words, closing the current cut X by removing a polyline $\chi \in X$ would introduce too much strain or lead to a mold for which we are not able to find a successful extraction movement.

Even if we use a very efficient dynamic simulation system for the feasibility evaluation, it is necessary to keep the number of these tests as low as possible. For this reason, we use the two following lazy update heap strategies.

Mailboxing After each extraction operation that involves the patches p_i, p_j, we should re-evaluate the heap score for all the χ lying on the boundary of p_i, p_j. To avoid all these evaluations, we mark these entries as not updated in the heap. When we extract one of these outdated entries from the heap, we re-evaluate it and eventually put it in the heap again. Such an approach is a reasonable heuristic because it is highly probable that such scores will increase. In most cases, two patches are more difficult to remove when merged than individually.

Early abort of evaluation The most time-consuming evaluations are the ones that require many steps and large deformations to extract the mold piece. For this reason, we abort the simulation when the deformation grows above a given threshold. This deformation threshold is dynamically increased during the whole greedy optimization loop and is set to be the 30 percentile of the entries in the heap. In other words, we abort all the evaluations that involve deformations higher than the 30 best values in the heap. We mark these entries in the heap, and when they pop out from the heap we proceed as above, re-evaluating and putting them in the heap again. The rationale is to delay the slow evaluations only when the process is close to the end and they are really needed.

This approach guarantees that a feasible cut layout will always be generated. It is based on two prerequisites: we require an initial feasible patch decomposition and a method to evaluate whether a candidate cut layout X allows the resulting mold to be extracted. It should be noted that, while we cannot guarantee that the approach ends up with a single patch, all the experiments we performed generated a mold composed of a single piece. From a topological point of view, note that there is no need for having a mold that is disk-like; in fact, several of the presented results are molds with multiple holes.

Initial Patch Decomposition

Our method requires an initial patch decomposition in which each element of the generated mold can be easily detached from the object surface. There is a vast literature about obtaining patch decompositions through mesh segmentation and partitioning (see [Sha08]), and many methods can be reasonable candidates for generating it. We evaluated different approaches for this task, and the results are discussed in Section 4.4. For our experiments, we chose the quad layout proposed in [PPM$^+$16] that is generated starting from a cross-field over S, mostly for two intuitive reasons. First, quad patches are simple enough to guarantee feasibility but sufficiently large to not slow down the greedy optimization process. Second, the edges of this class of quad layouts are aligned with a smoothed version of the surface curvature. This characteristic generates curvature-aligned cut layouts, a feature that, intuitively, strongly resembles the natural way of decomposing a mold; see, for example, the long lines in Figure 4.7 that align with the overall shape of the model. These lines constitute a reasonable set of cut candidates to

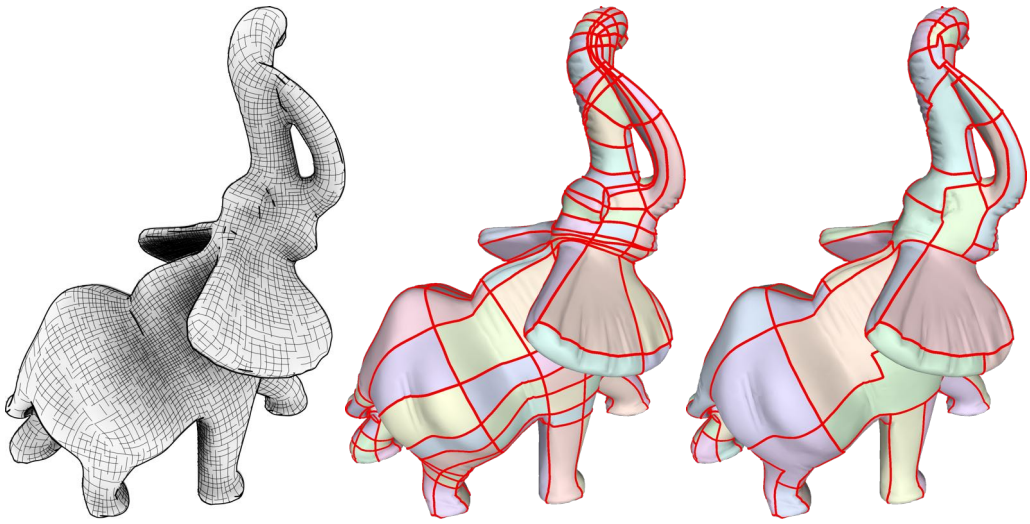


Figure 4.7: An example of an input cross-field (left), the resulting quad layout (center), and the merged patch decomposition (right) used as the starting point for the greedy optimization process.

open the mold without too much deformation. A more detailed comparison between the different decomposition strategies is shown in Section 4.4.

Evaluation of a Patch Layout

Detaching the mold from an object induces deformations. A single, localized excessive stress in one moment of the extraction might already damage the mold. For this reason, the induced stress must be considered at the local level.

This requires a physically-based simulation of the mold and the derivation of the forces imposed by the user to extract the cast rigid object from the deformable mold surrounding it. While a full FEM simulation would be highly accurate, it comes with significant computational cost. For this reason, we opted for a dynamic simulation and a heuristic to compute a force field that drives the mold off the surface.

We based our dynamic simulation on the projective dynamics method [BML$^+$14], which ensures a sufficient degree of accuracy and robustness at relatively low computational costs. We apply this approach on a thin shell model of our mold M_X. The contact between the mold M and the cast object S is simulated using dynamic plane constraints. We uniformly remesh each patch of the initial decomposition to ensure the quality of the triangulation and sufficient degrees of freedom for the physical simulation.

For each step of the extraction simulation, we evaluate the *current* deformation of each triangle. We then record the maximum deformation, defined as the principal stretch of the plane strain, i.e., the maximum singular value obtained from the polar

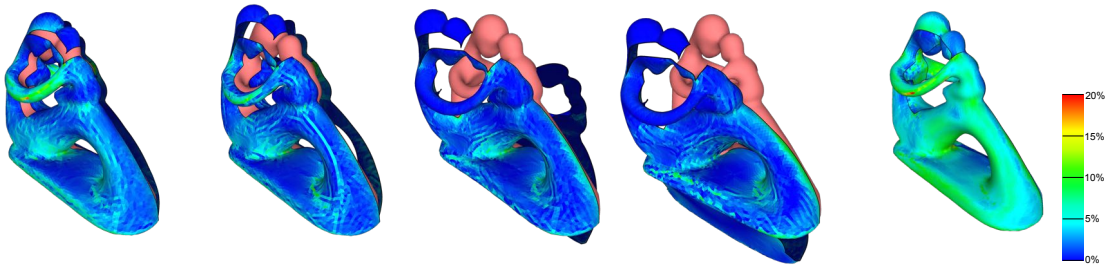


Figure 4.8: The feasibility evaluation process, used in the greedy optimization to score candidate cut layouts X, attempts to extract the mold M_X and records the maximum deformation suffered during this process. Deformation is color coded according to the maximum feasible deformation threshold.

decomposition of the in-plane deformation gradient. Our measure is based on the St. Venant maximum principal strain theory, which assumes that yield occurs when the maximum principal strain reaches the strain corresponding to the yield point during a simple tensile test. However, since we know the deformation field, other measures could be easily employed as well. The goodness of a cut is quantified as being inversely proportional to the maximum deformation among all triangles during all the steps of the extraction process.

Figure 4.8 shows some steps of one of these extraction simulations, where the color indicates, for the intermediate four steps on the left, the current deformation of each triangle. The last figure on the right shows the maximum deformation encountered by each triangle during the entire simulation; color coding of the deformation is done according to the maximum acceptable deformation.

Extraction Path

We use a dynamic system in which the extraction of a mold patch is simulated by applying a force field $F_M(t)$ to each vertex. The force field $F_M(t)$ deforms and drives the extraction from the cast model. It is defined as the sum of two fields: the *detaching forces* $F^D(S)$, a volumetric force field that pushes the mold piece away from the object surface, and the *moving forces* $F^M(P_k, t)$, an evolving field that depends on the actual shape of each mold piece P_k.

The **Detaching Force** field $F(S)$ is a volumetric static force field that is directed along the negative gradient of the distance field computed for the surface S (see Figure 4.9). Its main purpose is to push each piece away from the casted object. The magnitude of this force linearly decreases from a fixed maximum on the surface to zero for all the points with a distance greater than one third of the surface bounding box diagonal. For simple cases, detaching forces may be sufficient to separate the mold from the inner object. Unfortunately, for complex objects the mold can still be stuck in a position that wraps the object, even if "locally detached" from the cast. To overcome this problem, we introduced moving forces.

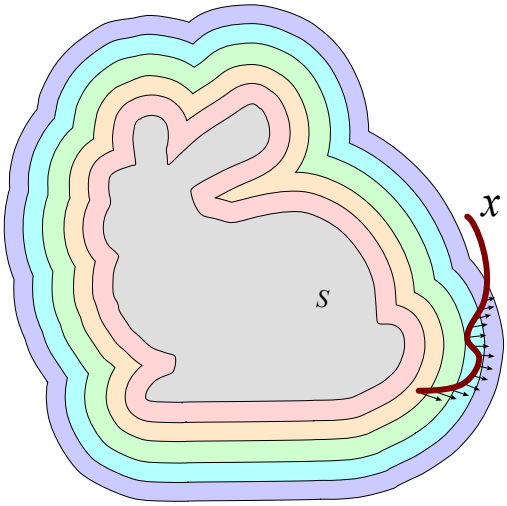


Figure 4.9: An illustrative representation of detaching forces. With increasing distance from the surface the forces linearly decrease to zero.

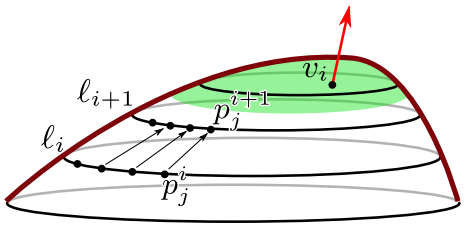


Figure 4.10: Moving Forces: for a given mold piece, in the dynamic simulation, we use the transformation τ matching point pairs p_j^i, p_j^{i+1} lying on the geodesic isolines ℓ_i, ℓ_{i+1} to determine, for each point in the inner (green) part of the patch, the direction of the moving force.

Moving Forces are defined by a dynamic field that depends on current shape and position of the mold: intuitively they grab the mold from the farthest region from the cuts, and pull it away, along a single direction, from the cast. For each step of the simulation, given the deformed patch P_k of the mold M at time t, we compute the isolines ℓ_i at an increasing constant geodesic distance from the border (with a step of 1/20 of the bounding box). For each pair of consecutive lines ℓ_i, ℓ_{i+1}, we regularly sample points, and for each point p_j^i on ℓ_i, we find the corresponding point p_j^{i+1} closest on ℓ_{i+1} along the geodesic gradient direction. We find the rigid transformation τ_k^t that best matches all these point pairs ($p_j^i \rightarrow p_j^{i+1}$ $\forall i, j$), and we use it to compute the moving force direction as follows: we consider the set V_k of the 30 percentile vertices v_i farthest from the border and, for each vertex v_i, we use the displacement vector $\tau_k^t(v_i) - v_i$ as the moving force direction. The intensity of the force grows linearly with the geodesic distance from zero on the boundary of V_k up to a fixed maximum on the farthest point from the

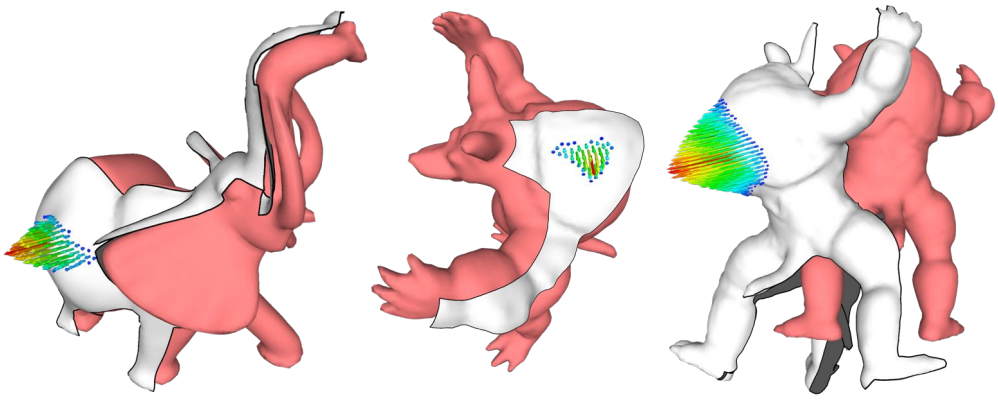


Figure 4.11: Moving Forces: a few patches with the resulting moving force directions; each patch P_k^t is shown, in its deformed state, at time t. Color and length of the arrow represent force intensity. The first two examples on the left show patches that only partially cover the model, while the one on the right covers the whole model.

boundary. Figure 4.10 illustrates this concept and the resulting force direction used to generate the force $F^M(P_k, t)$ applied to one of the points of the set V_k of the inner vertices of the patch (shown in green).

Figure 4.11 shows a few patches in their deformed state during the dynamic simulation and the moving force direction applied on their vertices. For the sake of efficiency, assuming the mold piece will suffer only limited deformations, we compute the isolines at rest shape and we track corresponding point pairs during the simulation; then for each step of the simulation, to compute the new forces, we only have to update the transformation τ.

Both detaching and moving forces are essential to derive the extraction sequence in the general case. While detaching force depends only on the shape of the casted object and drive the mold on a global perspective, moving forces depend on the current deformation of the mold and allows the mold to escape from locally intricate situations. We experimentally observed that omitting the detaching forces significantly reduces the convergence speed of the simulation, while removing the moving forces can prevent the mold extraction, even for simple cases.

The evaluation process for a mold piece P_k starts at its rest position. We then start the simulation by applying detaching and moving forces until the mold is completely extracted from the surface S (the cast object). This condition is tested by checking if the intersection between the mold and the cast object's oriented bounding box is zero. If the number of simulation steps exceeds a given threshold, we mark the current mold as non feasible.

Our strategy is designed to guarantee that the computed extraction sequence can completely remove the cast object from its FlexMold. Our method is capable of leaving additional cuts to lower the deformation due to the extraction process

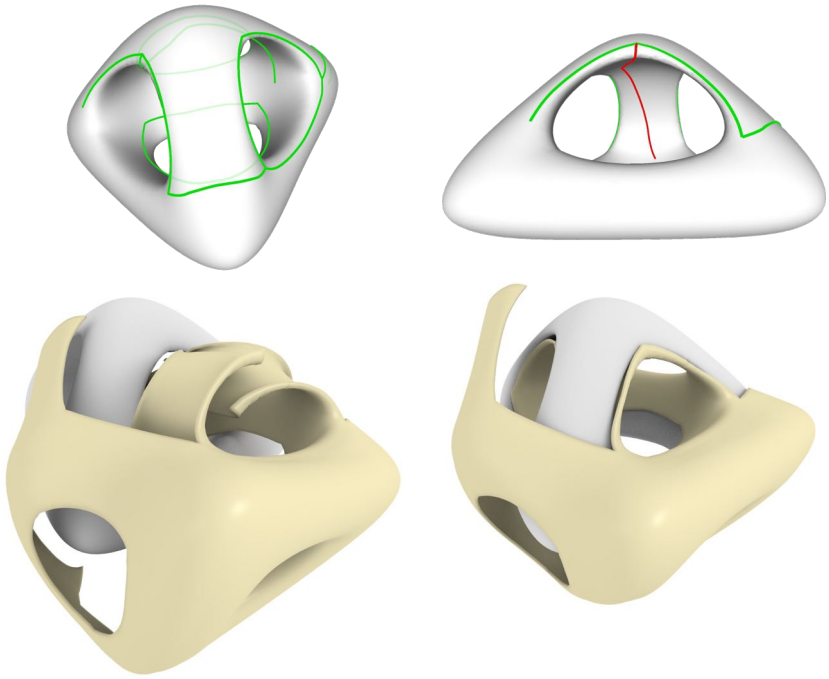


Figure 4.12: Left: an internal cut is used to enable a greater deformation of the internal portion of the shell, allowing for its extraction. Right: when the internal cut (shown in red) is not present, the mold is stuck and cannot be extracted.

or to allow the mold to bend and pass more easily through narrow passages during the extraction. Figure 4.12 shows an example where an internal cut in the model becomes mandatory for extraction. Figure 4.13 further illustrates the importance of small cuts to reduce the stress caused by the extraction process. To achieve this result, the simulation of the entire extraction process is required, including modeling the contact between the FlexMold and the cast object.

4.3 Generating Fabrication-ready FlexMolds

As result of our greedy optimization process, we have a cut layout X defined on the surface S of our starting model. However, to fabricate the final mold M_X, we need to extend it to a 3D solid. Hence, the initial surface is first inflated, then cuts are modeled as solid V-shaped prisms and finally subtracted from the volume by performing boolean operations on the volume.

4.3.1 Placing Air Vents

To ensure proper air escape when pouring the resin inside the mold shell, we have to accurately place small holes in the mold (see Figure 4.14). A bigger hole is used

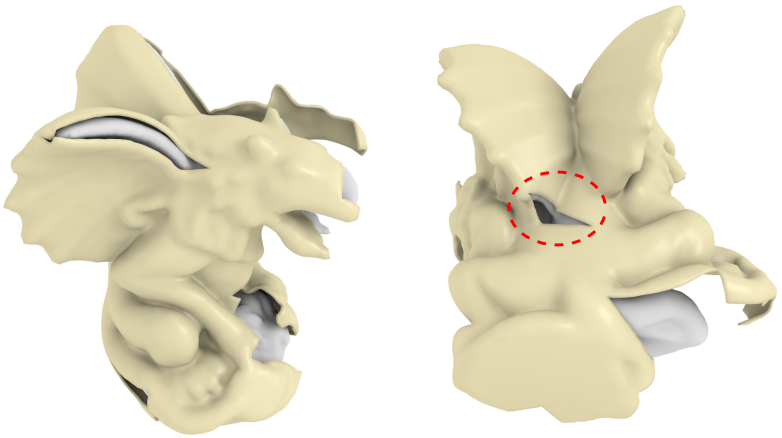


Figure 4.13: Some additional cuts help to reduce the local deformations during the extraction of the mold.

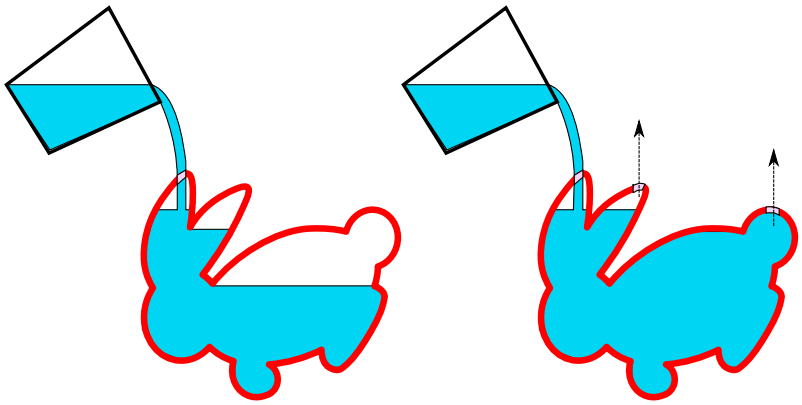


Figure 4.14: Diagram illustrating the bubble trap problem.

to pour the resin inside the mold.

Theoretically, for a given mold orientation, every local maxima along the gravity direction should be equipped with a hole for letting the air escape. We select the best orientation by sampling directions on a sphere [KISS15] and pick the one that minimizes the number of required air vents. Due to geometrical noise and high-frequency details, this approach may still find a large number of maxima; however, some of the holes may be practically unnecessary. If we slightly shake the mold before the resin solidifies, the air may escape through nearby maxima. Formally, we can think of shaking being equivalent to varying the mold's vertical direction by an angle α. Following this intuition, we reduce the number of maxima by using a simple strategy.

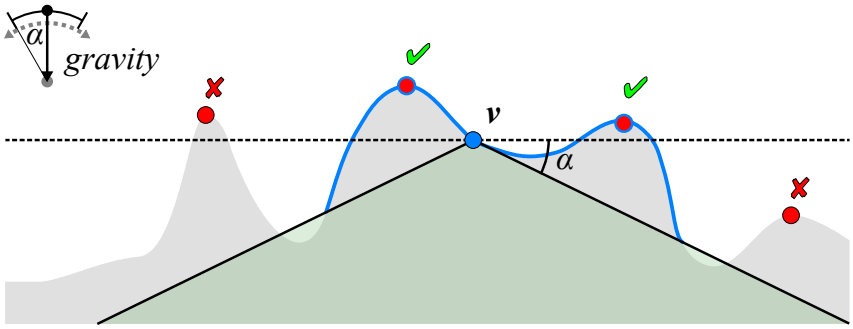


Figure 4.15: Diagram illustrating the shaking cone of a vertex.

For each vertex v, we define a shaking cone with an aperture equal to $\pi - 2\alpha$. We select the portion S_v of the mesh that is above the cone and connected to v and then associate to v all the maxima that are in S_v. Figure 4.15 shows a vertex v, the shaking cone, the region of S associated to v in blue, and the two maxima that can cover that vertex according to the shaking angle α. At the end of this procedure, each vertex is associated to one or more maxima. We reduce the holes by searching for the smallest set of maxima that cover all vertices. We follow a classical greedy heuristic for this minimum set cover problem. We iteratively add one maxima at a time, favoring the ones that cover the larger uncovered portion of the surface, until the entire mesh is covered. The result of this process on the dragon model is shown in Figure 4.16.

4.3.2 Using the Mold

During the casting process, the mold has to be properly sealed. Therefore, we print a sequence of small posts along each side of the cut, tie them together using a thin elastic rubber band (as shown in Figure 4.17), and seal the seams with silicone. While casting, the mold should be kept in the optimized orientation, otherwise the air vents may not work properly. For this purpose, we print a custom lower

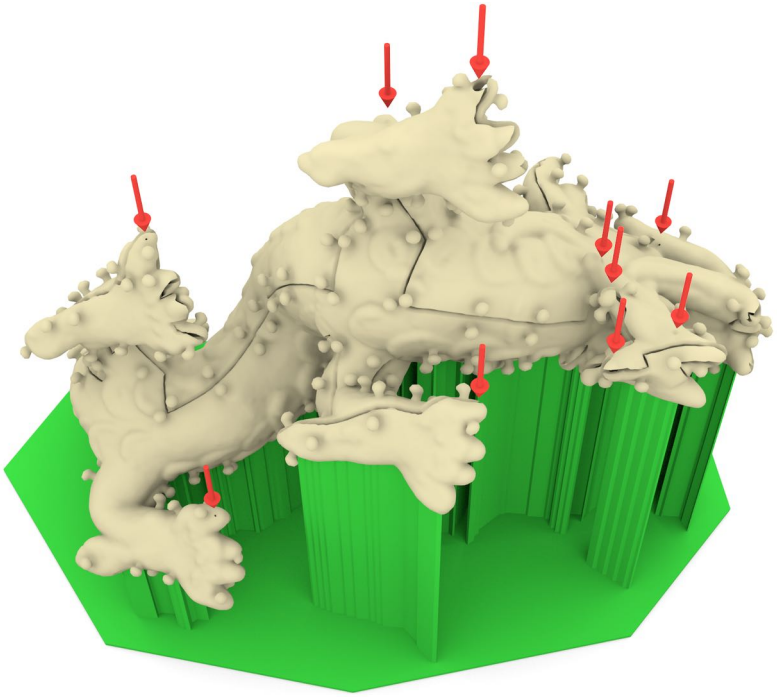


Figure 4.16: The holes needed to cast the dragon model after the reduction process has been applied.

(a) (b)

Figure 4.17: (a) The mold is tied together before the resin is poured; (b) the result of the cast of the elephant model.

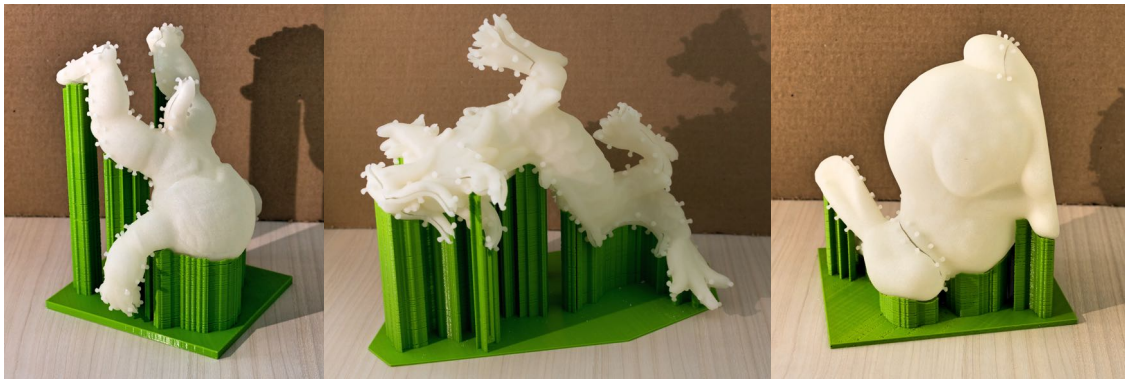


Figure 4.18: To keep the models in their correct position, some simple 3D printed supports are used.

envelope support (see Figure 4.18). As a byproduct, this support helps to more evenly distribute the overall deformation under the pressure of the cast liquid.

4.4 Results

We evaluated our approach by generating cut layouts on a wide range of different shapes and initial partitionings. Then we actually 3D printed several of the most interesting FlexMolds and used them in practice to cast multiple copies of objects. In the following, we first validate some of our design choices and then demonstrate the effectiveness of our approach by showcasing several complex models that can be easily reproduced using our method.

Initial patch layout Our approach for generating a proper cut layout works with any reasonable initial patch layout. Figure 4.19 shows the result of our algorithm when applied to different initial partitionings. The first row shows the initial partitioning we obtained by using Variational Shape Approximation [CSAD04], a Voronoi-based mesh partitioning with Lloyd relaxation [Lév14], the Almost Isometric patch layout of [PTC10], and the field-coherent quad patch layout decomposition method proposed in [PPM+16]. The second row shows the final cut layout, while the final row shows a snapshot of the extraction sequence. To evaluate the impact of different initial partitioning on the final cut layout, we measured how the max stretch induced by the extraction grows with the shortening of the cut. We used the cut length as an intuitive measure of the quality of a layout. If two layouts allow the extraction of their corresponding molds with the same deformation and arguably a similar effort, the shorter one must be preferred for practical reasons (easier to be closed and sealed). Figure 4.20 plots the relation between cut length and maximum deformation for the extraction during the optimization process. It has to be noted that the relation is not always

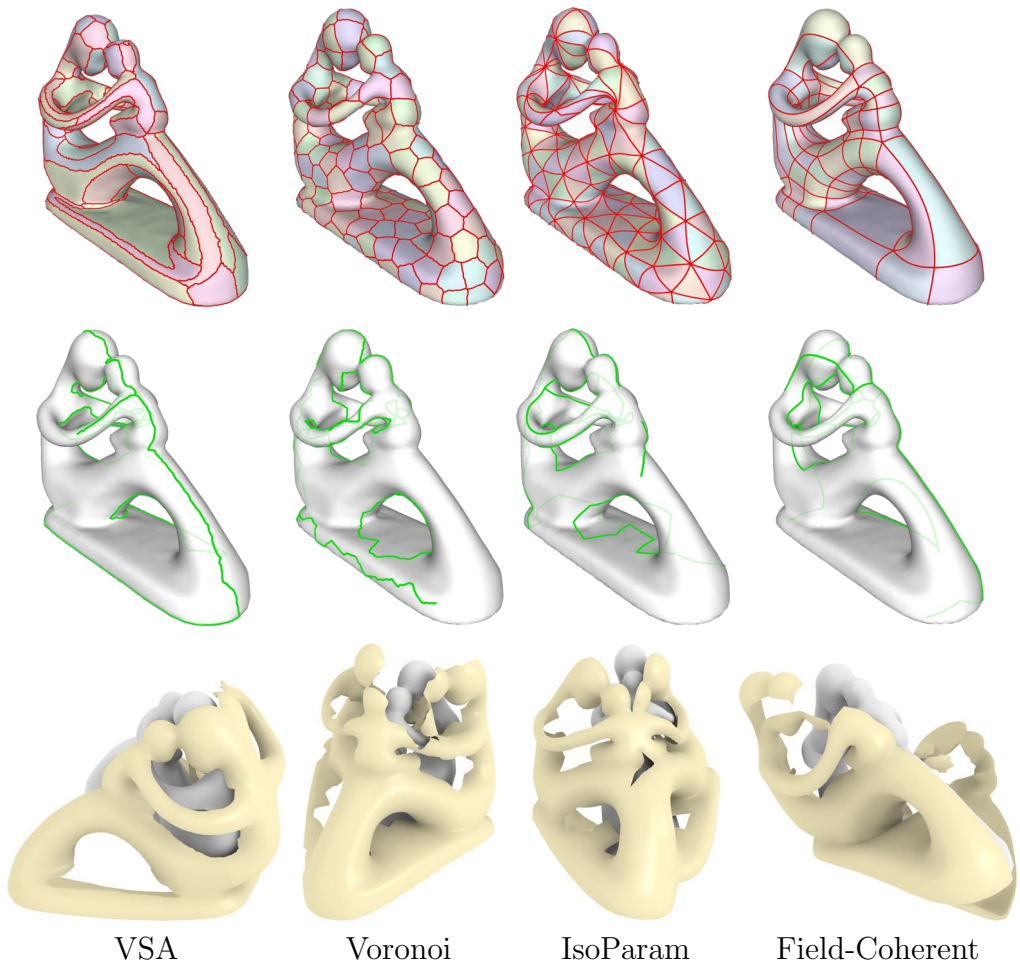


Figure 4.19: Initial patch layout, the produced final cut, and a step of the extraction procedure for the approaches proposed in [CSAD04], [Lév14], [PTC10], and [PPM+16].

monotone. Sometimes a change of topology may force the detaching procedure to choose a significantly different extraction path, resulting in an abrupt reduction of stretch values.

From our experiments, we found that, while any kind of reasonable initial partition works, quad-based, curvature-aligned coarse layouts [PPM+16] offer the best cut-length/deformation ratio most of the time. In the experiment, quad-based patch layout reduces the average deformation of VSA partitioning [CSAD04] by 5.1%, the Voronoi partitioning by 6%, and the Isoparametrizaion [PTC10] by 3.4%. Finally curvature-aligned cuts tend to remain straight and more regular, which makes them easier to seal.

In order to speed up the optimization process, we perform a simple initial patch simplification step. We merge the patches where the average normal does not differ more than a given threshold ($\pi/16$), using a VSA-style [CSAD04] approach (see

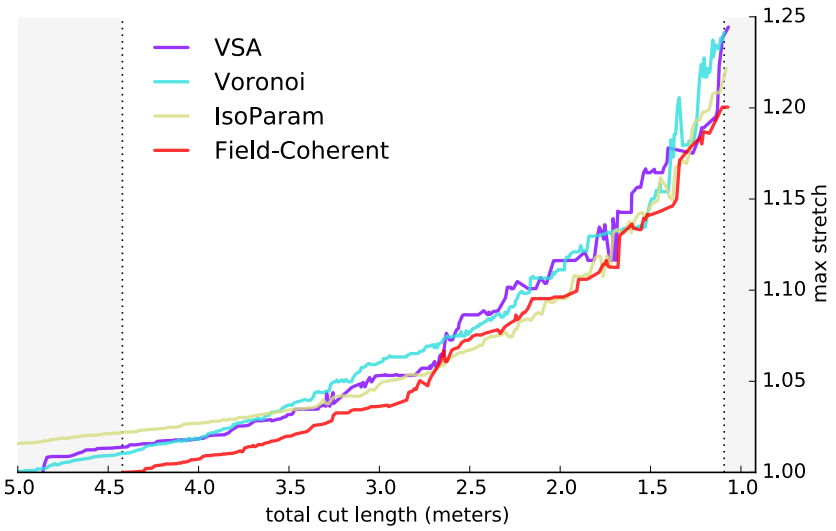


Figure 4.20: Maximum stretch reached (y axis) with respect to the total cut length (x axis) for the cut optimization process, using different initial cut layouts.

Figure 4.7).

Simple geometries We tested our method with several simple geometries, evaluating if the proposed approach produces reasonable cut layouts. As shown in Figure 4.21, cuts were left at locations where one would expect them intuitively. We can notice the slit at the cube edge, which allows the deformation of the top *hinge part* to be lowered during the extraction.

Fabricated examples We fabricated several models that are widely used in the Computer Graphics community to test our method. We used laser sintering to 3D print the flexible molds with thermoplastic polyurethane (TPU), a commercially available, flexible, and resistant material. The printing process we used requires a tolerance of 0.3 mm as a gap for the cuts; narrower gaps get fused. Our V-shaped cuts merged at the tip, but we were able to easily open them manually.

The first model we tested is the sculpt, as shown in Figure 4.6. This model is topologically challenging, requiring several cuts. However, due to the flexibility of the mold shell, we were able to fabricate the replicas with a single-piece mold. In comparison, methods based on rigid molds would require either a highly complex multi-piece mold or alternatively several cast operations to create pieces of the object that would then be assembled in a post-process. Instead, we are able to achieve reproduction in one single cast operation with a one-piece mold. We performed multiple casts with two different types of resins and gypsum. We did not post-process any of our models except by removing excess resin from the seams. All our models were produced by a novice user who had never performed any castings,

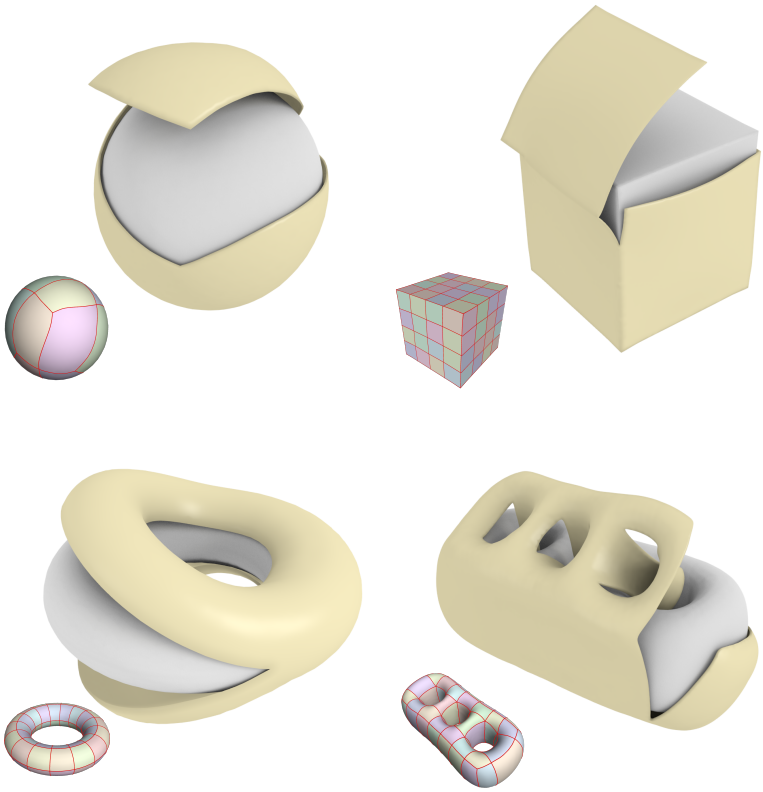


Figure 4.21: Some casts obtained for basic shapes.

demonstrating the easy applicability of our approach. Tiny air bubbles trapped in the resin could have been removed by performing vacuum degassing before casting, a step that we ignored to keep the process as simple and lightweight as possible.

Figure 4.22: Molds and casts obtained for the bunny, the fertility, and the armadillo models.

Figure 4.23: The result of the cast of the bimba model.

After this first experiment, we cast replicas of more complex models (see Figures 4.22 and 4.23). All the molds are composed of a single connected component. As shown by the armadillo model, we successfully capture the high frequencies of the surface detail and, as shown in the accompanying video, we can extract the object without damaging the surface details or the mold.

Finally, we pushed the proposed method to the limits by casting complex shapes with thin protruding features and high-frequency details. We successfully reproduced the dragon (see Figure 4.24) and the elephant (Figure 4.17) models that have thin tubular features (horns and fangs) and flat features (elephant ears). The elephant model required almost two liters of resin.

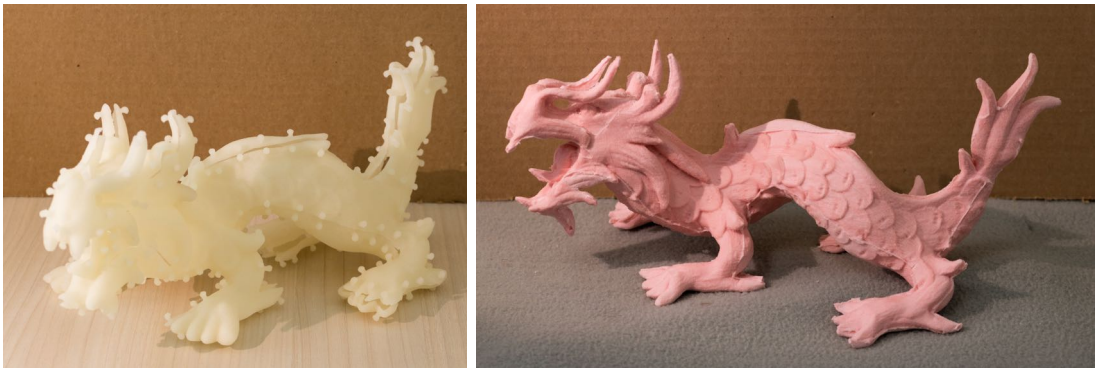


Figure 4.24: The result of the cast of the dragon model.

The numerical results of the patch decomposition of our algorithm are shown in Table 4.1. We successfully obtained one single patch for each model we processed. Processing time is provided for the algorithm both with and without the lazy update optimization strategy (see Section 4.2.1). As input decomposition we used [PPM+16] for the sculpt, fertility, bunny, bimba, and elephant models, and [ULP+15] for the dragon and armadillo models.

Table 4.1: Models on which we have tested our approach.

Mesh	start patches	reduced patches	final patches	air vents	# faces	time (minutes) w/ opt.	time (minutes) w/o opt.
sculpt	78	78	1	1	11k	15	21
bunny	138	138	1	2	31k	12	57
fertility	128	128	1	1	20k	23	91
armadillo	198	151	1	3	21k	12	67
bimba	544	151	1	3	27k	27	39
elephant	445	151	1	4	17k	33	73
dragon	604	151	1	10	12k	3	11

Chapter 5

Elastic Textures for Additive Fabrication

This chapter is based on our publication [PZM+15], in collaboration with Julian Panetta, Qingnan Zhou, Nico Pietroni, Paolo Cignoni and Denis Zorin. My contribution on this project included the development of the needed modeling tools for generating a 2D implementation of the algorithm; design of topological enumeration of patterns; formulation of tileability and printability constraints; setup for constrained coarse geometry sweep; shape boundary derivative implementation; supporting tools and design of most examples shown in this Chapter.

Rapid advances in the accessibility of additive fabrication has a significant impact on how manufacturable geometric models are constructed. A key distinctive feature of common additive fabrication technologies is that the cost and time of production are practically uncorrelated with structural complexity: in fact, a complex structure using less material may be both cheaper and faster to produce.

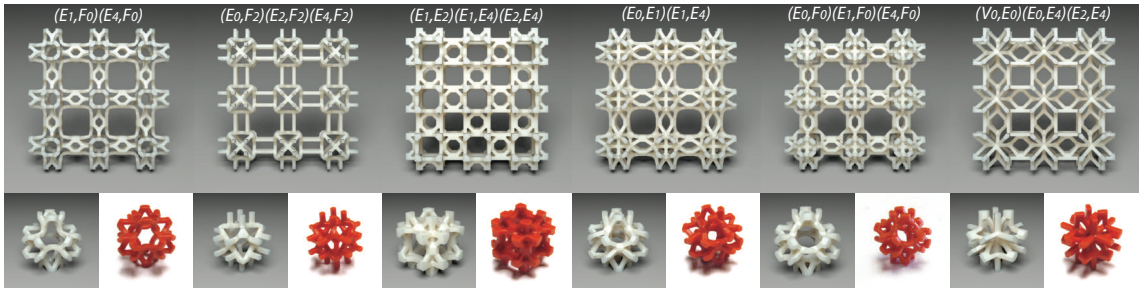


Figure 5.1: Six basic elastic textures are used to obtain a large range of homogenized isotropic material properties. A $3 \times 3 \times 1$ tiling of each pattern is shown, along with rendered (left) and fabricated (right) cell geometry below. The naming convention is explained in Section 5.3.

Complex structures, aside from potentially reducing costs, open up many new possibilities, in particular for manufacturing deformable objects. By varying a small-scale structure, one can adjust a variety of material properties, from elasticity to permeability. Importantly, these properties can be varied nearly continuously over the object, something that is not commonly done in traditional processes. As it was observed in prior work, this opens up many new possibilities for object behavior.

Small-scale structures present a set of new design challenges: in all but the simplest cases, these are hard or impossible to design by hand to meet specific goals. At the same time, computational optimization of fine-scale variable structure over a whole object, even of moderate size, can easily result in numerically difficult topology and shape optimization problems with millions of variables.

In this chapter, we describe *elastic volumetric textures*, a library of tileable parameterized 3D small-scale structures that can be used to control the elastic material properties of an object. Applying such textures to a hex mesh with target material properties specified per element is similar to using dithering to achieve a continuous variation of brightness or color.

In a sense, almost *all* material properties owe themselves to small-scale structures at the molecular or crystal level, and a large body of work in nanoscience aims to control material properties precisely by structure design. These works must accommodate constraints imposed by the specific properties of the elements and molecules used, the need for self-assembly, and other considerations.

Our focus is on larger-scale structures, which can be manufactured using existing 3D printing technology. With feature sizes at the scale of 10μm-100μm, these are well described by conventional elasticity theory. While this type of structure was also extensively studied, typically this was in the context of a specific problem, such as optimizing strength for a given material volume fraction. Our goal is to maximize the range of effective material properties that can be obtained using a single material by varying the structure.

We consider variable-thickness *truss-like structures*—i.e. structures composed of connected bars—as these cover a considerable range of properties on the one hand, and on the other hand, allow us to work with a relatively small number of parameters. We present a method for building a dictionary of structures that cover a large space of material properties. These structures are *tileable*, which makes it possible to vary material properties across an object, and *printable*.

We demonstrate that elastic volumetric textures allow one to control the deformation behavior of objects, either by painting material properties directly or by a two-stage *shape optimization* procedure, involving solving for variable continuous properties then approximating them using our texture dictionary. We validate our results by measuring samples for different choices of parameters and topologies and by demonstrating the deformation behavior of objects fabricated with spatially varying structures.

5.1 Related Work

Microstructure design and optimization. There is a huge literature on theoretical studies of effective moduli of composites (our periodic structures are an extreme example of a composite combining a material with void). Recent monographs include [CD99, Mil02, Tor02]. Much of the literature focuses on identifying microstructures with *extremal* effective behavior, i.e., with effective elasticity properties at the boundary of the achievable zone for a given class of composites [All02, Che00, Mil02]. Many classes of extremal structures were described (see, e.g., [CZCL13]), however most of these classes—e.g. sequentially laminated microstructures [Ave87] and microstructures based on inclusions [GK95, LJL07]—are either difficult or impossible to manufacture at this time. Interchangeable composites and other structures were found that maximize simultaneously, e.g., the bulk modulus and permeability [GP06] or electrical conductance [THD02, THD03, TD04], but these designs are of limited use for tailoring elastic behavior.

The closest work to ours is [Sig95], which constructs truss microstructures with prescribed elasticity tensors. It starts with a full "ground structure" containing about 2000 candidate members, then optimizes the members' thicknesses *but not offsets* to obtain a microstructure period cell whose homogenized properties (computed using a truss model) match the desired properties. Neither tileability of structures for different parameters nor printability can be guaranteed. We discuss the differences in greater detail at the end of Section 5.3. Further exploration of periodic structures of this type was done more recently in [CEGR10], comparing different methods for optimizing these structures.

A number of microstructures were obtained using various types of *topology optimization*, which was originally designed for global structure optimization. In the case of microstructure design, these methods look for a periodic structure minimizing, e.g., compliance for a fixed total volume fraction. The result is normally a single-scale structure, with scale controlled by the resolution of the simulation grid or other types of regularization. Important methods proposed for solving these problems include solid isotropic material with penalization (SIMP) and rational approximation of material properties (RAMP) [Ben89, BS03, NS10]. [RHX13] demonstrated design of isotropic materials maximizing bulk modulus.

Topology optimization offers more flexibility in the choice of structure, but it requires a relatively expensive optimization for each specific problem. The ability to undergo topological transitions under continuous parameter changes is both a strength, as it allows exploration of a broader space of structures, and a weakness, as it considerably complicates design of parametric families satisfying printability and tileability constraints, which motivates our approach.

Microstructure fabrication. Several groups focusing on additive fabrication have recently obtained encouraging results. In particular, materials previously

thought to be unmanufacturable were produced and behave as expected. Notably, the work of Hollister and collaborators [LKH04, LHCH04, Hol05, Kan10] in the context of bone scaffold design and fusion cage design demonstrated the use of optimized microstructures. The possibility of manufacturing auxetic (negative Poisson's ratio) materials was demonstrated in [GGLR11], and in [SWL+11, BSK+12, ALS14].

The idea of fabricating tileable structures with varying properties also appears in [HL09] in which the authors discuss "digital materials", as composed of a set of discrete voxels with predefined shapes that can be connected. Similarly, a building-block based approach was also used in the context of bio-printing [MVK+09], where the authors use spheroids of living materials with evolving and controllable composition, varying material and biological properties in time.

Compliant mechanisms. The material optimization method that we present solves a similar problem to that of compliant mechanism design. [BS03] reviews several existing approaches to designing mechanisms that maximize mechanical advantage/output deflection or tune an output displacement to a particular path. These approaches have little control over the resulting structure's macroscopic shape, whereas tuning deformation behavior using our microstructure approach creates a "mechanism" that still looks like the input shape.

Fabrication and computer graphics. A broad variety of fabrication-related work has been done in the computer graphics community (see Chaper 2).

A close work to ours [BBO+10], introduces an optimization process to find the best combination of stacked layers to satisfy an input deformation, enabling fabrication of objects with complex heterogeneous materials using multi-material 3D printers. Our work can be viewed as complementary, focusing on the design of structures that can be, e.g., used as a part of deformation behavior design; our material optimization method provides an alternative to the method in that paper. In [STC+13], multi-material printing and discrete material optimization is used in a similar way on complex characters to achieve desired deformations with actuation. Our elastic textures can be viewed as a tool for solving this type of problem. Our structures also can be employed in systems like [CLD+13] and [VWRKM13].

Homogenization. A central tool in our work, homogenization was used in graphics for reducing complexity of physical models in [KMOD09], finding the constitutive parameters of a low resolution discretization that best approximates the behavior of the original higher complexity material. The periodic homogenization method that we use is based on the one described in [All02].

5.2 Overview and Main Results

In this section, we describe our overall approach, visualized in Figure 5.2, and a specific set of patterns that we have obtained.

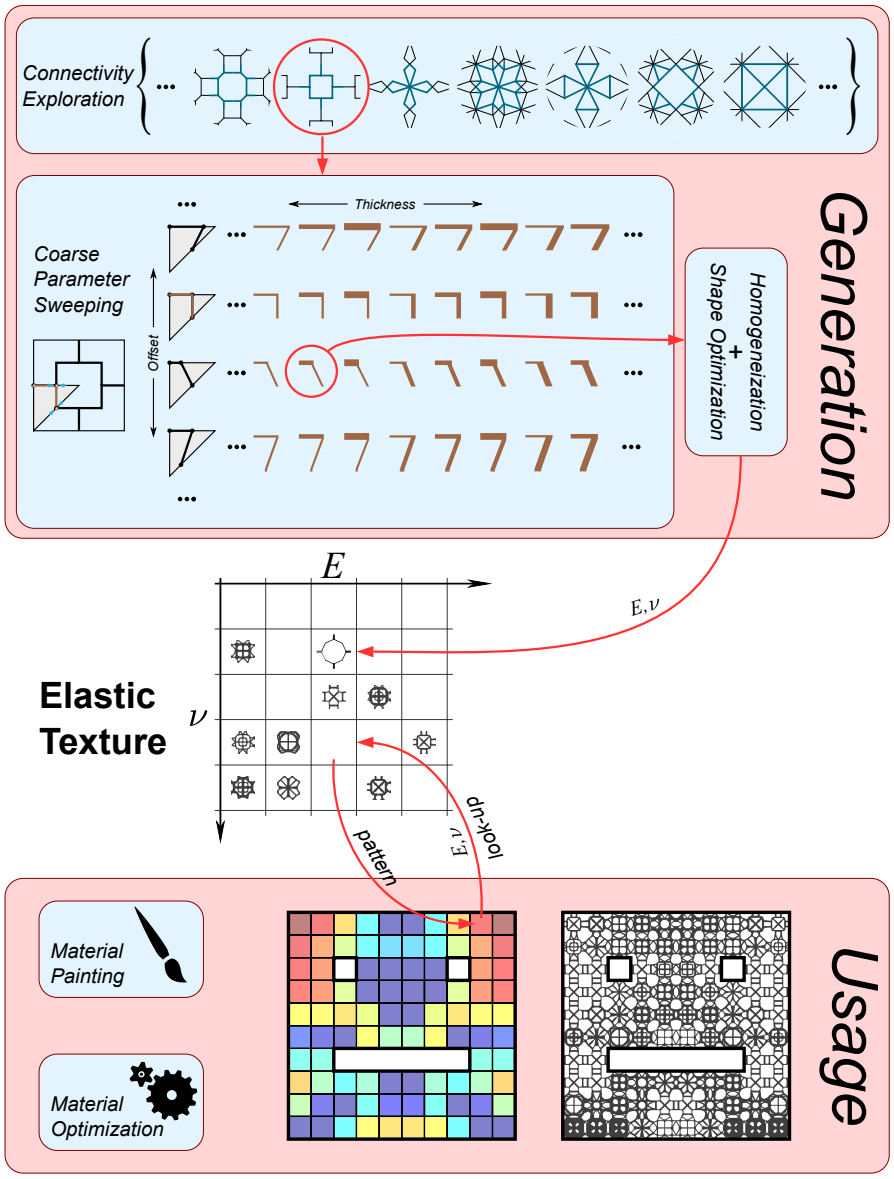


Figure 5.2: Overview of elastic texture generation and use.

Problem. The general problem we solve can be formulated as follows: *for each tensor C from a given range of elasticity tensors, and a base isotropic material with Young's modulus E^b and Poisson's ratio ν^b, find a structure made out of the base*

material in a unit cubic cell, such that if the cell is infinitely tiled in space, the resulting homogeneous material has elasticity tensor C.

As discussed in the introduction, we aim to construct a family of patterns that are printable and tileable to enable creation of variable material properties.

Printability is heavily dependent on the choice of technology. We focus on printability criteria related to stereolithography, the most accurate 3D printing method available at this time, but our approach can be easily modified to handle other technologies.

As the printing process proceeds layer-by-layer, we assume that the structure is defined with respect to a fixed coordinate system X, Y, and Z aligned with the printer, with Z being vertical. The (idealized) printability criteria that we use are:

1. There are no enclosed voids.
2. For any point of the structure, the extent covered by the structure in the X, Y, and Z directions from the point are above a printability threshold $d_{\min}$.
3. Every point of the pattern is supported: for every XY slice, all connected components of the slice have at least one point connected to lower points in the structure by a segment contained in the structure. While this condition does not prevent long horizontal bars supported at single points, which can be difficult to print, we have found it sufficient in practice for pattern sizes up to 10mm and a $d_{\min}$ of 0.3mm.

We also make our primary goal to generate periodic structures with *isotropic* homogenized properties. Such patterns have the elasticity tensor C defined by two parameters, Young's modulus E and Poisson's ratio ν, and its inverse, compliance tensor S, has the (Voigt notation) form

$$S = \frac{1}{E}\begin{pmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{pmatrix}. \tag{5.1}$$

Expressing in terms of the shear modulus, $G = E/(2(1+\nu))$, the last three diagonal terms of S are simply $1/G$.

While for many tasks anisotropic materials are either sufficient or preferable, periodic structures with isotropic homogenized properties are easiest to use, as cell orientation is decoupled from material properties. In addition, once an isotropic starting point is obtained, it is easy to obtain a controlled anisotropic behavior.

Searching the space of all possible structures in a cell, even at a finite resolution, is an impossible task. Instead, we choose a space of structures with a limited but sufficiently large set of parameters, that can be optimized to achieve specific material properties.

Truss-like structures. We focus on truss-like structures (*patterns*) as shown in Figure 5.1, consisting of bars of different thicknesses connecting a set of nodes in the cell. Unlike real truss structures, the connections between bars are not pin joints, and flexural rigidity at the nodes plays a major role.

This particular space of structures is motivated by several considerations. First, the space is known to contain both very stiff and very weak patterns, providing a broad range of behaviors. Second, tileability and printability requirements yield specific geometric conditions, expressed mostly as constraints on the structure's geometry. For example, the requirement of no enclosed voids is automatically satisfied if the frame structure has no self-intersections; the bound on extents can be obtained by bounding the thickness from below; and the support condition is easily formulated as a constraint on node positions. These conditions are detailed in Section 5.3.

Symmetry considerations, as well as restrictions on the number and placement of nodes, yield a space of patterns parametrized by their set of edges connecting some subset of the 15 candidate nodes we define on a tetrahedron (their *topology*), thicknesses of these edges, and offsets of the nodes from their default positions. This space is still very large, and we explore it using both topology and geometry searches, described in Section 5.3. These rely fundamentally on the homogenization and shape optimization procedures described in Sections 5.4 and 5.5.

Resulting family. The search procedure's final result is shown in Figure 5.3; the six pattern topologies themselves are illustrated in Figure 5.1. The complex boundaries of the $(E,\ \nu)$ regions arise from the multiple types of geometric constraints enforcing printability.
Figure 5.4 shows some patterns with topology "(E1,E2)(E1,E4)(E2,E4)".

The family of topologies covers a large range of Young's moduli, with a largest-to-smallest Young's modulus ratio of 1800, and a sizable range of Poisson's ratios, -0.16 to 0.48.

We note that the range of negative Poisson's ratios is somewhat limited, while on the high end, we are able to achieve ratios close to the theoretical maximum. This observation is consistent with [Sig95]: while it is relatively easy to obtain more extreme negative Poisson's ratios for patterns with cubic symmetry but with shear modulus too low for isotropy, the isotropy requirements restrict the range. Printability constraints restrict it further.

Quite remarkably, four out of the six topologies can be transformed into each other by simple operations (single vertex splits, addition of cross-shaped supports connecting some nodes). The other two are also related to each other by a simple transform, but are not related to the first sequence.

We do not claim that the proposed family is in any sense optimal. It is most likely possible to extend the coverage or to cover the same domain with fewer topologies. However, the presented set is already quite useful for controlling material properties,

as the examples of Section 5.6 show.

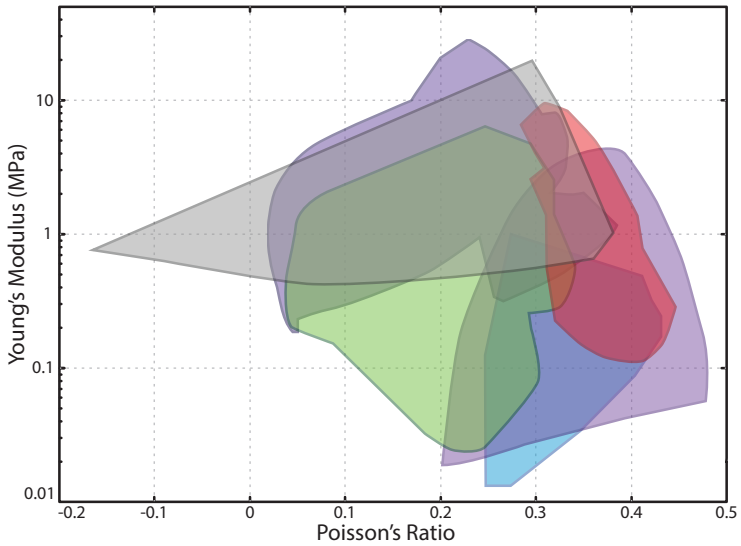


Figure 5.3: Region of the (E, ν) space covered by the selected set of patterns. Each topology's coverage is shown in a different color.

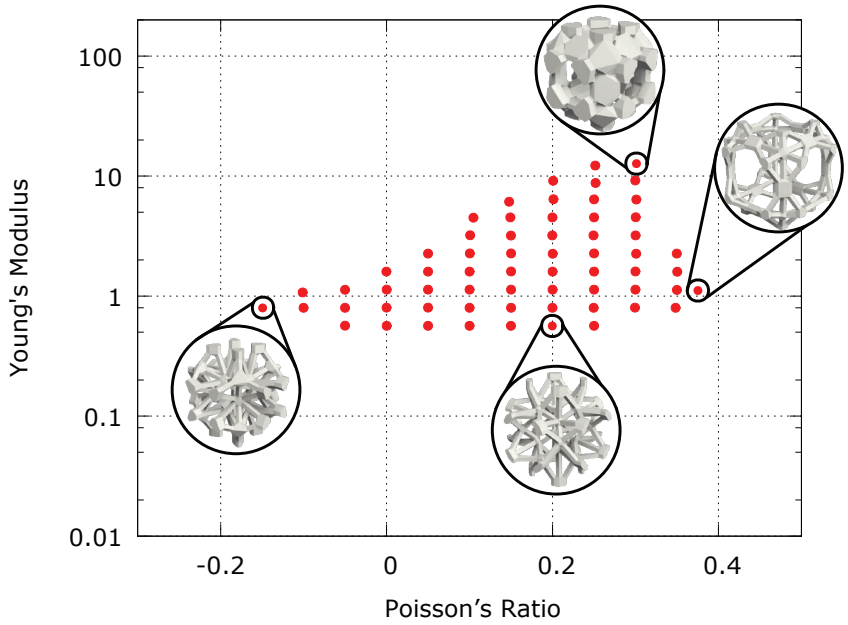


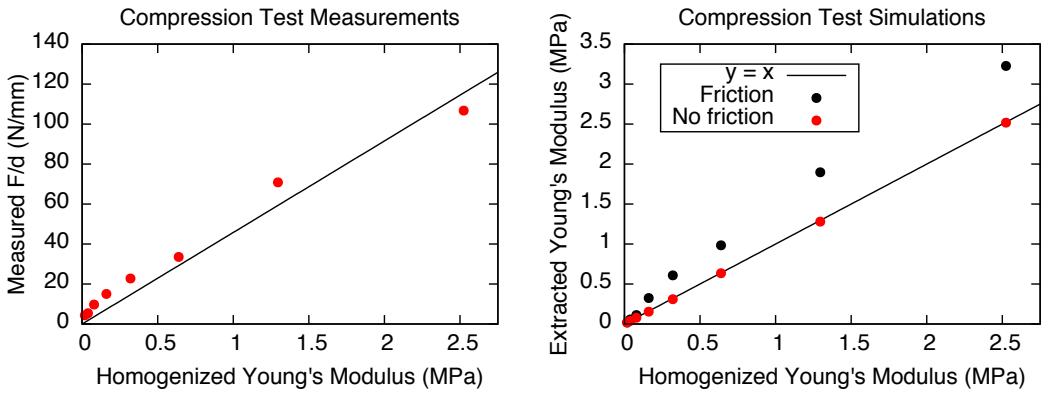
Figure 5.4: Samples of the (E, ν) space reached by patterns with topology "(E1,E2)(E1,E4)(E2,E4)".

Accuracy. We fabricated eight patterns with different homogenized Young's moduli using the B9Creator SLA printer and tested their stiffness using the BOSE

ElectroForce 3200 measurement system. The machine gradually compressed our samples in the Z direction between two compression plates and measured the displacement resulting from the applied force at each step. We used $6 \times 6 \times 2$ tilings of 5mm cells for this test.

We note that our measured force/displacement slopes are roughly proportional to the homogenized Young's moduli (Figure 5.5a), implying that the measurements are consistent with some (unmeasured) base Young's modulus. The curvature seen could be explained partly by friction in the compression testing setup (Figure 5.5b). Another significant source of error is the inaccuracy of our B9Creator, which tends to thicken thin geometries.

We used a lower-accuracy setup to measure Poisson's ratio but still obtained reasonable agreement with homogenization (Figure 5.6). We compressed the microstructures in the Z direction between two lubricated metal blocks and manually measured the expansion/contraction in the X and Y directions. From these displacement measurements, we computed the X, Y, and Z strains and their ratios.



(a) (b)

Figure 5.5: Compression test results for eight patterns with varying homogenized Young's moduli ($6 \times 6 \times 2$ tiling of 5mm cells).
(a) Slopes extracted from the measured force vs. displacement curves along with a best-fit line through the origin. (b) Moduli extracted from simulated compression tests, with and without modeling compression plate friction. Without friction, the simulated test agrees with homogenization perfectly, but friction introduces error.

We also validated the patterns' isotropy by printing a block filled with a tiled pattern that was rotated by 45° around the Z axis and clipped (Figure 5.7). The measured effective Young's moduli in the rotated orientations were in good agreement with the unrotated orientation: a compression test in the X, Y, and Z directions extracted effective Young's moduli of 0.635MPa, 0.6293MPa, and 0.628MPa for the example shown.

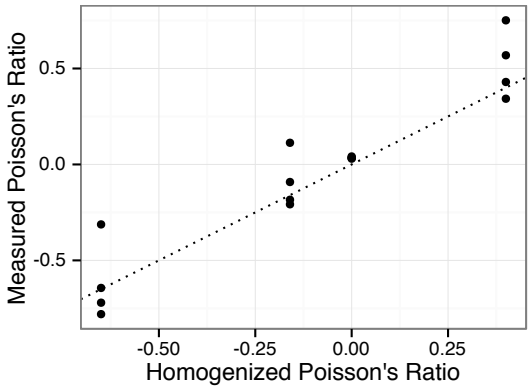


Figure 5.6: Poisson's ratios measured from $3 \times 3 \times 1$ printed tilings of 10mm cells vs. homogenized properties. The $\nu = -0.67$ sample, outside our family's range, violates isotropy and printability constraints (we added support structure manually for this experiment).

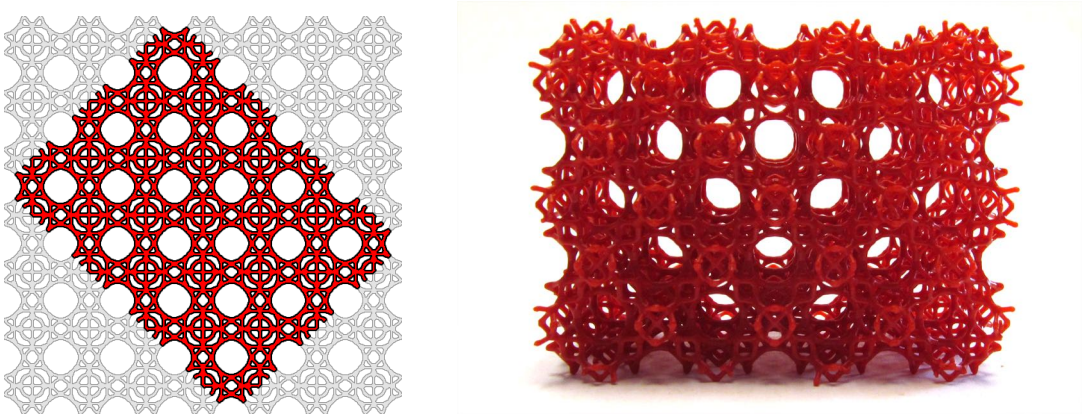


Figure 5.7: We extracted a 45° rotated rectangular block from a regularly tiled 10mm cell microstructure to test Young's modulus in non-axis aligned directions.

Base Material. We used estimated base material properties of $E^b = 200$MPa and $\nu^b = 0.35$ for all results. We estimated E^b using a three point bending test on rectangular bar samples, but the base Young's modulus can also be estimated from the microstructure compression test.

Using a different base Young's material would not qualitatively change our results, apart from making the patterns uniformly softer or stiffer. The homogenized Young's moduli depend linearly on E^b, and the Poisson's ratios are independent of E^b, so scaling the base modulus simply softens/stiffens every pattern by the same factor. In particular, for the displacement-based material optimization of Section 5.6, fabricating the solution with a different base Young's modulus maintains the same target deformation behavior (although the required force will change).

We note that our patterns are not very sensitive to moderate changes in the base Poisson's ratio. Changing from $\nu^b = 0.35$ to $\nu^b = 0.35 \pm 0.05$ results in a median relative change in Young's modulus of 0.683% (max: 1.88%) and a median absolute change in Poisson's ratio of 0.00229 (max: 0.0129) over all patterns. Since most additive fabrication materials fall within this range, we expect similar results for other printers and materials.

5.3 Search for Efficient Patterns

In this section, we describe the class of patterns that we consider and the main steps of the search method.

Ground class of patterns. The topology of patterns is defined by a set of edges connecting nodes in the cube cell. We generate the geometric variations by adding *offsets* to the node positions and by changing edge *thicknesses*.

Motivated by the isotropy requirement, we constrain our search to patterns with cube symmetries, which are guaranteed to have the same Young's moduli and Poisson's ratios in every axis-aligned direction. That is, the compliance tensor has the form (5.1) *except* the last three diagonal entries $1/G$ may not equal $2(1+\nu)/E$. This yields an easy-to-check isotropy measure:

$$A = 2(1+\nu)G/E, \tag{5.2}$$

which we use to identify isotropic patterns in our search. The symmetry will also dramatically reduce the space of pattern topologies to a tractable size after a few additional constraints are introduced. Note, however, that cube symmetry is not necessary for isotropy; other isotropic structures exist with, e.g., tet symmetry.

Consider the group of symmetries of a cube O_h, which includes reflections about three symmetry planes orthogonal to the X, Y, Z axes and the six planes orthogonal to the bisector of each pair of axes. By partitioning the cube according to these symmetry planes, we obtain 48 equal tetrahedra as in Figure 5.8a. O_h maps a single one of these tetrahedra to any other, so it is sufficient to define the nodes and edges of the pattern graph—as well as their offsets and thicknesses—on a single tetrahedron.

We generate the different topological configurations by changing the connectivity between 15 nodes on a tetrahedron (see Figure 5.8a): vertex nodes $\{V_0, V_1, V_2, V_3\}$, edge nodes $\{E_0, E_1, E_2, E_3, E_4, E_5\}$, face nodes $\{F_0, F_1, F_2, F_3\}$, and a single internal node, T_0. Configurations are named by their graphs' edge sets (see labels in Figure 5.1). Each node is constrained to stay on its respective simplex to preserve the topology, so vertex nodes are fixed, edge nodes have a single offset, and so on (Figure 5.8b).

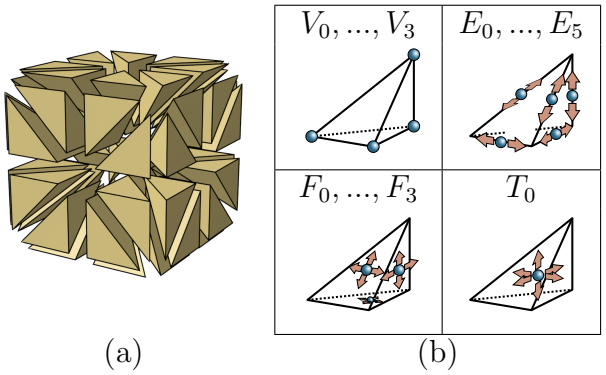


Figure 5.8: (a) The tetrahedral cube decomposition used to generate 3D patterns; (b) The 15 nodes defined on a tetrahedron together with their degrees of freedom.

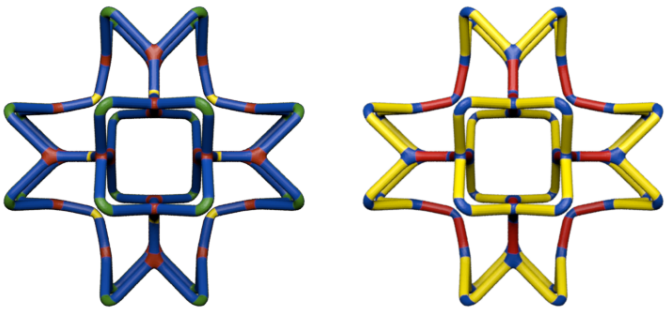


Figure 5.9: Symmetry orbits are colored with yellow, red and green. Left: vertex symmetry orbits. Right: edge symmetry orbits.

Figure 5.9 shows an example topology colored by its node and edge orbits with respect to symmetry group O_h, and Figure 5.10 demonstrates the effects of the node offset and edge thickness parameters.

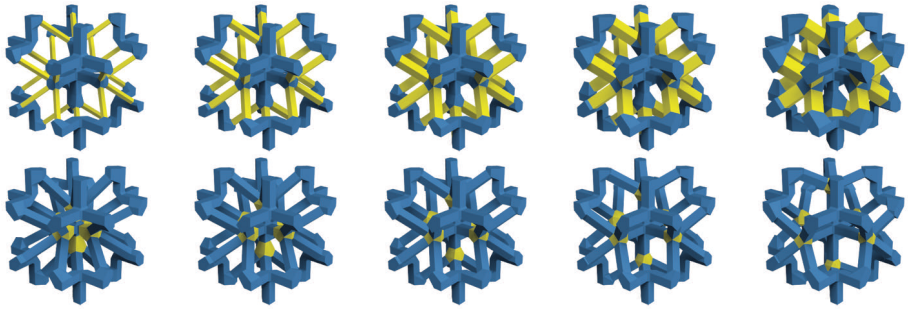


Figure 5.10: The results of varying the thickness (top) and offset (bottom) parameters of a particular pattern topology.

The space of possible connectivities, even after accounting for symmetries, is far too large to explore completely (on the order of 10^{32} configurations). We enforce the following constraints to reduce the space of patterns:

- **Connected**: the tiled pattern is a single connected component.
- **No coinciding edges**: no edge is contained within another. E.g., if graph edge (V_0, V_1) is chosen and E_0 is the midpoint node of the corresponding tet edge, graph edge (V_0, E_0) is forbidden since it overlaps the first for any offset.
- **No dangling edges**: every node has valence greater than 1.
- **Number of edges**: at most 3 graph edges per tetrahedron.
- **Max node valence**: node valences do not exceed 7.

Valences are computed on the graph after periodic tiling of the cube cell. The first two criteria reduce the space to 16221 topologies, and the remaining three to 1205 topologies.

Printability. For truss-like patterns, printability is affected by two main factors: the pattern graph structure and the edge thicknesses.

The first printability criterion can be defined on the nodes by considering their offset positions. We say that a node n_1 has supporting node n_2, if these are connected by an edge and n_1 is strictly above n_2. We say that n_1 and n_2 are at the same level if they have equal Z coordinates. A pattern is printable only if every connected set of nodes at one level has at least one supporting node.

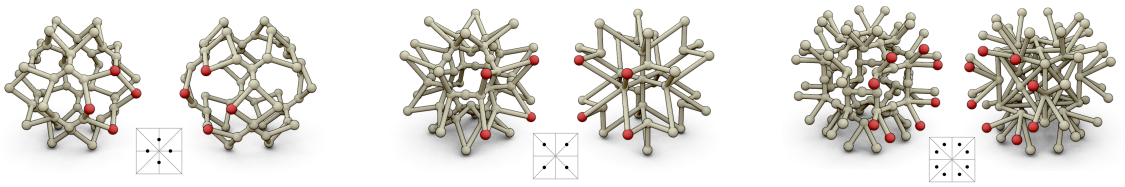


Figure 5.11: Two pattern topologies from each of three different families, shown with the families' *interfaces* (nodes on the cube cell faces).

Printability can be tested by a simple algorithm: we first mark as supported all nodes with a supporting node (considering periodicity). Then we propagate the front of supported nodes to neighbors at the same level. When this breadth first search terminates, the pattern is printable if and only if all nodes are marked as supported. The procedure is illustrated in Figure 5.12. We also note that this constraint can be expressed algebraically as a set of inequality constraints on the offset variables, which can be enforced by an optimization solver.

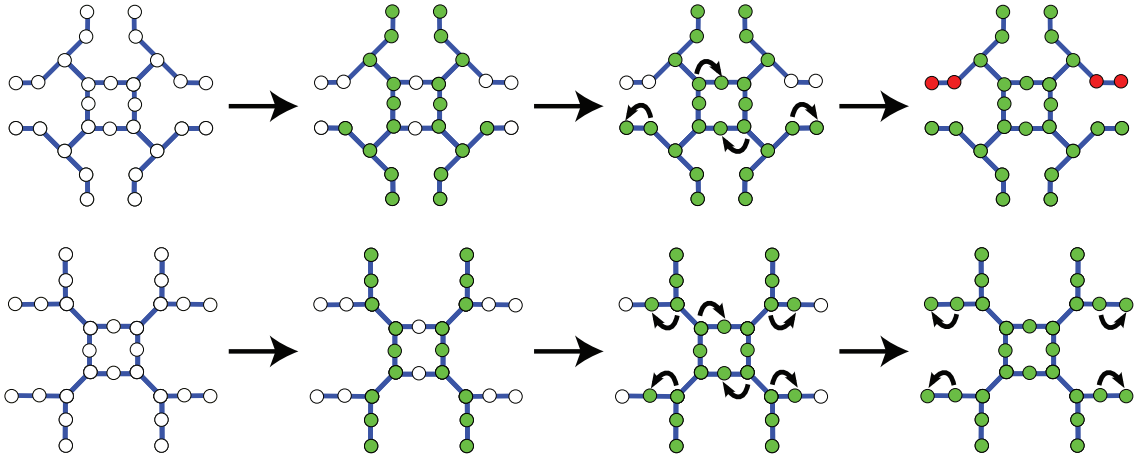


Figure 5.12: 2D examples of the printability detection algorithm. Vertices with supporting nodes are marked (green), then a breadth-first search extends the supported vertex front to horizontal neighbors. The remaining unmarked nodes are unsupported (red). Two cases are shown: unprintable (top) and printable (bottom).

Tileability. The tileability requirement means that all pattern topologies should belong to the same *family*, meaning topologies with the same set of nodes and edges appearing on the faces of the cube cell (Figure 5.11).

Searching the space of topologies. The goal of our search is to identify a family of pattern topologies that covers as much as possible of the (E, ν) space

while satisfying the printability and tileability requirements.

The initial space consists of all pattern topologies satisfying the constraints on graph connectivity mentioned previously. We proceed in the following steps:

1. *Coarse geometry sweep.* Geometric variations are generated for each pattern topology by trying thicknesses of 0.3mm and 0.7mm and node offsets corresponding to barycentric coordinates of 0.2, 0.35, 0.5, 0.65, and 0.8 on the associated tetrahedron simplex. The resulting *printable* patterns are meshed, self-intersecting meshes are discarded, and the remaining patterns' effective elasticity parameters are computed using periodic homogenization (Section 5.4).

2. *Isotropy filtering.* We select a subset of the patterns closest to isotropic (we use a heuristic bound of $0.8 < A < 1.2$), which we consider promising candidates as starting points for optimizing pattern parameters to precisely match a range of isotropic elasticity tensors.

3. *Topological family selection.* At this point, we have a rough map of the area in (E, ν) space covered by our set of patterns. We obtain a rough estimate of each topology's coverage by taking the convex hull associated with its nearly isotropic geometric configurations. We manually pick the single family whose pattern topologies cover the largest region of (E, ν) based on these estimates.

4. *Selection of a minimal covering set of topologies.* For the selected family, we run a finer sweep of offsets and thicknesses, again filtering for printability, to compute a more precise estimate of the boundary of the (E, ν) domain that each pattern topology can cover. Among all topologies in the family, we selected 6 such that the union of their coverage areas contains most of the domain covered by the family.

5. *Lookup map construction.* Finally, using the shape optimization machinery of Section 5.5 and the initial nearly-isotropic points for each of the 6 topologies chosen, we optimize each patterns' parameters to reach a grid of isotropic elasticity tensors evenly spaced in $(\log(E), \nu)$.

Our procedure is similar to [Sig95], but with several key differences. First, [Sig95] uses a simplified truss model, whereas our method directly homogenizes and optimizes the printable geometry. Second, by using a full topology (including all possible edges between nodes in a ground structure) and permitting zero edge thickness in optimization, the work avoids the topology enumeration stage. As a side-effect, it cannot accommodate the lower thickness bounds or support criteria needed for printability. While a mixed-integer formulation like [MK13] could allow enforcement of d_{min} by introducing separate binary variables to disable members, this would involve a difficult mixed-integer nonlinear programming problem in our

tensor-fitting setting. Finally, by introducing offset variables, our novel shape optimization approach enables much finer control of the elasticity tensors as the design is not limited to the discrete node positions of a ground structure.

5.4 From Patterns to Material Properties

Our goal is to find the homogenized elasticity tensor C^H, describing the effective properties of the microstructure when it is fabricated at a small enough scale and periodically repeated to fill the space. This elasticity tensor is almost never the spatial average of elasticity tensors (for example, if a cell is almost completely filled with material but is disconnected from other cells, it has zero Young's modulus). We first define more precisely what a homogenized elasticity tensor is and then explain how to compute it.

Defining the homogenized tensor. Consider heterogeneous object Ω^ϵ filled with a periodic microstructure, as shown schematically in Figure 5.13. Parameter ϵ determines the size of cell Y relative to the object Ω^ϵ and permits asymptotic analysis as $\epsilon \to 0$.

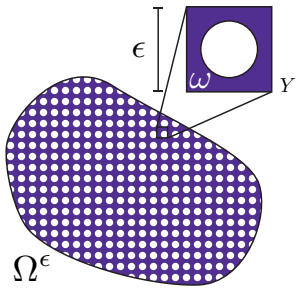


Figure 5.13: (Schematic) Periodic tiling of a domain Ω with base cell Y having geometry ω and length scale ϵ.

The elastic response of an object under macroscopic external load $\mathbf{f}$ is governed by the linear elastostatic equation

$$-\nabla \cdot [C : \varepsilon(\mathbf{u}^\epsilon)] = \mathbf{f} \text{ in } \Omega^\epsilon, \tag{5.3}$$

complemented by appropriate boundary conditions. Here, C is the periodically varying elasticity tensor, and $C : \varepsilon$ denotes its double contraction with strain ($C_{ijkl}\varepsilon_{kl}$) to compute stress. Considering a sequence of problems indexed by ϵ and letting $\mathbf{u}$ denote the limit of $\mathbf{u}^\epsilon$ as $\epsilon \to 0$, the homogenized elasticity tensor C^H is defined as the tensor that satisfies

$$-\nabla \cdot [C^H : \varepsilon(\mathbf{u})] = \mathbf{f} \text{ in } \Omega, \tag{5.4}$$

with same boundary conditions. Vectors $\mathbf{u}$ and $\mathbf{u}^\epsilon$ denote the displacement, and $\varepsilon(\mathbf{u}) \coloneqq \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ is the Cauchy strain tensor.

Expressions for the homogenized tensor. The standard derivation of the homogenized elasticity tensor based on a two-scale asymptotic expansion is provided in the additional material. Here we give an intuitive motivation for the periodic homogenization equations.

$\mathbf{u}^\epsilon$ has a high frequency periodic component that is averaged out to obtain $\mathbf{u}$ as period $\epsilon \to 0$. So $\mathbf{u}(\mathbf{x})$ can be thought of as the average displacement over the infinitesimal base cell Y at point $\mathbf{x}$. Likewise, $\varepsilon(\mathbf{u})$ is the average strain in the cell. For the object to be in equilibrium, (5.4) should represent an average force balance over the cell, meaning $C^H : \varepsilon(\mathbf{u})$ should be the average stress tensor. That gives the following intuitive interpretation of C^H: it maps the average strain applied at a point to the average stress resulting within the microstructure geometry.

Thus, applying C^H to $\varepsilon(\mathbf{u})$ is equivalent to simulating the microstructure's deformation under that average strain and averaging its stress. We formulate this simulation inside a single base cell Y by assuming that the displacement consists of a linear term (with constant strain $\varepsilon(\mathbf{u})$) plus a Y-periodic "microscopic fluctuation" term, $\mathbf{w}$ (with zero average strain by periodicity). This assumption of periodicity is reasonable because, by the translational symmetry of an infinite tiling, every cell deforms identically. Now we simply solve for fluctuation $\mathbf{w}$ putting the microstructure in equilibrium:

$$-\nabla \cdot (C(\mathbf{y}) : [\varepsilon(\mathbf{w}(\mathbf{y})) + \varepsilon(\mathbf{u})]) = 0 \text{ in } Y,$$

where $\mathbf{y}$ is the microscopic variable (the coordinate in Y). Then the average stress is $C^H : \varepsilon(\mathbf{u}) = \frac{1}{|Y|} \int_Y C : [\varepsilon(\mathbf{w}) + \varepsilon(\mathbf{u})] \, \mathrm{d}\mathbf{y}$.

We can extract the components of C^H by applying it to the 6 canonical symmetric rank 2 basis tensors, $e^{kl} \coloneqq \frac{1}{2}(\mathbf{e}_k \otimes \mathbf{e}_l + \mathbf{e}_l \otimes \mathbf{e}_k)$. Each application amounts to solving the *cell problem*:

$$-\nabla \cdot (C^{\text{base}} : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}])) = 0 \text{ in } \omega, \tag{5.5a}$$

$$[C^{\text{base}} : \varepsilon(\mathbf{w}^{kl})]\hat{\mathbf{n}} = -[C^{\text{base}} : e^{kl}]\hat{\mathbf{n}} \text{ on } \partial\omega \backslash \partial Y, \tag{5.5b}$$

$$\mathbf{w}^{kl}(\mathbf{y}) \; Y\text{-periodic}, \tag{5.5c}$$

$$\int_\omega \mathbf{w}^{kl}(\mathbf{y}) \, \mathrm{d}\mathbf{y} = \mathbf{0}, \tag{5.5d}$$

where we rephrased the microscopic force balance as a PDE over ω, since for a structure printed with base material properties C^{base},

$$C(\mathbf{y}) = \begin{cases} C^{\text{base}} & \text{if } \mathbf{y} \in \omega, \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

The last constraint in (5.5) eliminates the rigid translation degrees of freedom that still remain after enforcing Y-periodicity.

The homogenized elasticity tensor components are finally just the average over Y of the stress components corresponding to e^{kl}:

$$C^H_{ijkl} = \frac{1}{|Y|} \int_\omega C^{\text{base}}_{ijpq} [\varepsilon(\mathbf{w}^{kl}) + e^{kl}]_{pq} \, \mathrm{d}\mathbf{y}. \tag{5.7}$$

It is worth noting that C^H does not depend at all on the macroscopic details (shape Ω, force term $\mathbf{f}$, or boundary conditions).

FEM implementation. The cell problems (5.5) are solved numerically by a quadratic tetrahedral FEM discretization of ω. The piecewise linear integrand in (5.7) is integrated with exact quadrature.

Given the wire network of the microstructure, defining its topology (Section 5.3), a volume mesh is generated following the STRUT algorithm given in [Har08]: (i) A polygon is created around both ends of each segment. (ii) For each vertex, the convex hull of the nearby polygons and the vertex is constructed and the polygons themselves are removed from the hull. (iii) For each edge, the convex hull of its two polygons is constructed, and again the two polygons are removed from the hull. A tetrahedral volume mesh is finally created from the resulting closed surface.

The linear elasticity solver must support periodic boundary conditions, which requires the tetrahedral mesh to have an identical tessellation on the opposite periodic cell faces.

Convergence rate. Remarkably, we have observed that the homogenized coefficients remain accurate when the microstructure varies across cells (with few or no repetitions). In our experiments, the deformation behavior of even very coarse tilings closely matches the homogenized behavior (Figure 5.14), and we expect similar agreement in general for smoothly varying loads.

5.5 Optimizing Pattern Parameters

An essential step for creating a map of elastic textures is optimizing a pattern with fixed topology to match particular elasticity parameters. This is achieved using shape optimization with respect to the pattern parameters.

The optimization problem. Our goal is to minimize a functional, $J(\omega)$, measuring the difference between the homogenized elastic properties of the pattern and a target elasticity tensor C^*. We choose an objective that is suitable for designing material distributions with large deformations under moderate forces. The distance of compliance tensors, $S^H - S^*$, as opposed to elasticity tensors, is the better choice, since the strain for a constant stress is directly proportional to S^H, *not* C^H. In fact, minimizing the Frobenius norm of $S^H - S^*$ can be

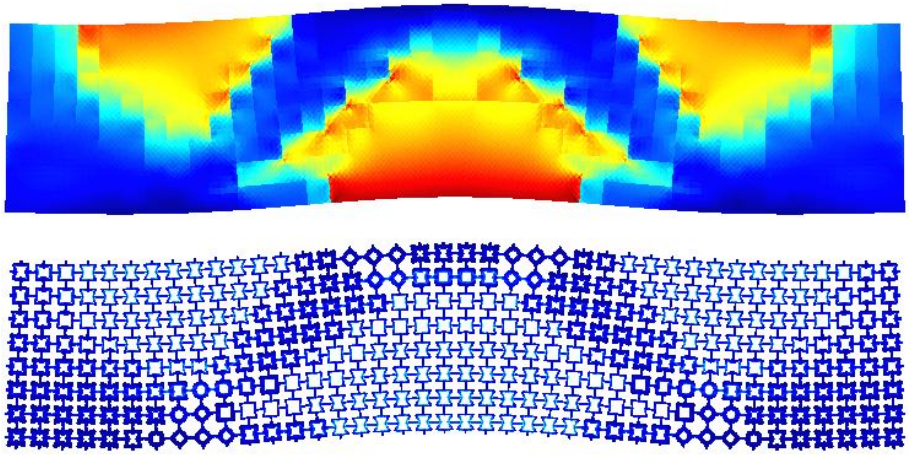


Figure 5.14: Deformation of an object with varying material properties per voxel, and the same object with the material in each voxel replaced with the corresponding pattern. The deformed objects are colored by max stress.

interpreted as a multi-objective least squares optimization to fit the displacements of two cubes—one filled with C^H and the other with C^*—under a set of axis-aligned stretching and shearing loads.

We choose this Frobenius norm as our functional:

$$J(\omega) = \frac{1}{2}\|S^H(\omega) - S^*\|_F^2, \tag{5.8}$$

which we optimize by varying the microstructure shape, ω.

In our case, the microstructure boundary $\partial\omega$ is determined by a small number of parameters $\mathbf{p}$, consisting of wire mesh node offsets and thicknesses. While the number of parameters is small, we still expect multiple solutions for minimizing $J(\omega)$ with respect to these parameters. A simple regularization term (staying close to the initial point of optimization) picks a unique solution. We note that instead some quantity of importance (e.g., weight) can be optimized as it is typically done (cf., [Sig95]), adding another nonlinear term to the functional.

The derivative of the boundary $\partial\omega$ with respect to a parameter p_α is a vector field $\mathbf{v}_{p_\alpha}(\mathbf{y})$ defined at points $\mathbf{y}$ of $\partial\omega$, with $\mathbf{v}_{p_\alpha}(\mathbf{y})$ being the velocity of $\mathbf{y}$ if parameter p_α changes at unit speed.

Using parameters $\mathbf{p}$ as variables, the minimization problem can be written as

$$\underset{\text{admissible } \mathbf{p}}{\arg\min}\ J(\mathbf{p}) \text{ where } J(\mathbf{p}) = \frac{1}{2}\|S^H(\mathbf{p}) - S^*\|_F^2. \tag{5.9}$$

The admissibility of parameters is determined by geometric intersection constraints and printability constraints.

The derivative of the objective function with respect to p_α can be obtained from $\mathbf{v}_{p_\alpha}$ using the chain rule:

$$\frac{\partial J}{\partial p_\alpha} = [S^H - S^*] : \frac{\partial S^H}{\partial p_\alpha} = [S^H - S^*] : \mathrm{d}S^H[\mathbf{v}_{p_\alpha}], \tag{5.10}$$

where $\mathrm{d}S^H[\mathbf{v}_{p_\alpha}]$ is the *shape derivative* of S^H applied to $\mathbf{v}_{p_\alpha}$.

Shape derivative of elasticity tensor The derivative of the microstructure's homogenized elasticity tensor in the direction of shape perturbation $\mathbf{v}$ is defined as the Gâteaux derivative,

$$\mathrm{d}C^H[\mathbf{v}] := \lim_{t \to 0} \frac{C^H(\omega(t, \mathbf{v})) - C^H(\omega)}{t}, \tag{5.11}$$

where $\omega(t, \mathbf{v}) := \{\mathbf{x} + t\mathbf{v} : \mathbf{x} \in \omega\}$. As shown in the additional material, the homogenized elasticity tensor, (5.7), can be rewritten in an energy-like form,

$$C^H_{ijkl} = \frac{1}{|Y|} \int_\omega (e^{ij} + \varepsilon(\mathbf{w}^{ij})) : C^{\text{base}} : (e^{kl} + \varepsilon(\mathbf{w}^{kl}))\, \mathrm{d}\mathbf{y}, \tag{5.12}$$

which is shown to have shape derivative:

$$\begin{aligned} \mathrm{d}C^H_{ijkl}[\mathbf{v}] = \frac{1}{|Y|} \int_{\partial\omega} [(e^{ij} + \varepsilon(\mathbf{w}^{ij})) : C^{\text{base}} \\ : (e^{kl} + \varepsilon(\mathbf{w}^{kl}))](\mathbf{v} \cdot \hat{\mathbf{n}}) \mathrm{d}A(\mathbf{y}). \end{aligned} \tag{5.13}$$

Shape derivative of compliance tensor. The compliance tensor is the symmetric rank 4 inverse of elasticity tensor, i.e. $S_{ijkl}C_{klmn} = \frac{1}{2}(\delta_{im}\delta_{jn} + \delta_{in}\delta_{jm})$. Differentiating and solving for $\mathrm{d}S^H$:

$$\mathrm{d}S^H[\mathbf{v}] = -S^H : \mathrm{d}C^H[\mathbf{v}] : S^H. \tag{5.14}$$

Combining the results from (5.10), (5.13), and (5.14), one can compute $\frac{\partial J}{\partial p_\alpha}$; the shape derivative and an example velocity field $\mathbf{v}_{p_\alpha}$ are shown in Figure 5.15.

Numerical computation. The integrand in (5.13) is cubic over each boundary element ($\varepsilon(\mathbf{w}^{ij})$ and $\mathbf{v} \cdot \hat{\mathbf{n}}$ are both linear), and we use quadrature that evaluates the surface integral exactly. To evaluate the gradient for a given shape we need (a) to mesh the shape (we use the TetGen package [Si10]); (b) solve 6 periodic elasticity problems to obtain $\mathbf{w}^{ij}$, as for homogenization. The cost of a single gradient evaluation (roughly 4.75s on a single core of an Intel Xeon E-2690 v2) is dominated by the cost of periodic meshing and the elasticity solves, which take roughly equal time. We use the Ceres solver [AMO]'s Levenberg-Marquardt implementation to minimize the objective; the convergence of the solver is quite fast (Figure 5.16). Typical effects of optimization are shown in Figure 5.17.

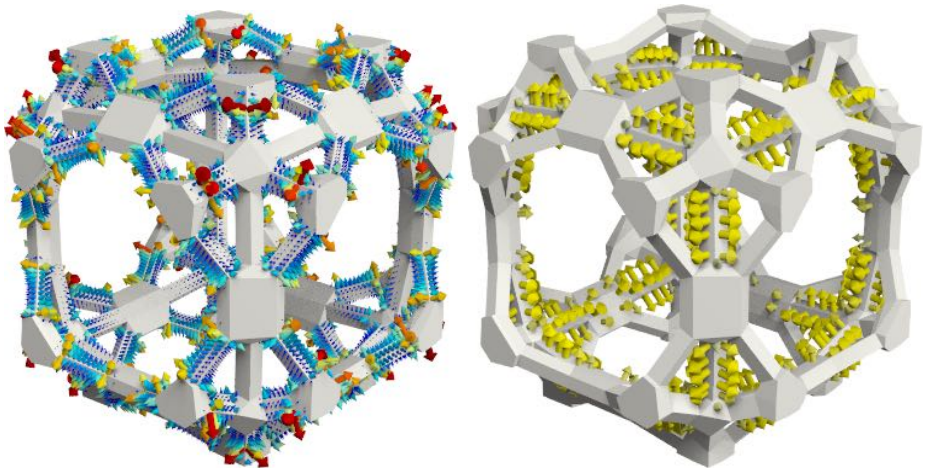


Figure 5.15: Left: a shape derivative, visualized as a steepest ascent normal velocity field for objective (5.8). Right: the shape velocity induced by one of the pattern's thickness parameters.

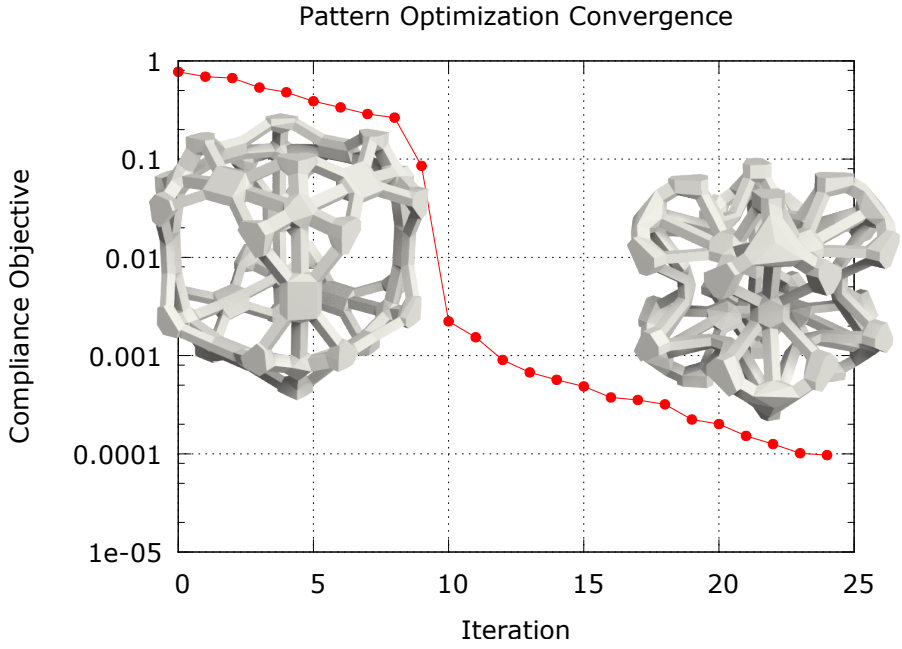


Figure 5.16: Convergence of a shape optimization on pattern "(E1,E2)(E1,E4)(E2,E4)." Left: optimization starting point. Right: optimized shape.

5.6 Applications

While our primary focus is on the design of our pattern family and the exploration of its coverage, we demonstrate the application of our elastic textures in two settings: painted material properties and specified deformation behavior. All printing was

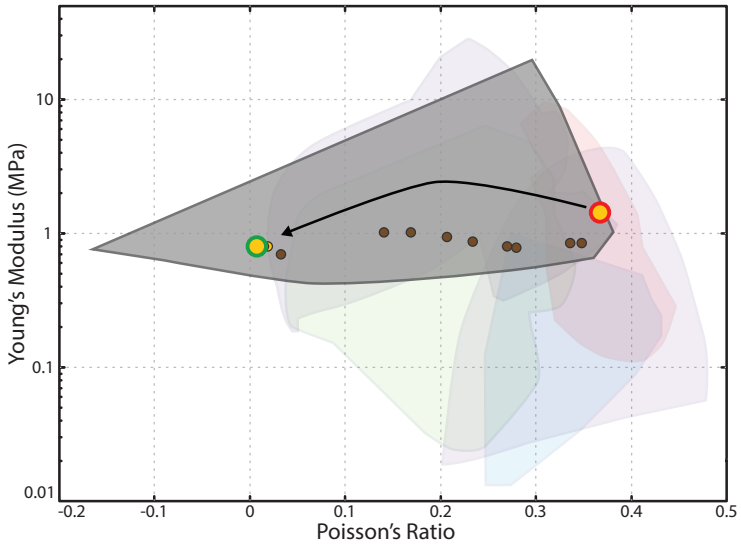


Figure 5.17: The path in $(E,\ \nu)$ space traversed by the optimization of pattern "(E1,E2)(E1,E4)(E2,E4)" shown in Figure 5.16. The brown points are intermediate anisotropic microstructures.

done using a B9Creator printer at 50 micron resolution with Cherry resin.

Overall workflow. The result of the preceding sections is a lookup map that, for a given (E, ν), produces an isotropic microstructure with nearby parameters. We assume that we are given a coarse volume mesh filled with identical cube cells.

First, we assign a pair (E_i, ν_i) to each cell i, either directly or via material optimization as described below. For each cell, we retrieve a corresponding pattern (topology id, thicknesses, and offsets) from the lookup table. The microstructures in adjacent cells are stitched together by averaging the offsets of each pair of shared face nodes so that they coincide. This might raise the lower node of the pair above some node it supports, n_s, violating printability, but printability can be restored by lowering the pair to n_s's height.

After this step, the resulting connected wire mesh is inflated with retrieved bar thicknesses, using the process described at the end of Section 5.4.This results in a fine mesh that can be printed.

Material painting. The simplest approach to specifying the material properties (E, ν) is to paint them on a voxel grid. We have created an editor enabling us to paint these layer by layer. The results of fabricating several structures of this type are shown in Figure 5.18.

Material optimization. Manually defining material properties to achieve desired behavior may be difficult, and a more systematic approach is to solve for them. For

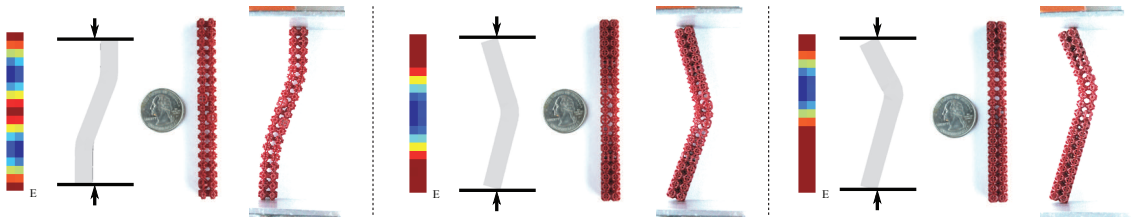


Figure 5.18: Examples of objects with painted material properties. All are fabricated with 5mm cells.

example, consider the following problem: for given applied displacement conditions on some part of the object's boundary, we would like to get some target deformation – e.g., if a bar is compressed along the Z axis, it twists in the X-Y plane – by varying material properties. In other words, we want to find a spatially varying tensor $C_p(\mathbf{x})$, parametrized by a vector of per-cell isotropic parameters p, such that the following system has a solution:

$$\begin{aligned} -\nabla \cdot (C_p : \varepsilon(\mathbf{u})) &= 0 \quad \text{in } \Omega \\ \mathbf{u} &= \mathbf{u}_{in} \quad \text{on } \Gamma_{in} \\ \mathbf{u} &= \mathbf{u}_{trg} \quad \text{on } \Gamma_{trg} \\ \sigma(\mathbf{u})\hat{\mathbf{n}} &= 0 \quad \text{on } \Gamma_{trg} \end{aligned} \tag{5.15}$$

where $\mathbf{u}_{in}$ are the applied displacements on compressed area Γ_{in}, and $\mathbf{u}_{trg}$ are the target displacements of the surface Γ_{trg} on which no forces are applied, as indicated by the last equation. In our implementation, these target/applied conditions can be specified on a per-component basis to set up, e.g., the twisting bar example.

In general, such problems are solved using PDE constrained optimization, requiring solving an adjoint problem at each iteration. However, we found that the following local-global iteration, inspired by related "as-rigid-as-possible" (ARAP) optimization techniques in geometry, works remarkably well. For a fixed $C_p(\mathbf{x})$, we call $\mathbf{u}_D$ the solution of the "Dirichlet problem", with the condition of zero tractions on Γ_{trg} removed. $\mathbf{u}_N$ is the solution of the "Neumann problem," in which the traction condition on Γ_{trg} is retained, but the Dirichlet condition $\mathbf{u} = \mathbf{u}_{trg}$ is removed.

We initialize the elastic tensor $C_p(\mathbf{x})$ to a constant. The iteration consists of two steps:

1. Solve the Dirichlet and Neumann problems with the current elasticity tensor, to obtain $\mathbf{u}_D$ and $\mathbf{u}_N$.

2. Update C_p, minimizing the following energy:

$$\min_p \int_\Omega \|\varepsilon(\mathbf{u}_D) - C_p^{-1} : \sigma(\mathbf{u}_N)\|_F^2 \, \mathrm{d}V \tag{5.16}$$

This energy can be minimized per-cell for a truly local-global method; however, in practice we find it desirable to regularize p with a Laplacian term, which requires that the "local" step be replaced by a global, but still quadratic optimization. The convergence of this method is shown in Figure 5.19.

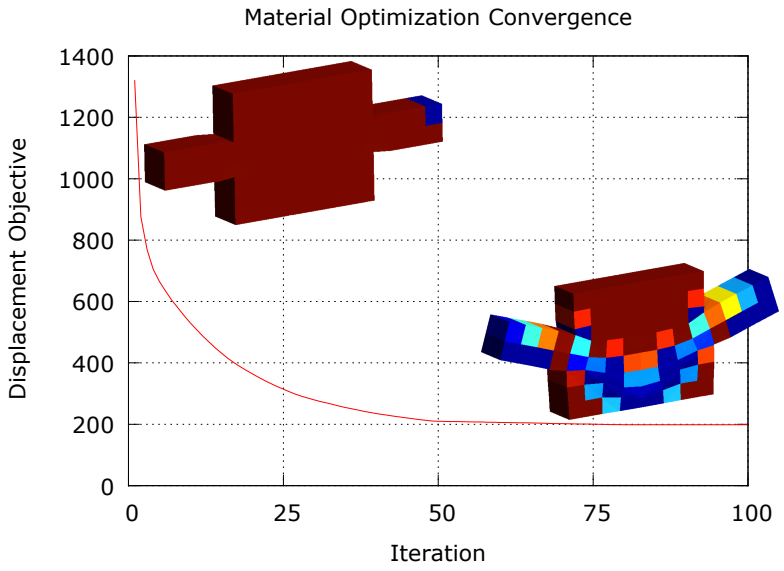


Figure 5.19: Convergence of material optimization.

We have used a number of simple voxelized shapes and created a variety of deformation behaviors shown in Figure 5.20. Finally, we have also generated a set of anisotropic samples, with controlled anisotropy ratio, one of which is shown in Figure 5.21.

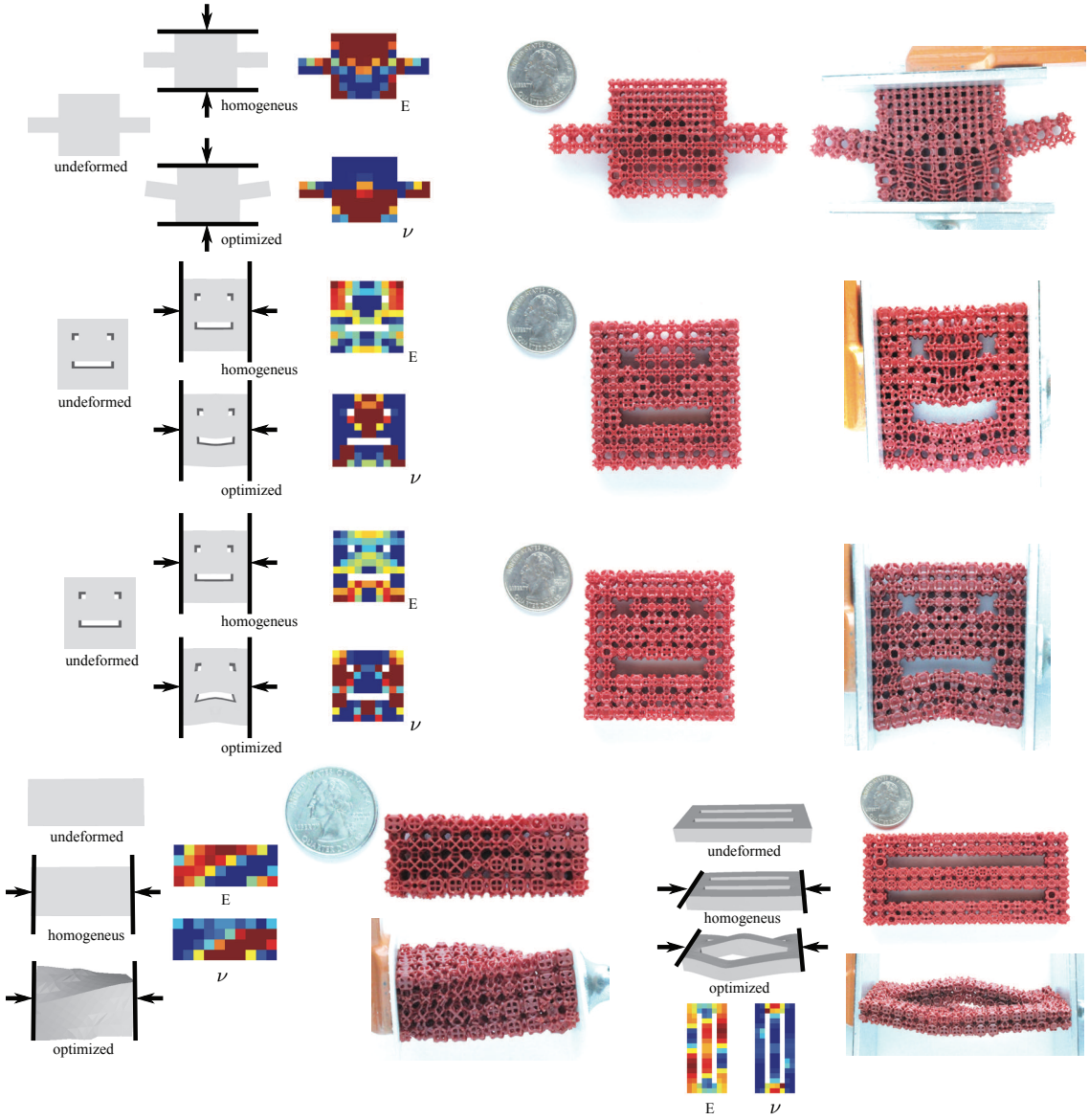


Figure 5.20: Examples of objects with optimized material properties. All are fabricated with 5mm cells.

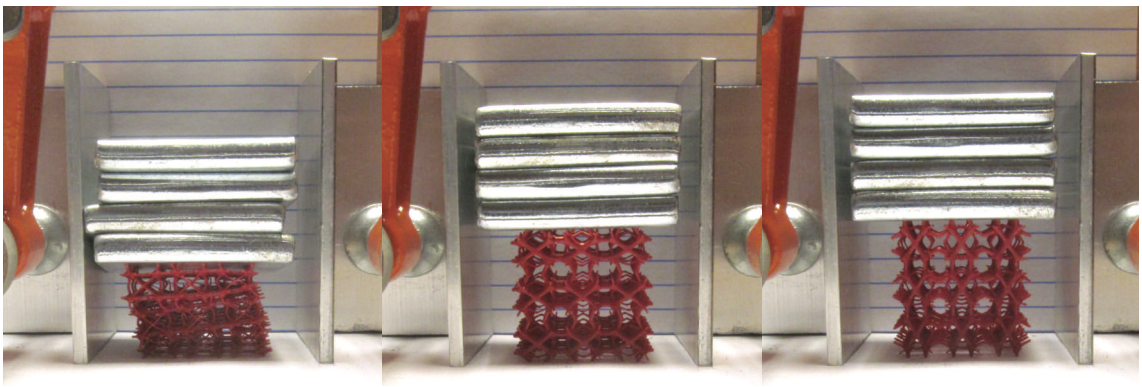


Figure 5.21: Compression of an anisotropic sample along the X, Y, and Z directions.

Chapter 6

Conclusions

In this thesis we have presented three innovative approaches to actually produce real objects that are based on a thorough analysis of the inherent problems from a geometry processing point of view. As a result we are able to provide novel, cost-effective and very expressive ways of producing objects using the already available digital fabrication technologies. Our techniques, by design, are intended for empowering final users with new digital fabrication possibilities. In fact, the common objective of each of our technique, was to reduce the production cost to allow the final users to exploit our methods, even in a domestic environment.

3D printing is a powerful enabling technology but, currently, its impact on the actual production pipelines is limited because additive manufacturing is used only when the results to accomplish make the use of this technology cost-efficient. For example, 3D printing falls short for producing large objects. In this thesis we have faced many of the technical engineering limitations from a geometric point of view and found novel solutions for them. To fulfill our goal, in particular to keep the cost low, we proposed a method that, by analyzing the shape and exploiting the properties of cross-fields, reproduces large approximated objects by sacrificing accuracy and using cheaper laser cutting technology. Instead, with flexible molds shells, by analyzing the geometric properties of cuts and simulating the mold extraction process, we allowed to produce limited series of objects bridging the gap between mass production and custom unique design. We mixed 3D printing and traditional (manual) manufacturing techniques and offer a solution to automatically generate just one 3D printable mold that can be used to cost-efficiently produce multiple copies of an input 3D model, exploiting cheaper mold-casting techniques. In the case of the elastic volumetric texture method, we fully exploited the capability of 3D printing to provide a technique that results cost-efficient using this technology only. By a focused combinatorial search in the space of the possible patterns, considering both topology and geometry, we devised a set of microstructures that provide variable elastic properties. After arranging them into the volume of an object, this can be directly 3D printed without the use of support structures.

All of our techniques are valuable examples of how methods and algorithms commonly employed in the computer science and the computer graphics domains, when combined with physics, mathematics and engineering approaches, could provide innovative solutions for tangible applications.

In the following sections we recap the contributions of this thesis, analyzing limitations and tracing possible improvements for future work.

6.1 Field-aligned Mesh Joinery

In Chapter 3 we have proposed a novel method that, starting from a given 3D mesh and a cross-field computed on it, automatically fabricates an illustrative representation of the input, 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 when slices are sampled uniformly over the surface. In addition, due to the slice decomposition, mesh joinery is suitable for the production of medium- to large-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.

Limitations Although the range of shapes that we can efficiently approximate is wide, our method suffers from minor limitations. Our technique uses shallow ribbon-shaped slices; this choice allows to avoid considering complex intersection constraints of fully planar section of the model. Unfortunately, when the depth of such slices exceeds the available space inside the model volume we fall into the previous case. For example, an extruded slice passing through a low-thickness part of the model becomes a fully planar section and we cannot handle additional slices intersecting it in this "full depth" region. Related to this problem, when the sampling resolution is too coarse, we may end up with thin object parts sampled with just one slice. Even if in the produced models we have never experienced such issues, it can happen that such a thin part connects two large portions of the object, creating a robustness issue.

In our approach 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 Chapter 3.

6.2 FlexMolds

In Chaper 4 we have presented FlexMolds, a method for computing flexible molds that allow the reproduction of complex shapes. The core of our system is an automatic method for generating feasible cut layouts for molds, building on a physics-based approach for simulating and evaluating the parting of the mold from the cast object. Our tests have shown that our method can handle geometrically difficult cases, such as models with high genus and intricate surface details. The flexible molds can be easily fabricated using 3D printing technology and have proven to be robust enough for multiple uses. We believe our system provides an exciting new way of bringing virtual models to the real world as it empowers common users to cast objects with a wide range of resins and provides an attractive option, filling a gap between mass production and direct 3D printing. As demonstrated in the results section, we validated our method by successfully reproducing several of the most popular 3D models in the Computer Graphics community.

Limitations Our method has some limitations and exciting opportunities for future work remain. Currently, closing the mold watertight requires sealing the seams with a removable silicone. We believe this problem could be solved by designing a more sophisticated interlocking mechanism, similar, for example, to the click-lock mechanism used for flooring.

We also checked the robustness of our method when applied to models with extremely complex topology. We successfully generated a mold for the heptoroid model (Figure 6.1, top), which is composed of a single piece and allows for extraction. Unfortunately, the resulting cut is extremely long and complex, which makes it difficult to seal and practically unfeasible for casting. There are some other cases in which the algorithm produces a small number of molds. This does not represent a problem as each mold piece, by construction, can be extracted independently. From our experience we know that the dominant cost, which is the objective that we minimize, is the length of the cuts and thus the time required to close them. Having multiple mold pieces does not affect substantially this process, while it represents a problem when this decomposition determines a large number of cuts. Unavoidably, models with high genus will generate complex cuts. The hollowed sphere model shown in Figure 6.1 (bottom) generates a FlexMold composed of a large piece for the external part and multiple small pieces for the internal part. As the internal portion of the mold must pass through one of the small holes to be extracted, aggregating pieces makes this process more difficult or even impossible. Hence, our system correctly leaves the internal region decomposed in multiple small patches.

Overall, the behavior of our algorithm is justified even for these complex examples; however, in those cases, the resulting molds are practically useless.

A reasonable strategy to overcome these limitations may consist in decomposing the initial pieces in multiple partitions that can be cast singularly and merged together once fabricated. For example, the hollowed sphere can be split in two pieces along the great circle, then each piece can be easily manufactured with our method.

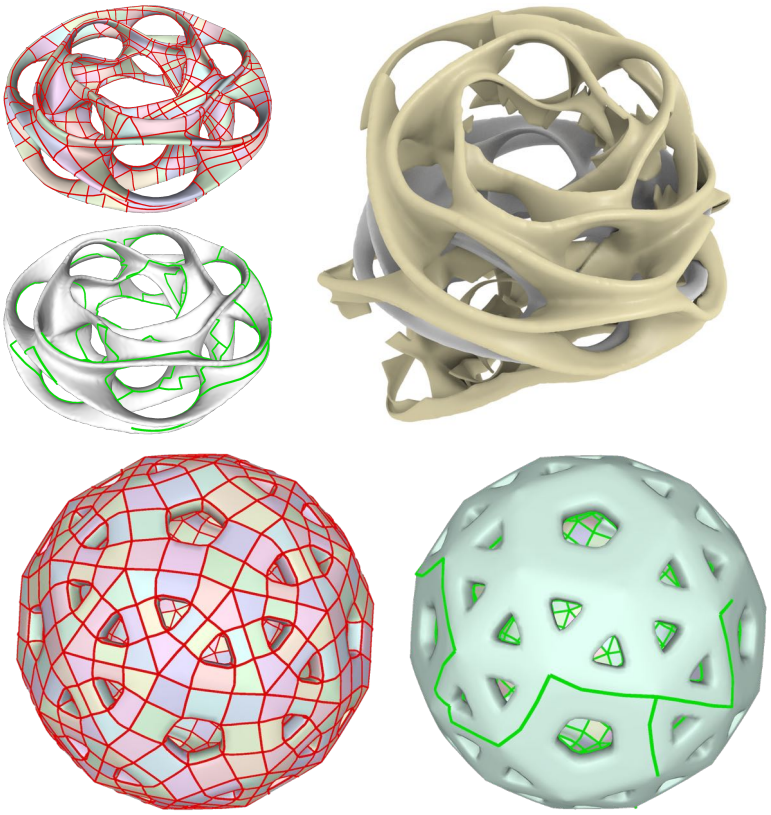


Figure 6.1: Deriving the cut for two topologically complex examples, the heptoroid model (top) and the hollowed sphere (bottom).

Currently, we do not explicitly take the static stability of the shell into account. While in our experiments for all of our models the shell has proven to be sufficiently stable when filled with resin, large pressure could lead to deformations. One could counterbalance these effects by either taking the pressure into account and computing an optimized rest shape, similar to [STBG12], or creating a mold with spatially varying thickness to provide additional stability if required. Finally, currently all our models are solidly filled. For future work, it will be interesting to provide strategies to create hollow shapes, making the objects more lightweight and saving material.

6.3 Elastic Textures

In Chapter 5 we introduced *elastic textures*: a set of parametric, tileable, printable, truss-like patterns that, when subject to pressure and forces deforms in a predictable way, so that an object tiled with such structures exhibits a broad range of isotropic elastic material properties: the softest pattern is over a thousand times softer than the stiffest, and the Poisson's ratios range from below zero to nearly 0.5. Using a combinatorial search over topologies followed by shape optimization, we explored a wide space of truss-like, symmetric 3D patterns to obtain a small family of parametrically-controllable shapes. This pattern family can be printed without internal support structure on a single-material 3D printer and can be used to fabricate objects with prescribed mechanical behavior. We demonstrated that our elastic textures are able to achieve a user-supplied varying material property distribution. We also present a material optimization algorithm to choose material properties at each point within an object to best fit a target deformation under a prescribed scenario. We show that, by fabricating these spatially varying materials with elastic textures, the desired behavior is achieved. The chosen pattern family has proved useful on a number of simple shape optimization examples: remarkably, all examples in Section 5.6 worked as predicted by simulation without requiring much tuning.

Limitations There are several limitations of our pattern family. First, some parts of the (E, ν) space are poorly covered. While it is difficult to predict which part of space is theoretically reachable, we conjecture that the space may be significantly broadened. All our simulations and constructions work in the linear regime, not taking into account, e.g., the potential for pattern buckling or other damage. Fortunately, isotropy is correlated with sufficiently high shear modulus, which makes the patterns less prone to buckling. Nevertheless, including this and other nonlinear effects in pattern design is important.

For practical use, it is difficult to restrict the tessellations of objects to equal sized cubes (though one can construct cut cells covered with relatively soft skin). A desirable solution would be to allow patterns to distort to fill arbitrary reasonably well-shaped hexahedral cells.

6.4 Final Remarks

From a technical point of view we have shown the multi-valence purposes of shape modeling, in particular when applied to specific target application. As recalled in the introduction, this thesis is yet another proof that applications of many different techniques, traditionally used to translate real life phenomena into digital form or employed into digital-only processing, can offer valuable tools for fabricating real objects.

Referring specifically to the contributions of Chapters 3, 4 and 5, it is interesting to point out some of the common techniques used to carry out the research. In Chapter 3 we have shown how curvature-aligned cross-fields, generated on mesh surfaces, serve to guide the sampling strategy of slices approximating a model. With respect to other sampling techniques, the usage of curvature aligned pieces greatly improves the visual understanding of the fabricated objects. It is fascinating how this simple expedient, that starts from geometric understanding of the shape, has been used for illustrative purpose in non photorealistic rendering techniques (e.g., [HZ00]) and can be straightly applied for the actual fabrication of robust piece-interlocking objects. In Chapter 4, instead, curvature-aligned cross-fields are used to generate patches with curvature-aligned boundaries. Interestingly, these boundaries strongly resembles to the way a human would cut a mold shell wrapping an object and, as demonstrated, offer a very good way of limiting the space of possibile cuts required to fabricate a flexible mold.
Another example of different usage of similar techniques is demonstrated by physical simulation in Chapters 4 and 5. In the former chapter we have used fast, approximated physical simulation, commonly employed in interactive applications, to heuristically simulate the mold extraction process. This analysis is continuously employed to guide the optimization procedure that selects a valid set of cuts for the mold design. Again, much more accurate physical simulation has been used in the technique of Chapter 5: linear finite element analysis has been employed to precisely describe the elastic behavior of a huge amount of microstructures. Luckily enough this has been carried out as a pre-computation step and thus the use of slower and more accurate simulation is justified.

In general, relying on previous knowledge and techniques is the foundation of digital fabrication but it is very hard to predict what future research will bring. As last decade has taught us, it has always been a surprise the kind of ideas and applications that are being introduced. From the recent past it is known that the domain, especially within the computer graphics community, is gaining more and more interest and for sure in the next years we foresee that the trend will be the same and the field will broaden its scope.

Concerning this thesis, we have provided three novel contributions to the field that could serve as inspiration or technical base for future research. In Chapter 3 we have presented a complete pipeline for producing an illustrative target shape made of interlocked planar pieces. Together with concurrent works the possibility of easily realizing such structures has already been recognized and improved. For example, introducing physical simulation for interlocking structures, can provide a functional value to these objects, as demonstrated by the work in [MUS14].
Even if published only recently, the FlexMolds approach introduced in Chapter 4 could hint future works in the usage of modern technologies mixed with traditional manufacturing. Also, from a technical point of view, the optimization algorithm employed demonstrated that heuristics based on physical simulation can be successfully used to guide the design of functional-specific designs.

The results obtained in Chapter 5 demonstrated a completely new usage of additive manufacturing technologies. It is still early to predict the potential of customizing the elastic behavior of objects with such granularity. Traditionally, from an engineering point of view, the design of objects is carried out keeping in consideration a discrete set of materials that can be employed, therefore the possibility of using spatially varying elastic properties inside the volume is still a relatively unexplored field. The test examples we have shown are just a proof of concept of what can be achieved. One of the many application that comes to mind is to extend the approach to allow the creation of custom objects designed to absorb shock and distribute it in a controlled way. In the short term I think that we will see more breaking-through new applications based on this work.

Bibliography

[3D 16] 3D Systems. 3d printers. More info on: `http://www.3dsystems.com/`, 2016. 1

[All02] Grégoire Allaire. *Shape optimization by the homogenization method*, volume 146. Springer Science & Business Media, 2002. 91, 92

[ALS14] Erik Andreassen, Boyan S. Lazarov, and Ole Sigmund. Design of manufacturable 3D extremal elastic microstructure. *Mechanics of Materials*, 69(1):1–10, 2014. 92

[AMO] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`. 108

[ASH15] Sami Arpa, Sabine Ssstrunk, and Roger D. Hersch. High reliefs from 3d scenes. *Computer Graphics Forum*, 34(2):253–263, 2015. 32

[Aut13] Autodesk. 123D make. `http://www.123dapp.com/make/`, 2013. 2

[Ave87] M Avellaneda. Optimal bounds and microgeometries for elastic two-phase composites. *SIAM Journal on Applied Mathematics*, 47(6):1216–1228, 1987. 91

[BAU15] Alan Brunton, Can Ates Arikan, and Philipp Urban. Pushing the limits of 3d color printing: Error diffusion with translucent materials. *ACM Trans. Graph.*, 35(1):4:1–4:13, December 2015. xi, 14

[BBAM12] Amit Bermano, Ilya Baran, Marc Alexa, and Wojciech Matusk. Shadowpix: Multiple images from self shadowing. *Computer Graphics Forum*, 31(2pt3):593–602, 2012. 35

[BBJP12] Moritz Bächer, Bernd Bickel, Doug L. James, and Hanspeter Pfister. Fabricating articulated characters from skinned meshes. *ACM Trans. Graph.*, 31(4):47:1–47:9, July 2012. 19

[BBO+10] Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. Design

and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.*, 29(4):63:1–63:10, July 2010. 20, 21, 92

[Ben89] Martin Philip Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1(4):193–202, 1989. 91

[BH11] Zhe Bian and Shi-Min Hu. Preserving detailed features in digital bas-relief making. *Computer Aided Geometric Design*, 28(4), 2011. 31

[BHT00] S. Bobkov, C. Houdr, and P. Tetali. λ_∞,vertex isoperimetry and concentration. *Combinatorica*, 20(2):153–172, 2000. 57

[BLP+12] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin. State of the art in quad meshing. In Marie-Paule Cani and Fabio Ganovelli, editors, *EG 2012 - State of the Art Reports*, Cagliari, Sardinia, Italy, 2012. Eurographics Association. 41

[BML+14] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics (TOG)*, 33(4):154, 2014. 74

[BMOT13] Jan Bender, Matthias Müller, Miguel A Otaduy, and Matthias Teschner. Position-based methods for the simulation of solid objects in computer graphics. *EUROGRAPHICS 2013 State of the Art Reports*, 2013. 68

[BOP10] Tim Bruckner, Zach Oat, and Ruben Procopio. *Pop sculpture.* Watson-Guptill Publications, 2010. 65, 68

[BPK+11] Pengbo Bo, Helmut Pottmann, Martin Kilian, Wenping Wang, and Johannes Wallner. Circular arc structures. *ACM Trans. Graph.*, 30:#101,1–11, 2011. 19

[BS03] Martin Philip Bendsøe and Ole Sigmund. *Topology optimization: theory, methods and applications.* Springer, 2003. 91, 92

[BSK+12] Tiemo Bückmann, Nicolas Stenger, Muamer Kadic, Johannes Kaschke, Andreas Frölich, Tobias Kennerknecht, Christoph Eberl, Michael Thiel, and Martin Wegener. Tailored 3d mechanical metamaterials made by dip-in direct-laser-writing optical lithography. *Advanced Materials*, 24(20):2710–2714, 2012. 92

[BWBSH14] Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. Spin-it: optimizing moment of inertia for spinnable objects. *ACM Transactions on Graphics (TOG)*, 33(4):96, 2014. 21

[BZK09] David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3):77:1–77:10, July 2009. xii, 41, 43, 45, 69

[CBK12] Marcel Campen, David Bommes, and Leif Kobbelt. Dual loops meshing: Quality quad layouts on manifolds. *ACM Trans. Graph.*, 31(4):110:1–110:11, July 2012. 69

[CCA+12] Jacques Calì, Dan A. Calian, Cristina Amati, Rebecca Kleinberger, Anthony Steed, Jan Kautz, and Tim Weyrich. 3d-printing of non-assembly, articulated models. *ACM Trans. Graph.*, 31(6):130:1–130:8, November 2012. 19

[CD99] Doina Cioranescu and Patrizia Donato. *An introduction to homogenization*. Oxford University Press, 1999. 91

[CEGR10] Jane Chu, Sarah Engelbrecht, Gregory Graf, and David W Rosen. A comparison of synthesis methods for cellular structures with application to additive manufacturing. *Rapid Prototyping Journal*, 16(4):275–283, 2010. 91

[CGPS08] Paolo Cignoni, Enrico Gobbetti, Ruggero Pintus, and Roberto Scopigno. Color enhancement for rapid prototyping. In *Proceedings of the 9th International conference on Virtual Reality, Archaeology and Cultural Heritage*, VAST'08, pages 9–16, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association. 13

[Che00] Andrej Cherkaev. *Variational methods for structural optimization*, volume 140. Springer, 2000. 91

[CJR95] Kumar Chalasani, Larry Jones, and Larry Roscoe. Support generation for fused deposition modeling. 1995. 9

[CLD+13] Desai Chen, David I. W. Levin, Piotr Didyk, Pitchaya Sitthi-Amorn, and Wojciech Matusik. Spec2fab: A reducer-tuner model for translating specifications to 3d prints. *ACM Trans. Graph.*, 32(4):135:1–135:10, July 2013. 21, 92

[CMS97] P. Cignoni, C. Montani, and R. Scopigno. Computer-assisted generation of bas- and high-reliefs. *J. Graph. Tools*, 2(3), December 1997. xii, 31, 32

[CPMS14] Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. Field-aligned mesh joinery. *ACM Trans. Graph.*, 33(1):11:1–11:12, February 2014. 41

[CR09] Pritam Chakraborty and N Venkata Reddy. Automatic determination of parting directions, parting lines and surfaces for two-piece permanent molds. *journal of materials processing technology*, 209(5):2464–2476, 2009. 67

[CSAD04] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. *ACM Trans. Graph.*, 23(3):905–914, August 2004. xiv, xv, 70, 71, 82, 83

[CSaLM13] Desai Chen, Pitchaya Sitthi-amorn, Justin T. Lan, and Wojciech Matusik. Computing and fabricating multiplanar models. *Computer Graphics Forum*, 32(2pt3):305–315, 2013. 26, 68

[CZCL13] Joseph E Cadman, Shiwei Zhou, Yuhang Chen, and Qing Li. On design of multi-functional microstructural materials. *Journal of Materials Science*, 48(1):51–66, 2013. 91

[CZL+15] Xuelin Chen, Hao Zhang, Jinjie Lin, Ruizhen Hu, Lin Lu, Qixing Huang, Bedrich Benes, Daniel Cohen-Or, and Baoquan Chen. Dapper: decompose-and-pack for 3d printing. *ACM Transactions on Graphics (TOG)*, 34(6):213, 2015. 11, 68

[CZX+16] Weikai Chen, Xiaolong Zhang, Shiqing Xin, Yang Xia, Sylvain Lefebvre, and Wenping Wang. Synthesis of filigrees for digital fabrication. *ACM Trans. Graph.*, 35(4):98:1–98:13, July 2016. 22

[CZXZ14] Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. An asymptotic numerical method for inverse elastic shape design. *ACM Trans. Graph.*, 33(4):95:1–95:11, July 2014. 16

[Dey94] Tamal K. Dey. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, SCG '94, pages 277–284, New York, NY, USA, 1994. ACM. 70

[DGBK11] Ernest P De Garmo, J Temple Black, and Ronald A Kohser. *DeGarmo's materials and processes in manufacturing*. John Wiley & Sons, 2011. 67

[DGLC15] H. Quynh Dinh, Filipp Gelman, Sylvain Lefebvre, and Frédéric Claux. Modeling and toolpath generation for consumer-level 3d printing. In *ACM SIGGRAPH 2015 Courses*, SIGGRAPH '15, pages 17:1–17:273, New York, NY, USA, 2015. ACM. 9

[DHL14] Jérémie Dumas, Jean Hergel, and Sylvain Lefebvre. Bridging the gap: Automated steady scaffoldings for 3d printing. *ACM Trans. Graph.*, 33(4):98:1–98:10, July 2014. 11

[DLL+15] Jérémie Dumas, An Lu, Sylvain Lefebvre, Jun Wu, T. U. München, Christian Dick, and T. U. München. By-example synthesis of structurally sound patterns. *ACM Trans. Graph.*, 34(4):137:1–137:12, July 2015. 22

[DPRW00] Debasish Dutta, Fritz B. Prinz, David Rosen, and Lee Weiss. Layered manufacturing: Current status and future trends. *Journal of Computing and Information Science in Engineering*, 1(1), 2000. 9

[DPW+14] Mario Deuss, Daniele Panozzo, Emily Whiting, Yang Liu, Philippe Block, Olga Sorkine-Hornung, and Mark Pauly. Assembling self-supporting structures. *ACM Trans. Graph.*, 33(6):214:1–214:10, November 2014. xi, 18, 19

[DSdB06] D. Dimitrov, K. Schreve, and N. de Beer. Advances in three dimensional printing state of the art and future perspectives. *Rapid Prototyping Journal*, 12(3):136–147, 2006. 42

[DWP+10] Yue Dong, Jiaping Wang, Fabio Pellacini, Xin Tong, and Baining Guo. Fabricating spatially-varying subsurface scattering. *ACM Trans. Graph.*, 29:62:1–62:10, July 2010. 13

[EKS+10] Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. Paneling architectural freeform surfaces. *ACM Trans. Graph.*, 29(4):45:1–45:10, July 2010. 19

[Elb11] Gershon Elber. Modeling (seemingly) impossible models. *Computers & Graphics*, 35(3):632–638, 2011. 39

[FLHCO10] Chi-Wing Fu, Chi-Fu Lai, Ying He, and Daniel Cohen-Or. K-set tilable surfaces. *ACM Trans. Graph.*, 29(4):44:1–44:6, July 2010. 19

[GGH02] Xianfeng Gu, Steven J. Gortler, and Hugues Hoppe. Geometry images. *ACM Trans. Graph.*, 21(3):355–361, July 2002. 71

[GGLR11] G. N. Greaves, A. L. Greer, R. S. Lakes, and T. Rouxel. Poisson's ratio and modern materials. *Nature Materials*, 10(11):823–837, 2011. 92

[GHDS03] Eitan Grinspun, Anil N Hirani, Mathieu Desbrun, and Peter Schröder. Discrete shells. In *Proceedings of the 2003 ACM*

SIGGRAPH/Eurographics symposium on Computer animation, pages 62–67. Eurographics Association, 2003. 68

[GK95] Yury Grabovsky and Robert V Kohn. Microstructures minimizing the energy of a two phase elastic composite in two space dimensions. II: the Vigdergauz microstructure. *Journal of the Mechanics and Physics of Solids*, 43(6):949–972, 1995. 91

[GP06] James K. Guest and Jean H. Prévost. Optimizing multifunctional materials: Design of microstructures for maximized stiffness and fluid permeability. *International Journal of Solids and Structures*, 43(2223):7028 – 7047, 2006. 91

[GPT07] Marco Actis Grande, Lorenzo Porta, and Dario Tiberto. Computer simulation of the investment casting process: Widening of the filling step. In *The Santa Fe Symposium on Jewelry Manufacturing Technology 2007*, 2007. 68

[GRS14] Ian Gibson, David Rosen, and Brent Stucker. *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing*. Springer, 2014. 7

[GSFD+14] Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. Wire mesh design. *ACM Trans. Graph.*, 33(4), July 2014. xi, 27, 28

[Gür15] Y Gür. Digital fabrication of mathematical models via low-cost 3d fdm desktop printer. *Acta Physica Polonica A*, 128(2B), 2015. 38

[HA92] Yong K. Hwang and Narendra Ahuja. Gross motion planning—a survey. *ACM Comput. Surv.*, 24(3):219–291, September 1992. 70

[Har08] George W Hart. Sculptural forms from hyperbolic tessellations. In *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on*, pages 155–161. IEEE, 2008. 106

[HBA12] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. crdbrd: Shape fabrication by sliding planar slices. *Computer Graphics Forum*, 31(2pt3):583–592, 2012. xi, 25, 41, 43, 44, 51, 62, 68

[HBA13] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. Orthogonal slicing for additive manufacturing. *Computers & Graphics*, 37(6):669 – 675, 2013. Shape Modeling International (SMI) Conference 2013. 13

[HBLM11] Michael Holroyd, Ilya Baran, Jason Lawrence, and Wojciech Matusik. Computing and fabricating multilayer models. *ACM Trans. Graph.*, 30:187:1–187:8, December 2011. xii, 34

[HFM+10] Miloš Hašan, Martin Fuchs, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Physical reproduction of materials with specified subsurface scattering. *ACM Trans. Graph.*, 29(4):61:1–61:10, July 2010. 13, 21

[HIH+13] Matthias B. Hullin, Ivo Ihrke, Wolfgang Heidrich, Tim Weyrich, Gerwin Damberg, and Martin Fuchs. Computational Fabrication and Display of Material Appearance. In M. Sbert and L. Szirmay-Kalos, editors, *Eurographics 2013 - State of the Art Reports*. The Eurographics Association, 2013. 15, 33, 34

[HL09] Jonathan Hiller and Hod Lipson. Design and analysis of digital materials for physical 3d voxel printing. *Rapid Prototyping Journal*, 15(2):137–149, 2009. 92

[HL14] Jean Hergel and Sylvain Lefebvre. Clean color: Improving multi-filament 3d prints. *Comput. Graph. Forum*, 33(2):469–478, May 2014. 10

[HLZCO14] Ruizhen Hu, Honghua Li, Hao Zhang, and Daniel Cohen-Or. Approximate pyramidal shape decomposition. *ACM Trans. Graph.*, 33(6):213:1–213:12, November 2014. 11, 67

[HMA15] Philipp Herholz, Wojciech Matusik, and Marc Alexa. Approximating Free-form Geometry with Height Fields for Manufacturing. *Computer Graphics Forum (Proc. of Eurographics)*, 34(2):239–251, 2015. 67

[Hol05] Scott J. Hollister. Porous scaffold design for tissue engineering. *Nature Materials*, 4(7):518–524, 2005. 92

[Hud14] Scott E. Hudson. Printing teddy bears: A technique for 3d printing of soft interactive objects. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '14, New York, NY, USA, 2014. ACM. 30

[HWS+16] Jhen-Yao Hong, Der-Lor Way, Zen-Chung Shih, Wen-Kai Tai, and Chin-Chen Chang. Inner engraving for the creation of a balanced lego sculpture. *The Visual Computer*, 32(5):569–578, 2016. 37

[HZ00] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 517–526, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 43, 120

[IIM12] Yuki Igarashi, Takeo Igarashi, and Jun Mitani. Beady: Interactive beadwork design and construction. *ACM Trans. Graph.*, 31(4):49:1–49:9, July 2012. 29

[ILB15] Emmanuel Iarussi, Wilmot Li, and Adrien Bousseau. Wrapit: Computer-assisted crafting of wire wrapped jewelry. *ACM Trans. Graph.*, 34(6):221:1–221:8, October 2015. xi, 28

[Jim12] P Jiménez. Survey on model-based manipulation planning of deformable objects. *Robotics and computer-integrated manufacturing*, 28(2):154–163, 2012. 68

[JKS05] Dan Julius, Vladislav Kraevoy, and Alla Sheffer. D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum*, 24(3):581–590, 2005. 68

[JMS14] Zhongping Ji, Weiyin Ma, and Xianfang Sun. Bas-relief modeling from normal images with intuitive styles. *IEEE Trans. Vis. Comput. Graph*, 20(5), 2014. 31

[Joh13] Angus Johnson. Clipper library 5.1.6 - an open source freeware polygon clipping library. `http://www.angusj.com/delphi/clipper.php`, 2013. 45

[Kan10] Hee Suk Kang. *Hierarchical design and simulation of tissue engineering scaffold mechanical, mass transport, and degradation properties*. PhD thesis, The University of Michigan, 2010. 92

[KEN+13] Thomas Kiser, Michael Eigensatz, Minh Man Nguyen, Philippe Bompas, and Mark Pauly. *Architectural Caustics — Controlling Light with Geometry*, pages 91–106. Springer Vienna, Vienna, 2013. 34

[KISS15] Benjamin Keinert, Matthias Innmann, Michael Sänger, and Marc Stamminger. Spherical fibonacci mapping. *ACM Trans. Graph.*, 34(6):193:1–193:7, October 2015. 80

[KLC+15] Ming-Hsun Kuo, You-En Lin, Hung-Kuo Chu, Ruen-Rone Lee, and Yong-Liang Yang. Pixel2brick: Constructing brick sculptures from pixel art. *Computer Graphics Forum*, 34(7):339–348, 2015. xii, 37

[KMOD09] Lily Kharevych, Patrick Mullen, Houman Owhadi, and Mathieu Desbrun. Numerical coarsening of inhomogeneous elastic materials. *ACM Trans. Graph.*, 28(3):51:1–51:8, July 2009. 92

[KNP07] Felix Kälberer, Matthias Nieser, and Konrad Polthier. Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum*, 26(3):375–384, 2007. 41

[KS13] Oliver Knill and Elizabeth Slavkovsky. Illustrating mathematics using 3d printers. *arXiv preprint arXiv:1306.5599*, 2013. 38

[KTB+09] Jens Kerber, Art Tevs, Alexander G. Belyaev, Rhaleb Zayer, and Hans-Peter Seidel. Feature sensitive bas relief generation. In *IEEE International Conf. on Shape Modeling and Applications*. IEEE Computer Society, 2009. 31

[KTM16] Takuya Kozaki, Hiroshi Tedenuma, and Takashi Maekawa. Automatic generation of lego building instructions from multiple photographic images of real objects. *Comput. Aided Des.*, 70(C):13–22, January 2016. 37

[KWC+12] J. Kerber, M. Wang, J. Chang, J. J. Zhang, A. Belyaev, and H.-P. Seidel. Computer assisted relief generation–a survey. *Comput. Graph. Forum*, 31(8):2363–2377, December 2012. 31

[LBRM12] Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, and Wojciech Matusik. Chopper: partitioning models into 3d-printable parts. *ACM Trans. Graph.*, 31(6):129, 2012. xi, 11, 12, 68

[Lév14] Bruno Lévy. Restricted voronoi diagrams for (re)-meshing surfaces and volumes. In *8th International Conference on Curves and Surfaces*, volume 6, page 14, 2014. xv, 82, 83

[LFL09] Kui-Yip Lo, Chi-Wing Fu, and Hongwei Li. 3d polyomino puzzle. *ACM Trans. Graph.*, 28(5):157:1–157:8, December 2009. xii, 36

[LHCH04] Chia-Ying Lin, Chun-Ching Hsiao, Po-Quan Chen, and Scott J Hollister. Interbody fusion cage design using integrated global layout and local microstructure topology optimization. *Spine*, 29(16):1747–1754, 2004. PMID: 15303018. 92

[LJGH11] Xian-Ying Li, Tao Ju, Yan Gu, and Shi-Min Hu. A geometric study of v-style pop-ups: theories and algorithms. *ACM Trans. Graph.*, 30(4):98:1–98:10, July 2011. 24

[LJL07] Liping Liu, Richard D James, and Perry H Leo. Periodic inclusionatrix microstructures with constant field inclusions. *Metallurgical and Materials Transactions A*, 38(4):781–787, 2007. 91

[LKH04] Cheng Yu Lin, Noboru Kikuchi, and Scott J. Hollister. A novel method for biomaterial scaffold internal architecture design to match bone elastic properties with desired porosity. *Journal of Biomechanics*, 37(5):623–636, 2004. 92

[LPRM02] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, July 2002. 23

[LQ14] Alan C Lin and Nguyen Huu Quang. Automatic generation of mold-piece regions and parting curves for complex cad models in multi-piece mold design. *Computer-Aided Design*, 57:15–28, 2014. 67

[LSH+10] Xian-Ying Li, Chao-Hui Shen, Shi-Sheng Huang, Tao Ju, and Shi-Min Hu. Popup: automatic paper architectures from 3d models. *ACM Trans. Graph.*, 29(4):111:1–111:9, July 2010. xi, 24, 42

[LSZ+14] Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. Build-to-last: Strength to weight 3d printed objects. *ACM Trans. Graph.*, 33(4):97:1–97:10, July 2014. 17

[LWSW14] Ligang Liu, Charlie Wang, Ariel Shamir, and Emily Whiting. 3d printing oriented design: geometry and optimization. In *SIGGRAPH Asia 2014 Courses*, page 1. ACM, 2014. 7, 66

[LYH+15] Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. Legolization: Optimizing lego designs. *ACM Trans. Graph.*, 34(6):222:1–222:12, October 2015. xii, 37

[MAN+16] James McCann, Lea Albaugh, Vidya Narayanan, April Grow, Wojciech Matusik, Jennifer Mankoff, and Jessica Hodgins. A compiler for 3d machine knitting. *ACM Trans. Graph.*, 35(4):49:1–49:11, July 2016. 30

[MDL16] Jonàs Martínez, Jérémie Dumas, and Sylvain Lefebvre. Procedural voronoi foams for additive manufacturing. *ACM Trans. Graph.*, 35(4):44:1–44:12, July 2016. 20

[MGE07] Fady Massarwi, Craig Gotsman, and Gershon Elber. Papercraft models using generalized cylinders. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 148–157, Washington, DC, USA, 2007. IEEE Computer Society. 23, 42

[MHHR07] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007. 68

[MI06] Yuki Mori and Takeo Igarashi. Pillow: interactive pattern design for stuffed animals. In *ACM SIGGRAPH 2006 Sketches*. ACM, 2006. 29

[MI07] Yuki Mori and Takeo Igarashi. Plushie: an interactive design system for plush toys. *ACM Trans. Graph.*, 26:45:1–45:8, July 2007. xi, 29, 42

[MIG+14] Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. Wireprint: 3d printed previews for fast prototyping. In *Proc. of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, New York, NY, USA, 2014. ACM. 27

[Mil02] Graeme W Milton. *The theory of composites*. Cambridge University Press, 2002. 91

[MK13] Kristo Mela and Juhani Koski. Distributed loads in truss topology optimization. In *Proceedings of the 10th world congress on structural and multidisciplinary optimization, Orlando*, 2013. 103

[MKD99] Ka Mani, Prashant Kulkarni, and Debasish Dutta. Region-based adaptive slicing. *Computer-Aided Design*, 31(5):317–333, 1999. 9

[MLB16] Eder Miguel, Mathias Lepoutre, and Bernd Bickel. Computational design of stable planar-rod structures. *ACM Trans. Graph.*, 35(4):86:1–86:11, July 2016. 28

[MP09] N. J. Mitra and M. Pauly. Shadow art. *ACM Transactions on Graphics*, 28(5), 2009. to appear. xii, 35

[MPBC16] Luigi Malomo, Nico Pietroni, Bernd Bickel, and Paolo Cignoni. Flexmolds: Automatic design of flexible shells for molding. *ACM Trans. Graph.*, 35(6):223:1–223:12, November 2016. 65

[MPVRD03] Pulak Mohan Pandey, N. Venkata Reddy, and Sanjay G. Dhande. Slicing procedures in layered manufacturing: a review. *Rapid Prototyping Journal*, 9(5):274–288, 2003. 9

[MPZ14] Ashish Myles, Nico Pietroni, and Denis Zorin. Robust field-aligned global parametrization. *ACM Trans. on Graphics - Siggraph 2014*, 33(4):Article No. 135, 2014. 69

[MS00] J. Michael McCarthy and Gim Song Soh. *Geometric design of linkages*, volume 11. Springer, Berlin, 2000. 47

[MS04a] J. Mitani and H. Suzuki. Computer aided design for origamic architecture models with polygonal representation. In *Computer Graphics International, 2004. Proceedings*, pages 93–99, June 2004. 24

[MS04b] Jun Mitani and Hiromasa Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph.*, 23(3):259–263, August 2004. xi, 22, 23, 42

[MSJT08] Matthias Müller, Jos Stam, Doug James, and Nils Thürey. Real time physics: class notes. In *ACM SIGGRAPH 2008 classes*, page 88. ACM, 2008. 68

[MSM11] James McCrae, Karan Singh, and Niloy J. Mitra. Slices: a shape-proxy based on planar sections. *ACM Trans. Graph.*, 30(6):168:1–168:12, December 2011. 25, 41, 43, 44

[MUS14] James McCrae, Nobuyuki Umetani, and Karan Singh. Flatfitfab: Interactive modeling with planar sections. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 13–22, New York, NY, USA, 2014. ACM. 26, 120

[MVK+09] Vladimir Mironov, Richard P Visconti, Vladimir Kasyanov, Gabor Forgacs, Christopher J Drake, and Roger R Markwald. Organ printing: tissue spheroids as building blocks. *Biomaterials*, 30(12):2164–2174, 2009. 92

[NR13] M Neumüller and A Reichinger. From stereoscopy to tactile photography. *PhotoResearcher*, 19, April 2013. 32

[NS10] PH Nakasone and ECN Silva. Dynamic design of piezoelectric laminated sensors and actuators using topology optimization. *Journal of Intelligent Material Systems and Structures*, 21(16):1627–1652, 2010. 91

[PB07] Irene Polo-Blanco. *Theory and history of geometric models*. PhD thesis, University of Groningen, 2007. 38

[PBSH13] Daniele Panozzo, Philippe Block, and Olga Sorkine-Hornung. Designing unreinforced masonry models. *ACM Trans. Graph.*, 32(4):91:1–91:12, July 2013. xi, 18, 19

[PDP+15] Daniele Panozzo, Olga Diamanti, Sylvain Paris, Marco Tarini, Evgeni Sorkine, and Olga Sorkine-Hornung. Texture mapping real-world objects with hydrographics. *Comput. Graph. Forum*, 34(5):65–75, August 2015. 14

[PHD+10] Helmut Pottmann, Qixing Huang, Bailin Deng, Alexander Schiftner, Martin Kilian, Leonidas Guibas, and Johannes Wallner. Geodesic patterns. *ACM Trans. Graph.*, 29(4):43:1–43:10, July 2010. 19

[PJJ+11] Marios Papas, Wojciech Jarosz, Wenzel Jakob, Szymon Rusinkiewicz, Wojciech Matusik, and Tim Weyrich. Goal-based Caustics. *Computer Graphics Forum*, 2011. 34

[PLPZ12] Daniele Panozzo, Yaron Lipman, Enrico Puppo, and Denis Zorin. Fields on symmetric surfaces. *ACM Trans. Graph.*, 31(4):111:1–111:12, July 2012. xii, 18, 43, 45

[Pon16] 3d printing and laser cutting made easy. `http://www.ponoko.com/`, 2016. 2

[PPM+16] Nico Pietroni, Enrico Puppo, Giorgio Marcias, Roberto Scopigno, and Paolo Cignoni. Tracing field-coherent quad layouts. *Computer Graphics Forum (Proceedings of Pacific Graphics 2016)*, 2016. xv, 73, 82, 83, 87

[PRM14] Thiago Pereira, Szymon Rusinkiewicz, and Wojciech Matusik. Computational light routing: 3d printed optical fibers for sensing and display. *ACM Trans. Graph.*, 33(3):24:1–24:13, June 2014. xii, 33

[PTC10] Nico Pietroni, Marco Tarini, and Paolo Cignoni. Almost isometric mesh parameterization through abstract domains. *IEEE Transaction on Visualization and Computer Graphics*, 16(4):621–635, July/August 2010. xv, 82, 83

[PTSZ11] Nico Pietroni, Marco Tarini, Olga Sorkine, and Denis Zorin. Global parametrization of range image sets. *ACM Trans. Graph.*, 30(6):149:1–149:10, December 2011. 41

[PWLSH13] Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. Make it stand: Balancing shapes for 3d fabrication. *ACM Trans. Graph.*, 32(4):81:1–81:10, July 2013. 21

[PZM+15] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. Elastic textures for additive fabrication. *ACM Trans. Graph.*, 34(4):135:1–135:12, July 2015. 89

[RA15] Ronald Richter and Marc Alexa. Beam meshes. *Computers & Graphics*, 53, Part A:28 – 36, 2015. 40 years of Computer Graphics in Darmstadt. xi, 25, 26

[RCM+14] Tim Reiner, Nathan Carr, Radomír Měch, Ondřej Šťava, Carsten Dachsbacher, and Gavin Miller. Dual-color mixing for fused deposition modeling printers. *Comput. Graph. Forum*, 33(2):479–486, May 2014. xi, 14

[RHX13] A Radman, X Huang, and YM Xie. Topological optimization for the design of microstructures of isotropic cellular materials. *Engineering Optimization*, 45(11):1331–1348, 2013. 91

[RLL+06] Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. Periodic global parameterization. *ACM Trans. Graph.*, 25:1460–1485, October 2006. 41

[RMP11] Andreas Reichinger, Stefan Maierhofer, and Werner Purgathofer. High-quality tactile paintings. *ACM Journal on Computing and Cultural Heritage (JOCCH)*, 4(2), 2011. 32

[RRP15] Faniry H. Razafindrazaka, Ulrich Reitebuch, and Konrad Polthier. Perfect matching quad layouts for manifold meshes. *Computer Graphics Forum (proceedings of EUROGRAPHICS Symposium on Geometry Processing)*, 34(5), 2015. 69

[RVAL09] Nicolas Ray, Bruno Vallet, Laurent Alonso, and Bruno Lévy. Geometry-aware direction field processing. *ACM Trans. Graph.*, 29(1):1:1–1:11, December 2009. 43, 69

[SÍ12] Carlo H. Séquin. Prototyping dissection puzzles with layered manufacturing. In *Fabrication and Sculpture Track, Shape Modeling International Conference*, College Station, Texas, 2012. Texas A&M Univ. xii, 36, 45

[SA07] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 109–116, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. 35

[SB12] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, page 20. ACM, 2012. 68

[SBM16] Ariel Shamir, Bernd Bickel, and Wojciech Matusik. Computational tools for 3d printing. In *ACM SIGGRAPH 2016 Courses*, SIGGRAPH '16, pages 9:1–9:34, New York, NY, USA, 2016. ACM. 7

[SBR+15] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. Microstructures to control elasticity in 3d printing. *ACM Trans. on Graphics (TOG)*, 34(4), 2015. 20

[SBS07] Wenhao Song, Alexander Belyaev, and Hans-Peter Seidel. Automatic generation of bas-reliefs from 3d shapes. In *Proc. of the IEEE International Conf. on Shape Modeling and Applications 2007*, SMI '07, Washington, DC, USA, 2007. IEEE Computer Society. 31

[SCGT15] Mélina Skouras, Stelian Coros, Eitan Grinspun, and Bernhard Thomaszewski. Interactive surface design with interlocking elements. *ACM Trans. Graph.*, 34(6):224:1–224:7, October 2015. xi, 26, 27

[Sch03] Martin Schilling. *Catalog mathematischer modelle: fur den hoheren mathematischen unterricht*. Martin Schilling, 1903. 38

[SDW+16] Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. Cofifab: Coarse-to-fine fabrication of large 3d objects. *ACM Trans. Graph.*, 35(4):45:1–45:11, July 2016. xi, 12

[SFCO12] Peng Song, Chi-Wing Fu, and Daniel Cohen-Or. Recursive interlocking puzzles. *ACM Trans. Graph.*, 31(6):128:1–128:10, November 2012. xii, 36

[Sha08] Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008. 73

[Sha16] Shapeways: 3d printing service. `http://www.shapeways.com/`, 2016. 1

[SI06] Mitul Saha and Pekka Isto. Motion planning for robotic manipulation of deformable linear objects. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2478–2484. IEEE, 2006. 68

[Si10] Hang Si. A quality tetrahedral mesh generator and a 3D Delaunay triangulator. *URL http://tetgen.berlios.de*, 2010. 108

[Sig95] Ole Sigmund. Tailoring materials with prescribed elastic properties. *Mechanics of Materials*, 20(4):351–368, 1995. 91, 95, 103, 107

[Sin10] Rupinder Singh. Three dimensional printing for casting applications: A state of art review and future perspectives. In *Advanced Materials Research*, volume 83, pages 342–349. Trans Tech Publ, 2010. 67

[SP12] Yuliy Schwartzburg and Mark Pauly. Design and optimization of orthogonally intersecting planar surfaces. In Christoph Gengnagel, Axel Kilian, Norbert Palz, and Fabian Scheurer, editors, *Computational Design Modelling*, pages 191–199. Springer, Berlin Heidelberg, 2012. 26, 41, 43, 44

[SP13] Yuliy Schwartzburg and Mark Pauly. Fabrication-aware design with intersecting planar pieces. *Computer Graphics Forum*, 32(2pt3):317–326, 2013. xi, 25, 26, 44, 45, 68

[SPG+16] Christian Schüller, Daniele Panozzo, Anselm Grundhöfer, Henning Zimmer, Evgeni Sorkine, and Olga Sorkine-Hornung. Computational thermoforming. *ACM Trans. Graph.*, 35(4):43:1–43:9, July 2016. xi, 15

[SPSH14] Christian Schüller, Daniele Panozzo, and Olga Sorkine-Hornung. Appearance-mimicking surfaces. *ACM Trans. Graph.*, 33(6), November 2014. xii, 32

[SRML09] Xianfang Sun, Paul L. Rosin, Ralph R. Martin, and Frank C. Langbein. Bas-relief generation using adaptive histogram equalization. *IEEE Trans on Visualization and Computer Graphics*, 15(4), July 2009. 31

[SS10a] Mayank Singh and Scott Schaefer. Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph.*, 29(4):46:1–46:7, July 2010. 19

[SS10b] Christian Stussak and Peter Schenzel. Realsurf – a tool for the interactive visualization of mathematical models. In Fay Huang and Reen-Cheng Wang, editors, *Arts and Technology: First International Conference, ArtsIT 2009, Yi-Lan, Taiwan, September 24-25, 2009*, pages 173–180, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 39

[STBG12] Melina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross. Computational design of rubber balloons. *Computer Graphics Forum*, 31(2pt4), 2012. 30, 118

[STC+13] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. Computational design of actuated deformable characters. *ACM Trans. Graph.*, 32(4):82:1–82:10, July 2013. 19, 92

[STK+14] Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. Designing inflatable structures. *ACM Trans. Graph.*, 33(4), July 2014. xi, 30

[STL06] Idan Shatz, Ayellet Tal, and George Leifman. Paper craft models from meshes. *Vis. Comput.*, 22:825–834, September 2006. xi, 22, 23, 42

[Str14] Stratasys. Objet500 connex3. http://www.stratasys.com/3d-printers/production-series/connex3-systems, 2014. 14

[STTP14] Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. High-contrast computational caustic design. *ACM Trans. Graph.*, 33(4):74:1–74:11, July 2014. xii, 34, 35

[SVB+12] Ondrej Stava, Juraj Vanek, Bedrich Benes, Nathan Carr, and Radomír Měch. Stress relief: Improving structural strength of 3d printable objects. *ACM Trans. Graph.*, 31(4):48:1–48:11, July 2012. 16

[SVWG12] Justin Solomon, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. Flexible developable surfaces. In *Computer Graphics Forum*, volume 31, pages 1567–1576. Wiley Online Library, 2012. 68

[SWL+11] J. Schwerdtfeger, F. Wein, G. Leugering, R. F. Singer, C. Krner, M. Stingl, and F. Schury. Design of auxetic structures via mathematical optimization. *Advanced Materials*, 23(22):2650–2654, 2011. 92

[Tac09] Tomohiro Tachi. 3d origami design based on tucking molecule. In *The Fourth International Conference on Origami in Science, Mathematics, and Education, R. Lang, ed., Pasadena*, pages 259–272, 2009. 24

[Tac10] Tomohiro Tachi. Origamizing polyhedral surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):298–311, March 2010. 24

[Tam] Tama Software Ltd. PePaKuRa Designer. `http://www.tamasoft.co.jp/pepakura-en/`. 23

[TBWP16] Chengcheng Tang, Pengbo Bo, Johannes Wallner, and Helmut Pottmann. Interactive design of developable surfaces. *ACM Transactions on Graphics (TOG)*, 35(2), 2016. 68

[TD04] S Torquato and A Donev. Minimal surfaces and multifunctionality. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2047):1849–1856, 2004. 91

[THD02] S Torquato, S Hyun, and A Donev. Multifunctional composites: optimizing microstructures for simultaneous transport of heat and electricity. *Physical review letters*, 89(26):266601, 2002. 91

[THD03] S Torquato, S Hyun, and A Donev. Optimal design of manufacturable three-dimensional composites with multifunctional characteristics. *Journal of Applied Physics*, 94(9):5748–5755, 2003. 91

[Tor02] Salvatore Torquato. *Random heterogeneous materials: microstructure and macroscopic properties*, volume 16. Springer, 2002. 91

[TPP+11] Marco Tarini, Enrico Puppo, Daniele Panozzo, Nico Pietroni, and Paolo Cignoni. Simple quad domains for field aligned mesh parametrization. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 142:1–142:12, New York, NY, USA, 2011. ACM. 69

[TSP13] Romain Testuz, Yuliy Schwartzburg, and Mark Pauly. Automatic Generation of Constructable Brick Sculptures. In M.-A. Otaduy and O. Sorkine, editors, *Eurographics 2013 - Short Papers*. The Eurographics Association, 2013. 37

[UBM15] Nobuyuki Umetani, Bernd Bickel, and Wojciech Matusik. Computational tools for 3d printing. In *ACM SIGGRAPH 2015 Courses*, page 9. ACM, 2015. 7, 66

[UKIG11] Nobuyuki Umetani, Danny M Kaufman, Takeo Igarashi, and Eitan Grinspun. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.*, 30(4), 2011. 29

[ULP+15] Francesco Usai, Marco Livesu, Enrico Puppo, Marco Tarini, and Riccardo Scateni. Extraction of the quad layout of a triangle mesh guided by its curve skeleton. *ACM Trans. Graph.*, 35(1):6:1–6:13, December 2015. 87

[VC+10] Angela Vierling-Claassen et al. Models of surfaces and abstract art in the early 20th century. In *Proceedings of Bridges 2010: Mathematics, Music, Art, Architecture, Culture*, pages 11–18. Tessellations Publishing, 2010. 39

[VGB14a] J. Vanek, J. A. G. Galicia, and B. Benes. Clever support: Efficient support structure generation for digital fabrication. *Comput. Graph. Forum*, 33(5):117–125, August 2014. 11

[VGB+14b] J. Vanek, J. A. Garcia Galicia, B. Benes, R. Měch, N. Carr, O. Stava, and G. S. Miller. Packmerger: A 3d print volume optimizer. *Computer Graphics Forum*, 33(6):322–332, 2014. 68

[VWRKM13] Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik. Openfab: A programmable pipeline for multi-material

fabrication. *ACM Transactions on Graphics (TOG)*, 32(4):136, 2013. 92

[Wan11] Somlak Wannarumon. Reviews of computer-aided technologies for jewelry design and casting. *Naresuan University Engineering Journal*, 6(1):45–56, 2011. 67

[WCT+15] Weiming Wang, Haiyuan Chao, Jing Tong, Zhouwang Yang, Xin Tong, Hang Li, Xiuping Liu, and Ligang Liu. Saliency-preserving slicing optimization for effective 3d printing. *Computer Graphics Forum*, 34(6):148–160, 2015. 10

[WDB+07] Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein. Digital bas-relief from 3d scenes. *ACM Trans. Graph.*, 26(3), 2007. 31

[WFY+10] Tai-Pang Wu, Chi-Wing Fu, Sai-Kit Yeung, Jiaya Jia, and Chi-Keung Tang. Modeling and rendering of impossible figures. *ACM Trans. Graph.*, 29(2), 2010. 39

[WPMR09] Tim Weyrich, Pieter Peers, Wojciech Matusik, and Szymon Rusinkiewicz. Fabricating microgeometry for custom surface reflectance. *ACM Trans. Graph.*, 28(3):32:1–32:6, July 2009. 34

[WSW+12] Emily Whiting, Hijung Shin, Robert Wang, John Ochsendorf, and Frédo Durand. Structural optimization of 3d masonry buildings. *ACM Trans. Graph.*, 31(6):159:1–159:11, November 2012. 18

[WW12] Martin Waßmann and Karsten Weicker. Maximum flow networks for stability analysis of legostructures. In Leah Epstein and Paolo Ferragina, editors, *Algorithms – ESA 2012: 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, pages 813–824, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 37

[WW16] L. Wang and E. Whiting. Buoyancy optimization for computational fabrication. *Computer Graphics Forum*, 35(2):49–58, 2016. 21

[WWY+13] Weiming Wang, Tuanfeng Y. Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen, and Xiuping Liu. Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph.*, 32(6):177:1–177:10, November 2013. xi, 17

[XKM07] Jie Xu, Craig S. Kaplan, and Xiaofeng Mi. Computer-generated papercutting. In *Computer Graphics and Applications, 2007. PG '07. 15th Pacific Conference on*, pages 343–350, 2007. 23

[XLF+11] Shiqing Xin, Chi-Fu Lai, Chi-Wing Fu, Tien-Tsin Wong, Ying He, and Daniel Cohen-Or. Making burr puzzles from 3D models. *ACM Trans. Graph.*, 30(4):97:1–97:8, July 2011. xii, 36

[XXY+15] Yue Xie, Weiwei Xu, Yin Yang, Xiaohu Guo, and Kun Zhou. Agile structural analysis for fabrication-aware shape editing. *Comput. Aided Geom. Des.*, 35(C):163–179, May 2015. 16

[YCL+15] Miaojun Yao, Zhili Chen, Linjie Luo, Rui Wang, and Huamin Wang. Level-set-based partitioning and packing optimization of a printable model. *ACM Transactions on Graphics (TOG)*, 34(6):214, 2015. 12, 68

[YCS02] Z Y Yang, Y G Chen, and W S Sze. Layer-based machining: Recent development and support structure design. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 216(7):979–991, 2002. 9

[YIC+12] Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. Pixel art with refracted light by rearrangeable sticks. *Computer Graphics Forum*, 31(2pt3), 2012. xii, 33, 34

[YIC+14] Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. Poisson-based continuous surface generation for goal-based caustics. *ACM Trans. Graph.*, 33(3):31:1–31:7, June 2014. 34

[ZGH+16] Haisen Zhao, Fanglin Gu, Qi-Xing Huang, Jorge Garcia, Yong Chen, Changhe Tu, Bedrich Benes, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Connected fermat spirals for layered fabrication. *ACM Trans. Graph.*, 35(4):100:1–100:10, July 2016. xi, 9, 10

[ZK14] H. Zimmer and L. Kobbelt. Zometool rationalization of freeform surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 20(10):1461–1473, Oct 2014. xii, 38

[ZLAK14] Henrik Zimmer, Florent Lafarge, Pierre Alliez, and Leif Kobbelt. Zometool shape approximation. *Graph. Models*, 76(5):390–401, September 2014. 38

[ZLP+15] Xiaoting Zhang, Xinyi Le, Athina Panotopoulou, Emily Whiting, and Charlie C. L. Wang. Perceptual models of preference in 3d printing direction. *ACM Trans. Graph.*, 34(6):215:1–215:12, October 2015. 13

[ZLW+16] Xiaoting Zhang, Xinyi Le, Zihao Wu, Emily Whiting, and Charlie C. L. Wang. Data-Driven Bending Elasticity Design by Shell Thickness. *Computer Graphics Forum*, 2016. 20

[ZPZ13] Qingnan Zhou, Julian Panetta, and Denis Zorin. Worst-case structural analysis. *ACM Trans. Graph.*, 32(4):137:1–137:12, July 2013. xi, 16, 17

[ZTZ16] Yizhong Zhang, Yiying Tong, and Kun Zhou. Coloring 3d printed surfaces by thermoforming. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016. 15

[ZYZZ15] Yizhong Zhang, Chunji Yin, Changxi Zheng, and Kun Zhou. Computational hydrographic printing. *ACM Trans. Graph.*, 34(4):131:1–131:11, July 2015. xi, 14, 15

[ZZL10] Chunjie Zhang, Xionghui Zhou, and Congxin Li. Feature extraction from freeform molded parts for moldability analysis. *The International Journal of Advanced Manufacturing Technology*, 48(1-4):273–282, 2010. 67

[ZZZY13] Yu-Wei Zhang, Yi-Qi Zhou, Xiao-Feng Zhao, and Gang Yu. Real-time bas-relief generation from a 3D mesh. *Graphical Models*, 75(1), 2013. 31