EUROGRAPHICS 2017
V. Ostromoukov and M. Zwicker
(Guest Editors)

Volume 36 (2017), Number 2
STAR – State of The Art Report

State of the Art in Methods and Representations for Fabrication-Aware Design

Amit H. Bermano[1] Thomas Funkhouser[1] Szymon Rusinkiewicz[1]

[1]Princeton University, Princeton, NJ, USA

Abstract

Computational manufacturing technologies such as 3D printing hold the potential for creating objects with previously undreamed-of combinations of functionality and physical properties. Human designers, however, typically cannot exploit the full geometric (and often material) complexity of which these devices are capable. This STAR examines recent systems developed by the computer graphics community in which designers specify higher-level goals ranging from structural integrity and deformation to appearance and aesthetics, with the final detailed shape and manufacturing instructions emerging as the result of computation. It summarizes frameworks for interaction, simulation, and optimization, as well as documents the range of general objectives and domain-specific goals that have been considered. An important unifying thread in this analysis is that different underlying geometric and physical representations are necessary for different tasks: we document over a dozen classes of representations that have been used for fabrication-aware design in the literature. We analyze how these classes possess obvious advantages for some needs, but have also been used in creative manners to facilitate unexpected problem solutions.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations, Physically based modeling; J.6 [Computer-Aided Engineering].

1. Introduction

Emerging fabrication technologies based on additive manufacturing offer immense control over shape, detail, and material. Unlike traditional manufacturing processes, technologies ranging from polymer-based 3D printing to metal sintering impose virtually no additional cost for near-arbitrary shape complexity and, frequently, high-frequency material variation. These technologies have already had an enormous impact on prototyping everything from shoes to airplane parts, and are beginning to demonstrate cost reduction and efficiency gains in production as well. They have provided an outlet for creativity in the "maker" community, due to their (relatively) low purchase and material costs. Most relatively, they have enabled the production of objects with unprecedented control over appearance, deformation, aesthetics, and functionality.

The need for new design tools to allow the exploitation of additive manufacturing technologies has been noted repeatedly [YZ15, GZR*15]. Indeed, there has been extensive work on tools that allow for the specification and analysis of geometrically-complex, multi-material objects intended to be produced with devices having control over non-traditional process parameters (including layering, extrusion speed, or sintering temperature). These tools, however, are insufficient if the complexity required to exploit the capability of additive manufacturing to the fullest extent exceeds what human designers are capable of understanding and specifying exhaustively. Instead, recent years have seen the emergence of systems in which designers specify the properties that the fabricated objects should have, while the computer optimizes for the (possibly highly geometrically-complex) structures that give rise to these properties.

For over a decade, the computer graphics community has been investigating fabrication-aware design tools and optimization-based digital manufacturing. These systems have exploited the data structures (shape and attribute representations) and algorithms (for synthesis, simulation, and rendering) that constitute the bedrock of computer graphics, and the range of objectives and physical properties that have been considered is wide. Among the 100+ papers summarized in this report are systems that consider appearance, strength, deformation, motion, balance, functionality, assembly, and control over the manufacturing process, with application domains including consumer products, clothing, arts and cultural heritage, robotics, and architecture.

This report analyzes recent research on fabrication-aware design by classifying systems along two axes. First, we consider the *goals* of the design process or the *tasks* for which the products of the system will be used. While some research addresses the creation of modular systems that combine multiple objectives with designer input (Section 2), other papers focus on individual goals such as:

- **appearance**, including control over the plenoptic function, scattering, color, light transmission, and patterning (Section 3);

- **deformation and motion**, including structural integrity, local and global deformation, and articulation (Section 4);
- **high-level objectives** ranging from center of mass and moments of inertia to buoyancy, aerodynamics, acoustics, and functionality (Section 5);
- **domain-specific objectives** in fields including art, clothing, furniture, and architecture (Section 6); and
- **process-specific objectives** involving minimizing manufacturing time and material usage(Section 7).

The second axis of our analysis involves the *underlying shape and attribute representations* used by each system (Section 8). It is our thesis that all representations embody some trade-off among conciseness, flexibility, control, and suitability for simulation and optimization. The purpose of our analysis is therefore partly to document the empirical importance of these properties, by noting which representations have been more or less popular among papers addressing the various goals. More importantly, looking at surprising use or non-use of representations in various contexts can provide inspiration for future research. For example, could optimization over surface-embedded vector fields provide new opportunities to balance between global smoothness and sensitivity to local features? Are there ways to perform structural analysis *without* explicitly representing stress and strain throughout the volume?

While the range of papers considered herein is broad, we focus on those published in the computer graphics community. We hope to complement recent surveys in mechanical and materials engineering covering multi-physics and topology optimization (e.g. [DG14]) by focusing on research addressing the much greater variety of goals and representations considered in graphics. In addition, we hope to complement the recent STARs on computational fabrication and display of material appearance [HIH*13] and digital fabrication technologies for cultural heritage [SCP*14] by covering a greater variety of goals and methodologies, deferring to these more specialized STARs for more in-depth coverage within these domains.

2. Computational Fabrication Pipeline

The papers considered in this report, roughly speaking, all follow a similar design pipeline. They begin with the set of capabilities of the manufacturing process, including working volume, material, resolution, and need for support. Together with common and basic rules for additive manufacturing such as connectedness and lack of inner voids, these conceptually define a "gamut" of realizable shapes, which is often restricted by individual projects to allow for faster optimization or simpler user interaction.

The next stage of the design process is to formulate a constrained optimization problem. The constraints range from structural soundness to manufacturability, while the objectives encode any of a huge variety of goals. For example, in an architectural context (Section 6.2), one goal might be to minimize the number of different types of basic elements or bricks. This reduces the quantity of molds that must be produced, which in turn reduces overall manufacturing costs. For planar-structures design (Section 6.1), the important aspect of constructibility is manifested through the ability to assemble the interlocking planes in a collision-free manner. The objectives and constraints are usually provided, or controlled, by user interaction, as is some initial guess. The optimization often involves structural, optical, or acoustic simulation, taking the shape as input in either its "primary" representation or some more convenient alternative.

When a final shape emerges that satisfies the constraints and cannot be improved in terms of the objectives, it is passed to a separate piece of code that determines exactly how the fabrication device should be driven (e.g., slices, tool paths, and support), in order to produce a high-quality object with minimum manufacturing time and cost.

Of course, many variants on the above pipeline are possible. For example, the details of the manufacturing process could be optimized together with the overall design. The design space could be restricted to a greater or lesser degree, leaving the designer with a lesser or greater degree of control over the ultimate shape. The optimization and simulation could use lower-fidelity versions of the shape, leading to faster computation at the expense of lower accuracy. Initial guesses for the shape, candidate perturbations, or local configurations of materials could be taken from pre-computed or pre-designed libraries. So, while our main focus is on analyzing papers along the two axes of "goals" and "representations", an additional undercurrent throughout the remainder of this report is the variety of ways in which the computational fabrication pipeline can be structured.

The following sections explore the large variety of goals, both general and domain-specific, that have been considered by systems for fabrication-aware design. Before describing those, however, we consider an emerging set of *general* design and fabrication systems with modular or interchangeable parts. These aim at facilitating complex creations for novice, high-level, and professional design, and often allow for the combination of multiple constraints and goals.

2.1. Design Frameworks

One example of a system designed for general-purpose computational fabrication is *openFab*, proposed by Vidimče et al. [VWRKM13]. This work defines a fabrication language and programmable pipeline for multi-material additive manufacturing. Inspired by the programmable shaders that form a major part of existing rendering systems, they provide a programmable pipeline to specify volumetric material distribution and thus synthesize 3D geometry for fabrication. To support complex geometries, the pipeline operates in a streaming manner, i.e. only a bounded amount of memory is occupied by the geometry at all times. The overall geometry is defined at a coarser scale, and is tessellated to "micro-polygons" on demand. Fine-scale geometric details (e.g. multi-material color dithering, micro-lenses, surface finishing etc.) are defined procedurally, in a shader-like manner. The framework supports additive manufacturing needs such as support structure construction (through a 2D grid on the printing plane) and density estimation per printing voxel. Efficient nearest neighbor queries are also proposed through Bounded Volume Hierarchies (BVH).

Instead of manipulating material distribution, the *Spec2Fab* system of Chen et al. [CLD*13] proposes to describe and design a

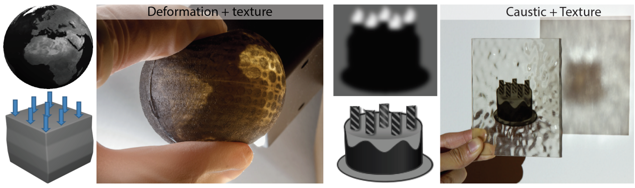


Figure 1: *3D-printed objects with combined specifications, designed using the spec2fab system [CLD*13]. **Left:** a miniature Earth model with a prescribed deformation behavior. **Right:** an optimized surface producing a caustic image as well as casting a prescribed shadow.*

fabricable object solely by its properties and desired functionality. They provide a framework and API to convert high-level specifications to practical fabrication instructions. The core concept is the combination of a *reducer tree*, in which inner nodes partition the space and leaf nodes assign materials, and a *tuner network*: these are algorithms that are attached to a reducer sub-tree, and optimize its parameters. The tuners may be interconnected and share information, and hence constitute a network. Example objectives are combinations of texturing, appearance control (goal-based caustics, shadows, etc.), and deformation behavior. The combined examples produced by the system can be seen in Figure 1.

Clearly, these design frameworks are still preliminary. As this field matures and many different applications and technologies continue to develop, a standardization of the process will most probably be beneficial. Much work and thought is still required, but this approach could facilitate system development and intercommunication between manufacturers, as has been demonstrated many times in the past in other fields (e.g. the previously-mentioned rendering, which has seen the standardization of shader stages and languages).

3. Design for Appearance

We now turn to describing fabrication-aware design systems that optimize for specific goals, beginning with appearance. For many applications, such as furniture or toy design, an object's appearance is as important as its physical or functional properties. Of course, the field of computer graphics has extensively studied the way objects look, and hence has naturally addressed these needs also in the fabrication context. In this section, we will describe fabrication-aware design aimed at shape appearance, both in the optical sense (Section 3.1) and geometric sense (Section 3.2).

3.1. Light Interaction Design

Studying the behavior of light has been at the heart of computer graphics ever since the field was created. In the past decade, the techniques developed in this field have been adapted to tackle the problem of controlling the interaction of physical objects with light, through computational fabrication. Note that we mention here only some of the most recent work, concentrating on additive manufacturing applications. A lot of work has been done to control shape appearance through optical phenomena. Inspired by the seminal work of Mitra and Pauly [MP09], several projects have tried to produce manufacturable shapes that cast meaningful shadows when illuminated correctly [BBAM12, BKB*12, ZLW*16]. Others have produced a similar effect with light passing through the object, using refraction and caustic effects [PJJ*11, PHN*12, YIC*12, THKM13, STTP14]. Producing a fully colored 3D objects is also a difficult task, mainly due to the manufacturing process. Solutions to this problem have been proposed by deforming a flat colored sheet to fit to, or form, the target 3D shape [ZYZZ15, PDP*15, SPG*16]. We refer the reader to focused surveys for more details on other appearance fabrication applications (such as BRDF control) [HIH*13].

An immediate need that arose from multi-material printing technologies is approximating desired appearance through spatial mixing of different colored materials. Hergel et al. [HL14] proposed a method to improve the quality of such multi-color prints in the case of multi-filament extrusion, by carefully planning the printing paths. First, they optimize the azimuth (angle around the z axis) of the object to reduce the chance of one extruder smearing while the other is active. The intersection volume induced by the paths of the two extruders is approximated based on Boolean mesh operations, and the best angle is determined through an exhaustive search. Second, they build a disposable wall (rampart) in proximity to the object, in the form of an offset surface, and thus creating a cleaning station which does not require long travel times. Finally, they alter the path plan to avoid potential smearing and oozing effects caused when the extruders are idle or traveling over the object. 2D slices with contour traces are considered during path planning. The different parts of the slice are printed according to ambient occlusion computations (on a sparse 3D grid), to hide away path interfaces in less visible regions.

A different approach for better color control is via halftoning. Reiner et al. [RCM*14] propose to slightly modulate the printed shape in a different manner on every slice. In low-end devices, this will expose or hide material of certain colors and thus minimize effects that are due to low accuracy. They propose a sine function modulation on the object's surface to offer smooth gray-level control for black and white material prints. For translucent materials, on the other hand, a different method has been introduced. This method determines color values for each voxel close enough to the surface, taking into account the printer's tonal gamut and subsurface scattering [BAU15]. The method diffuses the error locally following a novel iso-distance surface traversal scheme, and produces the results on fixed sized batches of slices (implying bounded memory and processing time). The desired appearance is converted to CMYK tonal values, and the error is diffused using a function similar to traditional 2D halftoning.

Of course, light interacts with manufactured surfaces in the entire volume, and not just near the surface. Papas et al. [PRJ*13] proposed a method to replicate the appearance of translucent materials using pigmented silicon. A set of silicon pigments is measured using a custom-designed spectrometer, and their translucency profile is estimated for five representative spectra. The same process is done for a target material, and a recipe for pigment mixing reproducing the appearance of the given material is sought. The optimization considers both color and reflectance profile, making the fabricated result resemble the target under different lighting conditions and geometric shapes.

Another approach for controlling the volumetric propagation of light is that of Pereira et al. [PRM14], who studied 3D printing with multiple transparent materials to yield an effect similar to optical fibers embedded in an object. Given a shape and its parametrization, a routing optimization computes fiber paths within the shape that conform to fabrication constraints and target light routing. The latter described connections between a given plane to its corresponding points on the mesh's parametrized surface. The fiber fabrication is done by printing a clear material, coated by a low index-of-refraction material (support material in this case). The fibers within the shape are optimized to be as smooth and as far apart as possible, to maximize internal reflections. The solution is formulated using implicit functions, such that each fiber is traced out over a different level set.

Because of the great interest in light interaction within graphics, it is natural that a large number of fabrication-related papers in the community have addressed this. The underlying representations have involved various subsets of the plenoptic and scattering functions (see [HIH*13] for details), often with novel constraints imposed on the optimization to reduce computational cost. In all, we are remarkably close to being able to reproduce many components of objects' appearance in a way that is indistinguishable from the originals: much as graphics has slowly approached photorealism in rendering, photorealism in 3D fabrication now appears to be within reach.

3.2. Geometric Pattern Design

Enriching the visual aspect of designs is not restricted to appearance alterations alone, but can also be done by adding intricate structural details. In the case of fabrication-aware design, an interesting challenge is to design objects consisting *only* of these patterns, instead of having a base object that is merely augmented by them. This is challenging due to the need to satisfy manufacturability constraints, as discussed in Section 2, while keeping the shape as similar as possible to the target.

To address this, a method for topology-constrained pattern synthesis was proposed by Zhou et al. [ZJL14]. Given a 2D exemplar, the method strives to tile it along a curve in a fabricable manner. The exemplar is divided into several vertical pieces, which are allowed to be connected in any order, as long as they have the same quantity of "portals" (non-void regions) along their interface. A dynamic program is employed to find the best combination of the sub-pieces, consisting of parts as whole as possible while maintaining connectedness. Other parameters are also available, such as the number of internal holes. Since drifts are introduced when connecting out-of-order sub-parts, a smooth deformation aligns the result back to the given target curve in a final step. Later, Martinez et al. [MDLW15] extended this approach to combine texture synthesis with topology optimization, i.e. to optimize for compliance under given loads and for similarity to a given example. This method is designed for the 2D case, but it should be extendible to 3D with some effort. The method divides the plane into grid elements, which are assigned void or full labels by the end of the process. Compliance is formulated through an objective that combines constraints computed by FEM, and local shape similarity with respect to the exemplar (as is popular in texture synthesis techniques). The highly non-linear problem is solved through the Globally Convergent Method of Moving Asymptotes (GCMMA).

In 3D, the approach of choice for this problem is covering the given surface with the exemplars. Dumas et al. [DLL*15] propose a method adapted from traditional by-example texture synthesis. Given a mesh and a pattern image, the method carves the mesh out. The carving seeks to resemble the input patterns, while respecting fabricability, in the form of connectedness, as well as resistance to given external and internal forces. The shell is hierarchically voxelized (resembling an octree), and each voxel is projected onto a plane. The planes are copies of the input image, distributed over the unit sphere. Image directions and positions on the sphere are determined by optimizing smoothness of image directions and colors. The carving is determined according to the color of the projected positions. After carving, a simplified version of the mesh and its new connectivity is constructed, from which an approximative shell is created for FEM simulation. When weak spots are detected, neighboring pieces are connected through "bridges" to reinforce the mesh, and the process is iterated to incorporate these additions in a pleasing manner.

Chen et al. [CZX*16] propose another solution for the same problem. The input basic exemplar elements are represented by their medial axes, and are distributed on the surface using a blue noise method. These are then translated and deformed to improve visual quality, aiming at tangential connections or partial overlaps between elements. This step employs the Hausdorff distance for shape matching and a connectivity graph to ensure rigidity. After the optimization, the medial axis is thickened to become a volumetric mesh for FEM analysis. If weak spots are found, new elements are added and the previous steps are repeated until convergence. The resulting patterns, and their potential applications are depicted in Figure 2.

Zehnder et al. [ZCT16] introduced an interactive way to cover a surface with user defined curves. The curves are drawn in 2D, and are represented as b-splines. Then, they are lifted to 3D and are embedded on the surface with minimal distortion. The shape is represented as a Loop subdivision surface, and elastic rules are applied to the curves in order to maintain their shape (geodesic curvature) and fit them to that of the surface (normal curvature). The curves can be manually placed, or randomly distributed and grown (in a user-controlled fashion) until reaching each other. Structural analysis is then run for shape stability and avoidance of large local deformations. Modal analysis (see Section 4.1) is used to find load vectors, and virtual edges are added between the curves to measure the amount of deformation. Large deformations of virtual curves are pointed out to the user for correction (by adding elements).

Repetitive patterning has been used to decorate both functional and ornamental objects for centuries. As seen in this section, additive manufacturing has enabled the design of shapes consisting of the patterns alone, without a supporting base surface. In addition, the methods here enable this type of design for novice users, which until now was only in the realm of expert jewelers and woodworkers. The analogy to texturing and texture synthesis is straightforward, and hence parametrization is naturally employed by these

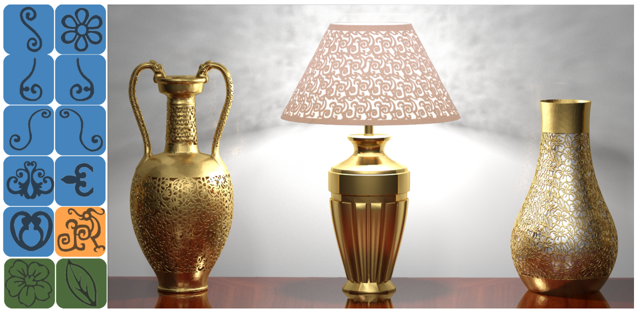


Figure 2: *Fascinating and aesthetic appearance can be created by modulating non-detailed objects (right), with combinations of pattern exemplars (left), as presented by Chen et al. [CZX*16]*

methods. In the future, it would be interesting to combine such patterning techniques with applications directed more at structural integrity, such as decorative yet functional furniture.

4. Design for Deformation and Motion

In addition to appearance, many other fields studied by the computer graphics community are applicable to computational fabrication. Subjects such as shape analysis, deformation, animation, high-level goal-based optimization, and physically based simulation are some examples, which have been applied to exciting new manufacturing applications. As we will see in this section, adapting traditional computer graphics knowledge to fabrication-related applications enables addressing new problems that were not considered in the past, especially relating to deformation and motion. Efficient physically-based modeling for complex geometries has been researched in computer graphics for many years, yielding different models and representations [NMK*06]. It turns out that these techniques are mature enough to approximate real-life physical objects, enabling applications such as meta-material and moving character design, when combined with example-based modeling or animation (Sections 4.2, 4.3, 4.4). We start (Section 4.1), however, with a classic aspect of any physical design — structural analysis.

4.1. Integrity Analysis and Design

The most prominent aspect of fabrication design is structural integrity analysis, i.e. making sure a design is durable under both its own weight (dead load) and other prescribed forces (live load). As we will see in later parts, structural analysis is an integral building block for most fabrication-aware tasks. An object's strength is classically analyzed by simulating the amount of deformation (or *strain*) the volumetric body undergoes when external and internal forces are applied to it. Typically, the user specifies the amount and location of said forces, and a Finite Element Method (FEM) based simulation is performed. Areas that result in excessive deformation are considered as weak points, which must be eliminated with revisions to the design. While this method yields accurate results in the vast majority of cases, it is sometimes impractical due to several reasons. Running a simulation correctly requires expertise that are not common among novice users and artistic designers. Specifying external stimuli for everyday usage of various objects can be an ill-posed or unfeasible problem. Simulation runtimes can also be tedious, especially when many iterations and experiments are needed. These challenges and others motivate this highly active field of research.

A possible solution for some of the mentioned challenges is to employ modal analysis [ZPZ13]. Modal analysis is traditionally used for vibration validation, but by testing several modes (which correlate to several vibration frequencies), one obtains a good guess for consistent weak spots. Such an analysis can locate structural weakness without any specified forces, and hence is a worst-case structural analysis, which nevertheless can easily be performed by novice users. In this method, a given volume (represented by tetrahedra or a triangular mesh that is tetrahedralized) undergoes inspection of the spectra and eigenfunctions (or modals) of its Laplacian operator. Another solution, proposed by Umetani et al. [US13] to significantly improve the performance of integrity analysis, is to consider only cross-sections of the volume. The method is able to analyze the structural integrity at interactive rates due to an extension of the Euler-Bernoulli assumption to 3D meshes. The latter assumption is common practice in beam engineering, where the moment arm is much longer than the cross-sectional dimensions. The method identifies, clusters, and analyzes such cross-sections in the object, assuming thin elongated parts.

Of course, once the analysis has identified weaknesses, the structural integrity of the designed object can also be automatically improved. One of the most rapidly growing research areas in this context is topology optimization. Topology optimization is the process of determining the optimal layout of material and connectivity inside a design domain, incorporating integrity analysis into a geometric evolution optimization. For example, one application would be to optimize for the structure that gives the highest stiffness for a limited amount of available material. While promising, these techniques involve non-linear optimization, making controlling the result challenging. Post-processing is typically required after the optimization, to account for aesthetics, manufacturability concerns etc. Hence, In computer graphics these methods are used less frequently, since more direct control is usually sought. For more details, we refer the reader to existing surveys in the field [DG14], and especially its integration with manufacturing constraints [LM16]. For example, Stava et al. [SVB*12] proposed a system for analysis and correction of structural integrity for general 3D objects. The analysis is done by traditional FEM simulation on the tetrahedralized interior of the shape. Untraditionally, the external stimuli are computed automatically by finding pinching pairs positioned where the objects are most likely to be grasped. The correction step considers the medial axis of the shape for improved performance and simplicity. Three types of corrections are proposed: adding struts, hollowing the interior, and thickening. The different options are prioritized according to their benefit and estimated aesthetic effect, and are iteratively proposed to the user.

Hollowing the volume is an effective way to reduce material and weight while maintaining the same visible shape. Therefore, it has been shown that hollowing in a honeycomb-like manner is effective in terms of material-to-strength ratio [LSZ*14]. In this method, the internal volume of a given mesh is tessellated and the stress is ana-

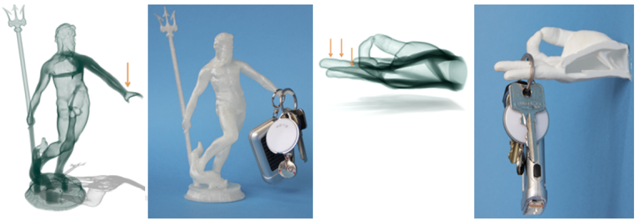


Figure 3: *Example results from the method introduced by Wu et al. [WDW15]. Both models were discretized with several million elements, and optimized in a few minutes.*

lyzed under given external loads. Then, sites are distributed inside the volume according to the computed stress map (in a halftoning-like manner), and Voronoi cells are created. Each cell is voxelized and a harmonic distance function is constructed in it. The cell is then hollowed according to the iso-surfaces induced by the distance function at a user-given value. This process, also known as porous extraction, ensures a smooth structure with controllable thickness.

Instead of hollowing, topology optimization has also been employed to improve the material-to-strength ratio, while remaining as close as possible to a prescribed shape. Christiansen et al. [CBNJ*15] propose a combined shape and topology optimization approach, where an initial shape is optimized to have a prescribed volume and withstand boundary conditions (loads). This work develops a topology optimization process that tries to remain as close as possible to the initial state. The initial meshed object, along with a box around it, are tetrahedralized, and an optimization is run to minimize the combined shape and FEM-based stress objective. The optimization is run on vertex positions of the tetrahedra and their labels (which can be solid or void). For numerical stability, a good tessellation of the volume is preserved throughout the process by remeshing operations performed after deformation iterations.

Wu et al. [WDW15] maximize strength at a prescribed material consumption ratio using topology optimization. Given a volumetric domain, it is first decomposed into a regular hexahedral grid. The stress distribution is analyzed, and the topology, only inside the volume, is evolved, in alternating steps. Through an efficient GPU implementation, this method is designed to accommodate high-resolution shapes with millions of elements. The topology optimization process is also altered to incorporate fabrication-related constraints such as minimum thickness, and design considerations such as symmetries and pattern repetition. Figure 3 depicts two fabricated results of the system for specified loads.

Efficient integrity analysis and optimization still encompass many open research questions. Accurate analysis for complex shapes at interactive rates will reduce design time significantly and potentially allow a process free of integrity concerns for the designer. To achieve this, correct approximations and reductions most probably need to be made, similar to the previously described use of modal analysis and medial axes.

4.2. Deformable Material Design

Elastic deformations are a fundamental part of the physical behavior of objects in our everyday life. The deformation of materials according to applied forces depends on many factors, from the molecular level to global object geometry. However, it is well known that the behavior of materials with micro-scale inhomogeneities can be uniformly approximated at medium scale, through numerical coarsening and *homogenization* [KMOD09]. Controlling the deformation behavior of materials through design of their micro-scale structures (called hierarchical materials, or *meta-materials*) is traditionally in the realm of material sciences [Kal15].

Pioneering work in material design in the field of computer graphics employed example-based modeling, a well established approach in the field [BBO*10]. The goal of the project is to 3D print materials with desired deformation behavior. In order to do so, first a set of base materials is collected and the deformation behavior of each one is measured using a robotic system. From these observations, via a homogenization algorithm, the nonlinear relationship between stress and strain is deduced to enable FEM simulation. This allows predicting the material deformation not only for different geometries, but also for different combinations of stacked materials. The approximation is done using radial basis function (RBF) interpolation, and the desired deformation is achieved through layering of the different materials. The target object is divided into voxels, where each can be assigned a single material. The material assignment problem is solved via a branch-and-bound algorithm with clustering. Quasi-static FEM is used to simulate the object and test whether the material assignments fit the desired behavior. The system and its results are illustrated in Figure 4. Later, the same approach of example-based modeling was used to design a silicon patch that forms specific wrinkles in the context of facial animatronics [BKS*12]. In this case, only one material is used and only its thickness is optimized (due to manufacturing constraints). As in the previous approach, the material properties are studied through example based vision of deformations. Unlike the previous approach, however, in this context continuum mechanics must be employed due to the large deformations.

Xu et al. [XLCB15] presented an interactive material design system for prescribed deformation under given external forces. The material distribution (density) is optimized continuously, and converted to discrete, fabricable material assignments in a later step. To speed up the optimization, the eigenvectors of the mesh Laplacian are computed, and the reduced model is optimized, instead of all tetrahedra directly — this approach is called modal reduction, and is similar to modal analysis, mentioned in Section 4.1.

Another approach to material design is through *micro-structures*, i.e. precomputed elements with known deformation behavior that are optimized to be tiled together to reproduce a target property. Panetta et al. [PZM*15] introduce a collection of tileable and fabricable base elements, spanning various combinations of Young's moduli and Poisson's ratios. A constrained exhaustive search over basic elements is performed to map out the different tile families and their properties. The search is constrained to tileable, manufacturable, and isotropic elements, and is performed by changing edge thickness and connectivity over 15 vertices inside one tetrahedron (which is reflected cubically to achieve isotropy). For var-

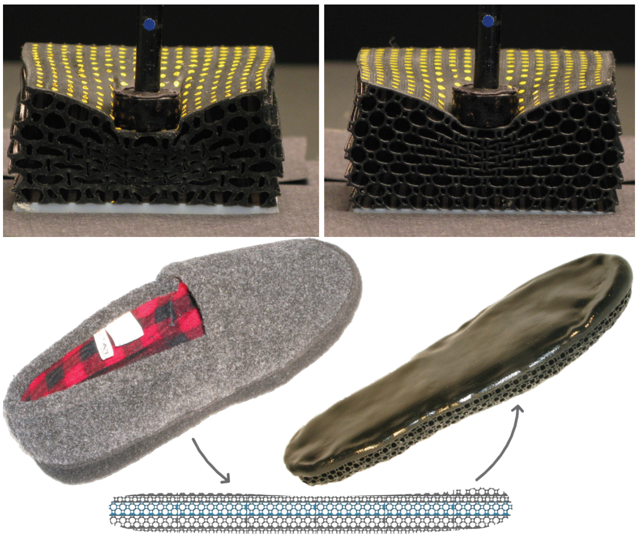


Figure 4: *Data-driven material modeling, as performed by Bickel et al. [BBO*10].* ***Top:*** *the deformation behavior of base materials and micro-structures is measured in a robotic arm system.* ***Bottom:*** *these are then stacked to replicate the deformation behavior of other measured objects.*

ious examples of given target shapes and mechanical properties, an object is fabricated through lithography, combining the different tiles to control the deformation behavior. Similarly, Schumacher et al. [SBR*15] use spatially varying micro-structure to control elastic properties. A database of micro-structure families is constructed through topology optimization on small voxelized cells, with mechanical properties computed through numerical coarsening. Then, a distance function is generated to enable interpolation between the precomputed elements during the design step. Unlike traditional topology design approaches, connectivity between cells is considered, and so are multiple candidates (instead of just one) during the optimization. This gives the optimization more flexibility in changing already-computed regions to achieve the target deformation response regardless of target shape, all while maintaining manufacturability.

Recently, a stochastic approach to generate elasticity-controlling micro-structures in a streamlined manner was proposed [MDL16]. In this setting, the elasticity (Young's modulus) of printed objects is controlled by varying the density of the micro-structures, following an open foam structure. The volume is sampled according to the desired density (derived from the desired Young's moduli through homogenization), and Voronoi cells are created accordingly. The edges of these cells constitute the open foam, and these are generated only on demand in small chunks, to enable streamlining. This construction ensures printability (no enclosed voids, and one connected component), and allows natural gradations of elasticity without any global optimization.

Deformable material design potentially allows the use of less material for elaborate deformation response, and could be combined with any of the other fields in computational fabrication to improve efficiency and performance. The work described in this part heavily depends on manufacturing technologies, and will most likely continue to evolve as additive manufacturing technologies mature. The proposed solutions focus mainly on explicit representations, enabling FEM simulation. However, the use of other representations, such as modal decomposition and Voronoi diagrams, are intriguing solutions for complexity reduction.

4.3. Deformable Shape Design

The same elasticity properties that are used in the previous section for material design could also be used to accurately predict the shape of a fabricated piece under given forces. This facilitates designing objects that match a desired shape, or several shapes, under different applied loads. In such cases, the inverse problem to the classic forward simulation must be solved; i.e. an initial shape is sought such that it will resemble a target *after* it is deformed by prescribed forces. For example, consider printing a shape that consists of two bulky parts which are weakly connected. Such a shape will probably deform under gravity. However, this can be compensated for, resulting in a fabricated shape with a distorted connection that is correct after the aforementioned deformation. This example can be challenging since classical iterative methods tend to converge slowly due to the low forces involved. It has been shown that ANM (Asymptotic Numerical Method) can be exploited to drastically improve performance in such cases [CZXZ14]. As another example, Skouras et al. [STBG12] developed a physical model and an optimization system to compute the needed rest shape of an inflatable balloon such that it will match a target shape once inflated.

The same goal, of matching a shape after deformation, can be applied for several target shapes under different boundary conditions. This will pose similar computational challenges. Perez et al. [PTC*15] propose to use a rod network to approximate the shape and reduce the dimensionality of the problem. A given triangular mesh is converted to hexagons through Voronoi tessellation, and the shape is approximated by fabricating only their edges (which are more compliant to stretch and shear). Using a rod representation (having reduced coordinates of just the central line and angles) with an elliptic cross section, the method optimizes the major and minor radii of the rods to match the desired target shapes. By correctly choosing two orthogonal directions of the ellipse, the optimization is able to control in-plane and out-of-plane deformations independently.

A more elaborate process can include multi-materials and the optimization of the actuation forces themselves. Such a system was suggested in the context character design [STC*13]. The input is a rest pose and a set of target shapes, and the system determines the volumetric material distribution, along with the actuation system (number and positions of actuators, and actuating forces). The method first distributes many actuators along the object's boundary, and computes the needed forces for each pose, restricted to physically viable directions, depending on the actu-

ation type (manual posing, pin-based, or string-based — see inset image). In a second step the actuators are unified according to a sparsity parameter, and lastly the material distribution is determined to fine tune the poses once the actuation is set.

An exotic example of deformable design is that of tensegrities: networks of rigid and elastic elements that are in static equilibrium [GCMT14]. Since these states of equilibrium are difficult to achieve, the proposed method requires a predefined library of stable building blocks that may be altered. During the design process the user can connect the basic library elements and move individual vertices, while an optimization process determines the length of required cables to maintain the equilibrium during these changes.

In summary, deformable shape design has experienced great progress thanks to additive manufacturing. Processes that involved weeks of trial and error by experienced professionals can now be dramatically sped up through simulation and accurate fabrication. In the future, soft and deformable characters are likely to be dominant in physical character interaction with humans. A purely soft character would pose no risk to a human, even when malfunctioning. In order to realize such characters, however, novel actuation mechanisms would probably need to be conceived.

4.4. Articulated Shape Design

The design of rigid bodies connected through joints has received a lot of attention recently. Such objects can be figurines or serve a supporting function when they are posable. More elaborate versions, which include mechanisms such as gear systems and motors, present a wide range of movement and can even be used to build walking robots. Realizing moving digital models physically in an automatic and efficient way is one of the holy grails of computational fabrication. This dream, of having a 3D counterpart to the traditional 2D-print button in design tools, both for CAD and animation, still requires much work; however, creating such objects via assemblies, rather than all at once, is still a significant step on this path.

One-piece fabricated models that can stably and independently hold different poses were proposed concurrently by two different publications. Calì et al. [CCA*12] proposed a system to design articulated models. A natural input for such goals are rigs [Stu98], which define the object's range of motion and thus already include locations of possible joints. The proposed method converts each such position in the rig to a mechanical printable joint from a predefined library. The user can specify angle constraints on the joints, and the method automatically adjusts the results to be fabricable, collision-free and withstand gravity. In the concurrent work, articulated characters were fabricated from skinned meshes [BBJP12]. Skinning is the process of controlling deformations of a given object using, typically, a skeleton. The surface (or *skin*) deforms along with the skeletal bones, according to the assigned weights [JDKL14]. In this case, the skinning weights are used to compute candidate joints (from a library), and are placed on the approximated medial axis. Structural integrity is approximated by considering cross-sectional width at joint locations.

In addition, more elaborate rigid motion can also be carried out through mechanical elements, such as pulleys, gears or sliders. Such *automata* can be driven by a motor or by hand, and their design traditionally requires a great amount of expertise, experience, and trial and error. The first attempt to automatically generate a mechanism assembly that replicates a prescribed motion was for mechanical toys [ZXS*12]. The user specifies the desired shape, a kinematic chain of its possible motion, and the target motion of specified features. The system then assigns a mechanism from a predefined library according to the type of specified motion, and optimizes its parameters through simulated annealing. In this setting, the target motion and geometry are restricted to be smooth, periodic and non-colliding. Later, Coros et al. [CTN*13] proposed an interactive framework for computational design of mechanical characters. A fully functional, ready to be printed as a single assembly, mechanical character is automatically generated from an input articulated mesh and desired motion curves of the end effectors. A parameterized set of motion assemblies is predefined, and the space of achievable curves is explored in pre-computation. During interactive design, assemblies are retrieved and parameters are optimized to match the desired motion curve, using curve features for comparison. Concurrently, Ceylan et al. [CLM*13] proposed the assembly of automata from motion capture data, where the mechanical elements are mainly off-the-shelf pieces, as opposed to custom printed ones. The motion is approximated through Fourier decomposition of the joint angles of a simple kinematic chain, restricted to three orthogonal planes.

Instead of exploring the whole range of mechanisms, Thomaszewski et al. [TCG*14] focused only on linkage systems. Such systems can be quite unintuitive since small changes may have significant and surprising effects on the end-effector motion. This work presents an interactive system to design such assemblies. The input is an explicitly animated skeletal character with joints, and their respective angles for each frame. The system starts by assigning a motor to each joint to reproduce the motion, and tries to minimize the number of motors by adding linkages instead. The user can interactively select from proposed combinatorial options, in order to guide the optimization to be more efficient and the result more aesthetic. A key challenge in such designs is avoiding singularities, or *mechanical locking* of the assembly. It is shown that a lock-free assembly is one that exhibits full-rank Jacobians and therefore Singular Value Decomposition (SVD) is employed to detect and avoid degeneracies during the optimization process. In a direct follow-up [BCT15a], several high-level operations were added to the system, such as directly designing the motion curves of the end effectors, controlling the overall dimensions of the linkage system etc. Instead of direct simulation, the system is represented only by joint positions and a connectivity graph, where the rest-pose length of neighboring joints must be kept. The system is modeled with an analytic (recursive) expression, which renders deriving and solving it significantly faster, allowing interactive rates.

Walking automata were also investigated. The difficult problem of a stably walking character without sensors was proposed to be handled by using a pre-configured database of mechanisms [BCT*15b]. In this method, high-dimensional valid kinematic parameters are learned from the database by using a multivariate Gaussian Mixture model (GMM), which guides an expectation-maximization optimization. The optimization considers distance,

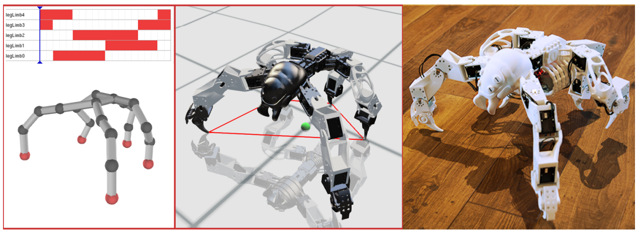


Figure 5: *Magero et al. [MTN*15] propose an interactive tool for the developments of walking robots. An arbitrary number of legs and gaits are supported.* ***Left:*** *the design of the footfall pattern graph.* ***Middle:*** *preview of the robot's support polygon and center of mass.* ***Right:*** *fabricated prototype.*

upright direction, smoothness, effort (force), and similarity to the initial design via Covariance Matrix Adaptation (CMA). Megaro et al. [MTN*15] propose an interactive tool that allows novice users to design a printable walking robot with an arbitrary number of legs, general gait style, and possibly a spine. The robot consists of 3D printed parts and off-the-shelf servo motors. The user can interactively change the dimensions of the limbs, their joints, and the walking style, represented by motion curves and footfall temporal patterns. The system interactively computes a walking sequence that respects all the desired preferences while keeping the object's center of mass within its support polygon for stability. The optimization is done on a kinematic chain representation, common to robotics applications. An illustration of the design system and its results can be seen in Figure 5.

An interesting aspect of all the publications described here is the representation of motion. Motion has to be conveniently given as input, and efficiently and accurately compared during optimizations. Traditional animation tools such as rigs and linear blend skinning were elegantly used to provide hints to the design systems about possible joint locations. Motion capture is another approach taken for input motion representation. It, however, like other forms of inputs described here, was converted to motion curves, which is the dominant representation for intermediate steps, where quality quantification is the main concern. Another interesting point is that all approaches used a prescribed library of mechanisms or joints to control the physical motion. An interesting challenge would be automatic computational development of these libraries.

5. Design for High-Level Objectives

Perhaps the most exciting contribution the computer graphics community can provide in the context of computational fabrication is design through high-level goals. One of the most prominent examples is the one proposed by Shugrina et al. [SSM15]. In order to bring the fabrication-aware design process closer to novice users, they propose to control the resulting shape by exposing only a small set of parameters to the designer. A system to explore predefined families of objects is introduced, where professional designers create the designs with many parameters and constraints. In addition, the expert designers define validity tests, as well as a small set of parameters to be exposed to the user. The system adaptively samples the highly multi-dimensional design space (using a k-d tree), and maps the user-exposed parameters to the underlying ones. The object is represented through base geometry generators and a tree of geometry processing operations, which could potentially hold anything from CSG operations to fluid simulations. During the design process, only manufacturable instantiations are allowed, according to the pre-defined validity tests. The user can tune the exposed values, and live feedback provides an indication of further valid exploration ranges, consisting of only valid geometry. In order to speed up the instantiation process, the sampled geometry is cached at sub-tree levels.

In this section, we will explore designing and optimizing shapes under specific non-trivial, non-local objectives. First, we will focus on designing for integral quantities, i.e. physical properties which require integration over the whole object (Section 5.1). Then, we will see how the functionality of an object, and the functional relationship between its parts can be used to guide the design process (Section 5.2). Lastly, we will dive into a specific case of functional design, which involves puzzles and other interlocking designs (Section 5.3).

5.1. Design for Integral Quantities

Controlling integral quantities, such as total mass (0th moment), center of mass (1st moment), rotational axes and moments of inertia (2nd moment), as well as buoyancy, aerodynamics, and acoustics is a challenging problem, addressed by the publications described in this section. Several different approaches have been taken, addressing the efficient accumulation of properties over the entirety of the designed object. Anything from explicit evaluation per element through regular or adaptive spatial partitioning to elaborate manifold and spherical harmonics are employed in order to tackle these non-trivial challenges.

Perhaps the first step in this direction was controlling the center of mass of fabricated objects, and thus ensuring their stability when placed on a surface, or hung from a string [PWLSH13]. Given a triangle mesh and the relative direction of gravity, this method ensures the object's stability by hollowing and deforming the model. Mass and center of mass are computed by integrating over the volume between the outer and inner surfaces of the model. To facilitate carving, a voxel grid represents the inside of the object. To control the center of mass and maintain printability, a heuristic carving scheme is proposed where all voxels outside of a cutting plane are marked as hollow, excluding some of the boundary ones (in accordance with minimal wall thickness constraints). User-defined handles are used to deform the surface via linear blend skinning (LBS), and an iterative optimization process alternates between inner voxel carving (discrete) and deformation (continuous). Gradient descent with analytically computed gradients (using triangular meshes with Laplacian coordinates) is used for solving the deformation step and keeping the interface at reasonable frame-rates.

As follow-up work, Bächer et al. [BWBSH14] propose to control the principal direction of the moment of inertia, and thus allow arbitrary shapes to spin about a given axis, converting a given shape to a spinning top or a yo-yo. Similarly, the optimization process employs carving and deformation. In this method, deformation is performed using an automatically generated cage and the carving optimization is more elaborate, employing sequential linear-quadratic programming (SLQP) over an adaptive and dynamic octree structure. Musialski et al. [MAB*15] propose to control various integral quantities, such as center of mass, moment of inertia, and buoyancy, by looking at offset surfaces. By computing inward and outward offsets from a given mesh, the different properties can be optimized. The offset directions are primarily orthogonal to the surface, but adjusted to avoid self-intersections. The inward offset is bounded by the medial axis. For faster and simpler optimization, a dimension-reduction scheme is proposed, employing manifold harmonic decomposition.

Another interesting aspect that was looked at is aerodynamics. Umetani et al. [UKSI14] introduced an interactive design system to make paper airplanes. Aerodynamics is analyzed locally through "wing elements", and the model is adapted for better flight performance as the user makes changes to the desired shape. Later, Martin et al. [MUB15] proposed a method for designing kites. Aerodynamics is pre-measured in a data-driven manner, and used to design and analyze new aerodynamic models. Spherical harmonics are employed to extend the aerodynamics model to many directions.

Recently, design for custom sound filters was explored [LLMZ16]. The method is based on a parameterized primitive which is simple and cubic (a cube with 6 small pipes, one on each face). The acoustic properties of this family of primitives are sampled and computed in a preprocessing step. Given a general 3D shape, an array of these elements is sought such that they follow a prescribed sound filtering behavior. A combinatorial optimization is run on the the interior of the given input shape to compute an array of voxel filters that roughly match the desired target behavior. Then, a continuous optimization fine tunes the parameters of each voxel. The result is a sound filter that dampens or passes different frequencies according to a specific target profile.

As can be seen, there is no dominant representation for optimization of integral quantities: almost every representation discussed in this survey is employed for this end in one way or another. Voxels were employed because they simplify volume and mass computations. Octrees were employed to improve the memory footprint of the latter, at the expense of some computation. Spectral representations were employed to reduce the problem's dimensionality without affecting high-frequency details. Primitives were also used to reduce imposed computations, by restricting the search space. Each method exhibits some benefits, but it seems that there is no clear consensus on any of them yet.

5.2. Design for Functionality

Knowing what an object is used for may be the most important tool to guide a high-level design process. For example, identifying that a door should cover a region when closed can easily guide its dimensions throughout the design process. Similarly, knowing that a part should act as a container allows an automatic system to indicate to the user whether objects placed in it might fall out.

In early research in this context, the valid design space of furniture was considered [UIM12]. An interactive framework was proposed that allows the user to freely design plank-based furniture, while the system provided feedback and suggestions to ensure that the design was stable and durable. The system warns the user when planks are not connected well, the design might topple, or durability is compromised due to excessive force on fasteners. The suggestion system proposes several solutions whenever such violations occur. The method is based on expressing the design space parametrically with constraints. The compact representation allows running physical simulation in real-time, and finding a few meaningful directions to explore when the constraints are violated. Koo et al. [KLY*14] suggest to enhance the design process through high-level relationships between furniture parts, and thus rendering the designed object fabricable. First, a set of relationships (e.g. *cover*, *fit-inside*, etc.) and joints dictating possible relative movement between parts (*slide*, *rotate*, *double pivot*, etc.) are defined. The initial mesh and parts are given by the user, and are restricted to cuboids; since this method is aimed for prototyping, this approximation is acceptable. During optimization, joint parameters and cuboid sizes are changed to minimize deviation and satisfy constraints, derived from user given relationships.

Lau et al. [LOMI11] propose to convert a given furniture model to fabricable parts, based on an understanding of their functionality. A grammar for man-made IKEA cabinets and tables is demonstrated. Given an aligned model and its class, the shape is labeled by a set of prescribed types, and a grammar graph that describes the mesh is sought. The labeling process uses voxels, and the graph is used to deduce part roles and relationships between parts to generate the fabrication and assembly instructions. Later, Schulz et al. [SSL*14] further present a paradigm for both designing new fabricable objects from a database of parameterized templates and how to convert an assembly of annotated CAD design to such a database. The database consists of a tree of relationships, a list of connectors, and parameterized parts. The design side includes snapping to probable locations, automatically adding connectors to parts, physical simulation for stability verification, and functionality considerations such as properly-closing doors, collision-free motion, and symmetry.

Another aspect that was explored is foldability, i.e. whether a functional object can be folded into a specific shape or volume. For example, *Boxelization* is a method to transform an object into a cube by folding [ZSMS14]. First, the space is voxelized: an initial grid is constructed, followed by a non-uniform voxelization that allows slight deformation to the object to reduce the number of near-empty voxels. Then, the cube parts of the objects are connected in a tree manner, upon which a collision-free path for all voxels to move to a boxed volume is sought. The search is done stochastically (beam search combined with simulated annealing), and template joints are assigned between voxel edges when a folding solution is found. Similarly Li et al. [LHAZ15] have suggested to add hinges to a given segmented mesh, to make the mesh able to be flattened through folding.

Recently, a method that automatically produces custom grippers and holders has been proposed [KSS*15]. Given a stationary geometry and the object that needs to be held, an automatic part is generated that can be fastened to the stationary geometry and holds the target. Several different types of parameterized template holders and grippers are precomputed, and their strength (how much force and torque is needed to move them) is measured in an example-

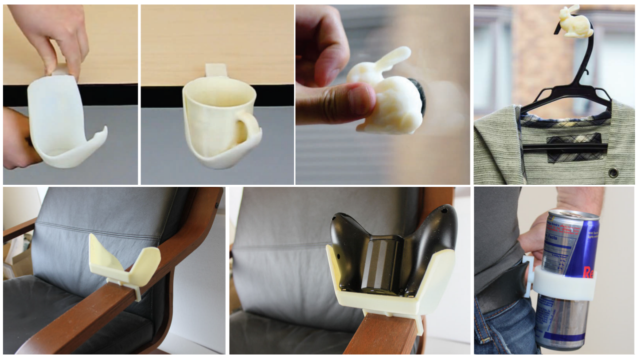


Figure 6: *Examples of the automatic generation of various grippers and holders, according to the method proposed by Koyama et al. [KSS*15]. From left to right and top to bottom: a mug holder at desk edges, a bunny-shaped coat hanger connected by suction to a window, a game controller holder on a chair arm, and a two-sided holder, connecting to a belt and a soda can.*

based, scattered data interpolation manner. An optimization is run to find the gripper with minimum volume that both holds on to the input geometry and provides enough strength. For non-standard geometry, a region growing approach is applied to sufficiently cover the held object. The user can select from different proposals for grippers on both the source and target objects interactively. Additionally, it is possible to leave one axis free for movement, or design a gripper with a cut that can be clicked into place. Some of the possible applications for this method are demonstrated in Figure 6.

As seen in the work described above, segmentation, or the indication of parts in the designed object, is imperative for functionality understanding and management. Similarly, defining the right relationships between parts is also a crucial aspect that has been explored in this section. Despite their simplicity, the ideas described here are undoubtedly milestones on the way to pure objective-based optimization. It will be interesting to see whether such functionality descriptions, or other properties interesting to the designer, can be deduced from analyzing large collections of shapes (e.g., *ShapeNet* [CFG*15]), and exploited as well in this context.

5.3. Design of Interlocking Assemblies

A special case of functionality-based design is design with interlocking pieces. Such objects have the nice property of not moving, without the need for fasteners. An appealing application for this class is burr puzzles, in which a single *key* piece locks all the pieces together, making assembling them challenging. These types of structures are also useful for educational purposes [Séq12], where students can experiment with fabricated pieces of dissected geometry. This helps training for spatial understanding, for geometric modeling, and for accuracy and tolerance in the manufacturing process. Alternatively, as we will see, this property can be used for fastening-free furniture.

Lo et al. [LFL09] propose a framework for creating an interlocking, stable, 3D puzzle out of an input mesh. The puzzle pieces are 2D polyominoes with an offset surface (according to a distance map), creating a thick shell. The mesh is first parameterized and quadrangulated, then a dual graph (or connectivity graph) is constructed for the quads, upon which a polyomino tiling procedure can run in a flood-fill like manner. A construction order is built according to a dependency graph and the shape's medial axis. Tabs and blanks (male and female connectors) are added to the pieces for stability. Burr puzzles can also be automatically computed for a given shape [XLF*11]. In this method, a template burr mechanism in used to instantiate one or more burr cores in a puzzle. A connectivity graph is constructed and the puzzle is created by connecting the burr cores according to cycles in the graph (in one of four prescribed ways), while ensuring collision-free motion of the pieces. Similarly, computational generation of other interlocking mechanisms for puzzles has been introduced (e.g. [SFCO12]) and so has a method to design "twisty" puzzles [SZ15] in which pieces can be rotated in different axes about each other in a collision-free manner (similarly to a Rubik's cube).

Recently, Fu et al. [FSY*15] have leveraged such mechanisms to design interlocking furniture, i.e. adding interlocking joints to rectangular furniture to create an immobilized structure with only one mobile key part. Given the (only orthogonal) parts of the furniture, they are joined into mid-level groups. Each group is analyzed for immobilization (by an exhaustive search over all pairs), and joints are selected from a library of predefined options. Each group is left with one *key* piece, which is immobilized by a different group. This renders the exhaustive immobilization computation local and therefore feasible, until only one key is left for the whole shape.

One of the grand visions of fabrication-aware design is designing through objectives alone, i.e. having a (novice or expert) user define only what she wishes the object to be or do, with a fully automatic and controllable optimization producing a manufacturable object accordingly. The ideas described throughout this section bring this goal closer. Even though much work still needs to be done for this concept to be realized, the solutions presented here would probably be included as tools in this collective optimization. In the next section, we discuss problems that arise in different, non-traditional forms of computational fabrication.

6. Design for Domain-Specific Applications

While this report has mostly focused on additive manufacturing technologies, in many cases this is too expensive (with respect to time or material cost). In this section, we explore systems that approximate a given shape, often very coarsely, while exploiting alternative manufacturing technologies. The key goals explored by these systems tend to involve producing visually-pleasing approximations, taking advantage of the fact that humans perceive shapes even when only a few of their features are present. We first examine fabrication-oriented shape approximation, followed by two domain-specific cases that are well-studied: planar structures (Section 6.1), and architectural construction (Section 6.2).

A clear example of truly minimalistic shape approximation is that of stable rod structure design [MLB16], in which thick wires are bent to approximate a shape on one hand, but at the same time ensure the sculpture is stable and can be bent and assembled efficiently (Figure 7 top left). One of the first publications in this context is an interactive tool for designing plush toys [MI07]. In this system, a prescribed set of simple operations can be performed on a triangle mesh by the user. These operations affect a set of 2D patches, which are generated, deformed, split or merged, while the 3D mesh is updated as a result of a simplistic simulation run over these patches, between which seam curves define the interfaces. Similarly, Skouras et al. [STK*14] present an interactive system to design inflatable (non-stretching) objects. The system automatically finds flat patches to connect, by employing an optimization over FEM simulation, and uses tension field theory to estimate wrinkling effects on simplified versions of the mesh (to speed up optimization and simulation).

Garg et al. [GSFD*14] propose a computational approach for designing with wire sheets (woven wires arranged in a regular grid — see Figure 7, top right). This type of design is difficult due to several mechanical properties induced by the physical configuration of the material: local changes have global effects, stretch is not possible but shear is, and local shear and curvature are dependent. It turns out that Chebyshev nets present this quality, and a novel representation for discrete Chebyshev nets is introduced. It is further extended to enable design under these constraints, employing several strategies to overcome mathematical and computational difficulties, such as integration, multi-grid, interpolation etc.

Recently, a method for interactive design via auxetic materials (i.e., flat flexible material that can stretch uniformly up to a certain extent) was presented [KCD*16]. This class of materials is realized through cutting non-extensible materials, such as metals or plastic, in a specific regular shape. Elements formed through this cutting scheme can rotate relative to one another, allowing for a limited stretching effect. This enables the approximation of doubly-curved surfaces (such as the sphere) using only flat pieces, making it attractive for fabrication. Various possible applications are demonstrated, through an elegant use of conformal geometry, which facilitates otherwise non-trivial or even infeasible optimization (see Figure 7, bottom).

6.1. Planar-Structure Design

The computer graphics community has also explored the approximation of desired shapes using only planar elements. For physical objects, this traditionally falls in the realm of papercraft [MS04,MGE07]. The emergence of widespread computational fabrication tools such as laser-cutters or Computerized Numerical Control (CNC) devices has given rise to new applications in this field. For example, accurately cutting a single sheet of paper can approximate elaborate shapes with only one fold, generating so-called *pop-up models* [LSH*10, LJGH11]. Motivated by the need for quick and inexpensive physical prototypes for visualization, planar structures were proposed as well, as portrayed in this section.

Hildebrand et al. [HBA12] proposed to create such an approximate physical visualization by generating planes based on planar shape cross-sections, sliding them onto one another. This method proposes to iteratively add new planes to the structure until the shape is sufficiently depicted. This approach poses several challenges. In order to approximate the shape well, selected cross-sections are altered to include important nearby geometric features marked by the user. Feasibility must be ensured at every step; i.e., it must be possible to slide the planes one onto another without obstruction. An extended BSP tree is proposed to facilitate the quick addition of planes into an existing array while ensuring constructibility. Lastly, insertion order is decided through a branch-and-bound strategy tree. Later, Schwarzburg et al. [SP13] introduced an interactive design framework for planar structures. By constructing a connectivity graph and running an optimization over plane positions and orientations, constructibility and rigidity are ensured. The user provides an input mesh and is able to modify the constraints (what plane fits with what other plane) interactively during the design process, yielding stable structures for a variety of uses from illustrative figurines to functional furniture without any external support or affixing.

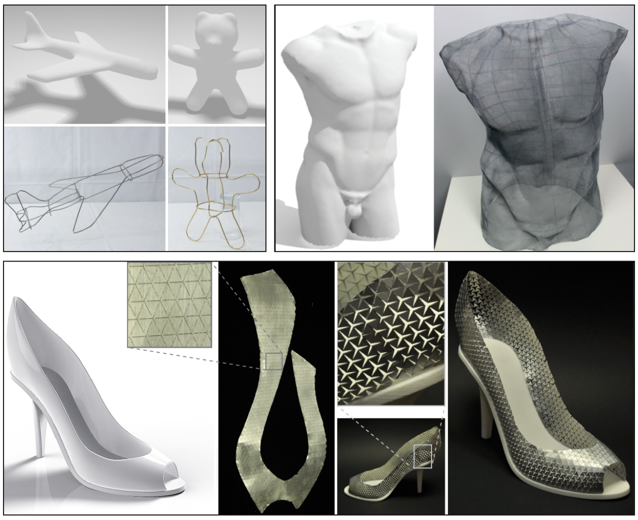


Figure 7: *Examples of domain-specific design.* ***Top left:*** *bent wire is used to sparsely approximate shapes, while ensuring stability [MLB16].* ***Top right:*** *designing with wire sheets is surprisingly challenging due to unique restrictions and global effects; this problem is solved through Chebyshev nets [GSFD*14].* ***Bottom:*** *auxetic materials allow limited stretch due to their special construct, and are addressed through conformal geometry [KCD*16].*

Planar structures can also be built according to cross fields, which are a good local approximation of the shape [CPMS14]. For this method, an input mesh is annotated with a cross field. This field, which can be automatically calculated, indicates the principal directions of the shape throughout the surface. Planar curves are then traced out on the surface, conforming with the cross field while still being as long and as smooth as possible. From these curves, planar *ribbon shape* slices are generated. Stability is estimated through a *divergence* measure, which quantifies orthogonality. Realizability is ensured through the construction of a directed intersection graph, which ensures feasibility when it is acyclic.

McCrae et al. [MUS14] further present *flatFitFab*, an interaction system for designing planar structures including a static simulation to analyze stress on the slits for fracture avoidance. The design pro-

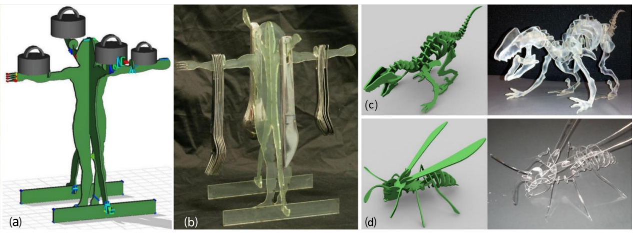


Figure 8: *Fabricated results generated by the system introduced by McCrae et al. [MUS14]. The method analyzes slit stress for given loads (a and b), and enables rapid and complex modeling by both experts (c) and amateurs (d).*

cess uses 2D sketching, imposing loads and procedural modeling, facilitating regularity in cases such as vertebrae. Physical assemblability is verified again through identifying cycles in a graph. An example of fabricated assemblies generated by the system can be seen in Figure 8. Of course, planar elements can be used to generate watertight shape approximations and not only cross-sectional ones. Such methods unify similar facing regions of an input mesh to single planes. These can then be held together through internal planar connectors [CSaLM13], or finger joint connectors [CS16].

Planar structures offer a cheap and fast method for producing a plausible representation of a full model. This method is very accessible through the growing popularity of laser cutters. As can be seen in the various results, planar structures have great expressive power for both casual and expert users, and possess a pleasing aesthetic quality. The great advantage of manufacturing only planes unfortunately also imposes the need for a manual assembly step. Thus, ensuring assemblability and post construction stability is a major concern for the aforementioned publications, and is commonly addressed by the use of a connectivity graph. One can imagine other applications that might emerge from this line of work, such as using these structures as a skeleton for a shape approximating skin (as mentioned above [GSFD*14]).

6.2. Architectural Design

As already discussed, structural integrity is an uncompromisable aspect of architectural applications. Even so, design and structural analysis are typically two independent steps, forcing the designer to alternate between them during the development process. The most common tool in architectural design is freeform geometry. This representation ensures smoothness and is convenient for design. Structural analysis is usually done using Finite Element Analysis (FEA), which requires a finite element representation. This is one of the main reasons why interleaving these two steps in uncommon. Interestingly, isogeometric analysis [CHB09] was developed to bridge the gap between a NURBS representation and FEA. This approach potentially facilitates structural analysis during the design process for CAD applications, but may be just as useful for architectural ones. Regardless, both representations do not consider manufacturing constraints. Most commonly, such constraints are aimed at lowering manufacturing costs through the basic construction elements, or panels. The following work takes this principle into account. Note that architectural design is a wide and extensive field of research, bringing together several research communities. In this part, we will partially cover the publications in the computer graphics community, concentrating on the work most relevant to fabrication-aware design. For a full survey, we refer the reader to existing reports on fabrication-aware design in the architectural context [Pot13].

Planar elements, for example, are significantly easier to manufacture than are general non-flat shapes. Liu et al. [LPW*06] explore and expand the use of PQ (Planar Quadrilateral) meshes. This work proposes a method to convert freeform structures to PQ meshes, and further introduces a new sub-class of PQ meshes, called conical meshes, which present properties advantageous to architectural applications. They do this by presenting an optimization framework to perturb the vertices of an input quadrilateral mesh. A set of terms to be optimized is specified, including planarity, fairness, and closeness to the original mesh. Additional terms are also introduced for the conversion to conical meshes, or circular arc meshes (which is another well studied sub-class of PQ meshes). Sequential Quadratic Programming (SQP) is employed for small meshes (~1k vertices) and a penalty method solved with a Gauss-Newton method for bigger meshes, due to efficiency considerations. This formulation can also be combined with subdivision surfaces to potentially create a powerful modeling tool. Similarly, Eigensatz et al. [EKS*10] minimize production cost through the approximation of a given freeform surface by easily manufactured panel templates. The method is restricted to planar, cylindrical, paraboloid, toric, and cubic patches, and seeks to minimize the set of required panel types while preserving a user-determined quality level. The solution is an iterative process alternating between continuous (nonlinear least squares via Gauss-Newton) and discrete (set cover problem) optimizations.

Another example for cost reduction in panel manufacturing is repetition of costly parts. A special mold must be made for unusual parts, and if this mold can be re-used then production costs are reduced dramatically. Given a triangular mesh, one can reduce the number of unique template triangles to a given number [SS10]. This method iteratively finds clusters of triangles (by their representative/canonical polygons) and labels each triangle appropriately. K-means is iteratively employed until the specified number of clusters is found, and the canonical polygons are computed via nonlinear least squares, minimizing the distances between all triangles in the cluster to the canonical one. Finally, the method globally optimizes vertex positions to match canonical polygons by solving a Poisson equation. This is iterated until convergence, and is rather dependent on input triangulation. Similarly, Fu et al. [FLHCO10] find a limited set of unique template non-planar quads and template assignments for a given quad surface. They iterate between clustering/generating representative quads and optimizing vertex positions until convergence. Clustering is done by analyzing an edge sharing graph and using a heuristic to solve a minimum K-clustering sum problem. This method employs conjugate gradients for non-linear optimization and is also dependent on the input quadrangulation quality.

Taking a different approach, Pietroni et al. [PTP*14] create statically strong hexagon-dominated grid-shell structures (e.g. for a steel-glass building) based on a Voronoi diagram, weighted by stress. Grid-shells are traditionally considered strongest when consisting of triangular structures, and are commonly designed using quadrilaterals due to aesthetics and other mathematical advantages. This work demonstrates how hexagon-dominated grid-shell structures can be designed to be stronger, while still remaining pleasing. First, a linear static analysis is run, which enables the decomposition of every point to the two principal stress directions (min/max stress, interpreted as a frame-field). This seeds a Voronoi diagram, which is used to compute the resulting grid-shell structure. This process yields structures with excellent static performance, since they are built to align with maximal stress direction. The method further optimizes the structures for cell regularity and symmetry, improving aesthetics.

7. Optimization of the Manufacturing Process

In most of the work in the literature, the optimization of the manufacturing process (such as tool paths and support structures) is done independently of the optimization of the shape itself, and is completely hidden from the designer. As technologies and design tools mature, however, there are many advantages to exposing these aspects to the user. For example, printing paths have some effect on the produced object strength, and hence a designer might wish to choose strength over surface finish through tool-path tuning. This section discusses these aspects, aimed at optimizing the manufacturing process itself, rather than the produced shape.

7.1. Tool-Path Optimization

One of the most fundamental problems in manufacturing process optimization is the tool-path problem, i.e. planning the order in which the manufacturing device's end effectors (e.g. extractor, milling bit, cutter etc.) are used and the paths along which they travel. This planning is critical for the quality and efficiency of the process. The factors that must be considered depend on the specific device, but usually share a few common concerns. To reduce manufacturing time, minimizing idle motion (where the end effector changes position without affecting the shape) is always a main concern. Path smoothness is another important example, since sharp turns usually require the end effector to slow down, can degrade quality due to discretization artifacts, and can even increase the chance for fracture. For many applications, transitioning from idle to active is also an important factor, since this results in some overhead in terms of time or material and could also affect quality.

This fundamental problem has been addressed extensively in the past, and solutions for it are rather mature [GRS14]. Nevertheless, this is still an active field of research, where new considerations and manufacturing technologies are brought into play. For example, an algorithm has been proposed to determine the printing order and head orientation for wireframe 3D printing on a 5DOF printer (a regular layer based 3D printing with a 2-axis tilting tray) [WPGM16]. This method first constructs a collision graph of the nozzle and the printed object. Then it determines a feasible printing order and orientation to avoid collisions while encouraging smoothness, enabling wireframe printing for a large class of shapes. Zhao et al. [ZGH*16] give special attention to continuous printing. Using the proposed method, a tool-path is planned for printing any connected component with one curve, based on Fermat spirals. A distance transform is first computed over the inside domain of a given slice, and iso-contours are extracted. Then, a tree structure is built for the disconnected iso-contours of the same value. A Fermat spiral-based space filling pattern is computed that fills each tree component, and connects all parts into one curve. This formulation follows the contour of the object, improves boundary quality, introduces fewer sharp turns, and completely eliminates idle head movement.

Sitthi-Amorn et al. [SARW*15] present a method to update the additive manufacturing process on-the-fly in order to account for inaccuracies. The system presents a 3D printer with support for multiple materials (up to 10), low cost (roughly USD 7000), but high accuracy (40 μm). The input is a 3D voxelized model with a material assigned to each voxel. After a few layers are printed in a traditional way, the system performs a quality check via 3D scanning. This enables the planning of a new correction layer based on the depth map. This adaptation allows for much greater accuracy for the same cost, and can also be used to print on top of previously printed parts. The latter opens up new possibilities, such as taking a piece out in mid-print and embedding objects within it (such as electrical circuits) before resuming the printing process.

As can be seen, to date the optimization of tool paths has been disconnected from the design step, even though surface finishing and visual quality can be affected greatly by the choice of the tool path. For example, in metallic additive manufacturing, it is well known that the melting beam's scan strategy has significant effect on the resulting mechanical properties [CMWA14]. Thus, involving the designer in this step, and integrating it with the rest of the design process is likely to have merit. This, however, would probably require a study to better understand the mechanical effects induced by the manufacturing tool-paths.

7.2. Time and Material Usage Optimization

Perhaps the most limiting factor on the growth of additive manufacturing to date is printing time. If the printing time could be reduced to seconds or minutes, instead of hours or days, a wide variety of new applications could become available, such as iterative or interactive manufacturing, more elaborate trial and error, etc.

Responding to this observation, Müller et al. [MIG*14] propose low-fidelity fabrication of a desired shape, in the form of a wireframe. To do this, the shape is sliced, and respective contours are extracted. The slicing can be regular, but can also adapt according to shape features. The gaps between slices are filled with a zigzag pattern, carefully designed for low-end FDM printers, with minor hardware adaptation (such as active cooling, to ensure the thin strand of material solidifies quickly enough before bending down). Similarly, Wang et al. [WWY*13] propose to convert a part of the mesh to a truss-and-skin structure, for material usage efficiency and reduced printing time. An optimization runs directly on strut nodes, to minimize strut volume and number in an alternating fashion (node positions are also allowed to change), while still withstanding internal and user-provided external forces. During the optimization, strength is analyzed through FEM simulation for the skin, and through a beam formulation for the struts.

Another approach to reduce printing time is through adaptive layer thickness. In traditional additive manufacturing, a given shape is sliced regularly, and the printing path is determined for each slice independently at the highest resolution. Wang et al. [WCT*15] propose slicing the mesh with adaptive layer thicknesses to reduce printing time. Thicker layers imply extruding more material at a given point, increasing speed while reducing resolution. The proposed method computes shape saliency (using existing work, employing spectral decomposition), and assigns each maximum resolution layer a visual quality score based on saliency, layer height and mesh angle. Then, an optimization process tries to minimize the amount of non-empty layers by controlling layer height, while considering the aforementioned score.

These approaches yield reduction factors of 2× to 10× in terms of printing time and material consumption. The variance, of course, comes at the cost of visual or structural degradation. While this improvement is welcome, further optimizations must be made in order to really achieve rapid design iterations. The ideas presented here, such as experimental low-fidelity printing and adaptive resolution are definitely steps in this direction, but further improvements are likely to originate from combined software and hardware systems better adapted for draft production.

7.3. Object Partitioning and Print Volume Optimization

For additive manufacturing, in almost all cases material is deposited on top of the partially existing object, or a support structure if nothing should be printed below. In addition, the printing volume is typically quite limited, due to physical constraints. Therefore, partitioning the print object into several parts is an efficient way to reduce print time, support requirements, and accommodate larger prints.

Taking this approach, Luo et al. [LBRM12] propose to decompose an object into smaller volumes that fit within the printing volume constraints of a given device. To do so, BSP trees are utilized to find planar cuts via beam search, considering several objectives such as number of parts, connector feasibility, aesthetics, and structural soundness. Voxels and tetrahedra are used for structural analysis via FEM, and a distance field inside the object is constructed to check whether connector addition is feasible. The partitioning is done recursively, and stops when all parts fit in the printing volume. An illustration can be seen in Figure 9.

Vanek et al. [VGB*14b] propose to solve the partitioning and packing problems together. Given a mesh, it is hollowed to a thin shell, which is tetrahedralized. Random seeds are picked to grow regions within the shell. These are merged to reduce their number and cross-sectional area, while maximizing volume. Then, a combinatorial optimization is run, ordering the pieces in space to minimize print volume and support-material usage. The optimization stores the "height horizon" (or height field) of each configuration as the objective. Later, another partitioning and packing solution was suggested, which takes printability, high stress regions and mesh features into account [YCL*15]. In this method, the input mesh is first tetrahedralized and a stress analysis is run on it. Then, the volume is partitioned by a grid, and the stress levels are assigned to the nodes. The mesh is segmented based on geometric features, and a signed distance field is constructed around each segment (on the grid nodes). These fields are used in an optimization to minimize packed height and cut-quality penalty in three iterative individual steps. Recently, Song et al. [SDW*16] proposed to allow for large efficient prints by connecting printed high-resolution parts and low-resolution interlocking planes, manufactured through faster and cheaper technologies such as laser-cutting.

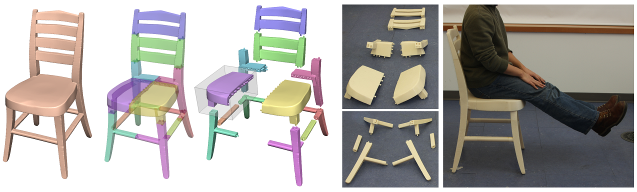


Figure 9: *In order to allow for fabrication of objects larger than the printing volume, a partitioning method is proposed to cut and connect the pieces in a structurally sound way [LBRM12].*

Hildebrand et al. [HBA13] suggest that the orientation in which a part is printed affects its properties. Since, as discussed earlier, the printing technologies are anisotropic, the vertical axis typically induces less resolution and strength. Therefore, different parts of the mesh are best printed in different directions. The proposed method cuts an input mesh along split planes to fabricates each part in the best of the 3 orthogonal orientations. The space is voxelized, and the error is measured on the boundary voxels. An optimization process finds a set of split planes that minimizes the error and number of parts.

A different approach employs generalized pyramids, which are flat-based structures with the remaining boundary forming a height function over the base. Such shapes require no support structure when printed, and are similarly optimal for various other manufacturing processes. Therefore, decomposing a shape into pyramidal parts would yield a partitioning that is support-free. Since the pyramidal decomposition of an arbitrary shape is an NP-hard problem, and even when solved would still yield too many parts for printing applications, a method for approximate pyramidal shape decomposition has been proposed [HLZCO14]. A uniformly sampled point set is chosen and each point votes for its best base from a few predetermined directions. Similarly-voting regions are merged into *cells*, which in turn are merged into *blocks*, according to an affinity measure. The best decomposition is then sought through the reduction of the problem to the NP-complete *Exact Cover Problem* (ECP), which is solvable in practice. Similarly, Chen et al. [CZL*15] propose to decompose a shape into pyramidal parts as a solution to the partitioning and packing problem introduced earlier, reducing support material usage and printing time.

Object partitioning is mainly done using spatial partitioning representations, as can be expected. Segmentation and saliency detection play an important role in semantic partitioning, enabling more intelligent decisions regarding cut placement. Similar to the previous problems presented in this section, involving the designer in the partitioning process would probably be beneficial in the future, even though the methods described here are primarily automatic.

7.4. Support Optimization and Support-Free Structures

As mentioned earlier, almost all additive manufacturing processes require a support structure. This is because for most technologies the manufacturing process involves depositing (or curing) material on top of an existing surface. Therefore, any point on the surface that is not above another, up to some device-specific inclination, requires a disposable structure to be built underneath it for temporary support. Other means of support also exist for other means of fabrication, such as chains or formwork for intermediate steps of architectural construction. In this section we will explore support usage optimization and the design of support-free shapes. Of course, much like in the tool-path case, this problem has already been thoroughly investigated [CJR95, HYML09, JBAG12, HHY*13, JHF15]. Even so, some novel ideas still emerge from time to time, as can be seen in this section.

In order to reduce material usage for support in FDM printers, a tree structure was proposed [VGB14a]. First, the orientation that results in the fewest overhangs for a given input mesh is selected. In this orientation, required support locations on the mesh are identified, and the space of potential support structure locations is represented as a cone. A heuristic plane-sweeping algorithm is employed to generate tree-like support structures from top to bottom, connecting at intersection points of several cones. The generated support structure consists of cylindrical struts, with an N-shaped profile, balancing printing speed (and material usage) and support strength. The most computationally heavy part in the algorithm is the cone-to-mesh intersection, which is therefore done on the GPU. Later, Dumas et al. [DHL14] proposed a method for automated computation of steady scaffoldings. Inspired by scaffoldings in construction, where only horizontal bars and vertical pillars are used, they show that a bridge structure is more stable than tree-structured supports with comparable amount of printing material. Slicing is used to determine support points. Stability while printing is ensured by checking center-of-mass and base-of-support in a bottom-to-top manner. Integer programming over a regular grid failed to find optimal bridges, so a greedy algorithm with a sweep strategy is used instead. Short slanted connectors are allowed only at the top of supporting structures, to better interface to the printed surface.

Of course, the most efficient support structure is no structure at all. The design of support-free structures, such as the previously mentioned pyramidal structures (Section 7.3), has been studied both in the context of architectural design and additive manufacturing. In the case of architectural design, typical support-free structures are vaults and domes. Even though these structures have been constructed since ancient times, modern FEM based analysis methods, which are the most common ones in industry and academia, are unsuitable for predicting their stability [Blo09]. To facilitate the analysis of a wide class of such structures, an elegant formulation has been developed, named *Thrust Network Analysis* (TNA) [BO07]. This method discretizes the forces acting within the structure by constructing a graph, which ensures structural stability when a few intuitive conditions are satisfied for all the graph nodes. Since its introduction, this method has been employed by several design approaches [VHWP12, dGAOD13, TSG*14].

In the context of additive manufacturing, Panozzo et al. [PBSH13] propose a method to find self-supporting structures that are as close as possible to a given target surface. To achieve this, the terms dictated by TNA must be satisfied. The forces are discretized in a smooth and uniform manner, according to mesh features and other structural heuristics. A careful discretization is crucial for result quality, and is facilitated through the construction and interpolation of a cross-field over the target surface. Then, the surface that is closest to the target while still adhering to the constraints described earlier is sought through gradient-descent optimization. Finally, the structure is tessellated in a way that ensures stability. The tessellated result does not take traditional architectural manufacturing constraints into account (such as mold generation costs, as described earlier), and hence is targeted at additive manufacturing and simulation only. Later, Deuss et al. [DPW*14] proposed a method to reduce the support structures required during the actual construction process of such structures. This is done by identifying self-supporting (or almost self-supporting) linear chains of bricks within the design, which if built first, can already support other parts during construction. Recently, another extension to this approach was proposed, supporting the use of freefrom representations, instead of explicit elements, through the use of Airy stress functions and a carefully designed set of algebraic rules [MIB15].

Unlike the problems previously described in this section, the design of self-supporting structures is already being solved with the help of the designer. This is due to the fact that self-supporting shapes are a small subset of the design space, and thus significant and noticeable changes to the geometry must be made to accommodate such requirements. As can also be seen, support-related analysis is done mainly through spatial partitioning and graph representations. This is because the forces inside the object and support structure flow in a graph-like manner, usually in a tree configuration.

8. Underlying Representation Analysis

The work covered by this survey employs many different geometric representations to encode shape, materials, motions, hierarchy, abstractions, and annotations. Of course, not all fabrication applications use the same geometric representations. Some work with surfaces, while others work with curves or volumes. Some use regularly sampled representations (e.g., voxels), while others use irregular meshes. Another important aspect is the information attached to the shape, along with the representation. These *attributes* can relate to physical properties of the material (e.g, elasticity or appearance descriptions) or the geometry of the object (e.g. its curvature). The choice of geometric representation and accompanying attributes depend on computational properties required by the application.

In this section, we analyze which geometric representations and attributes are most commonly used for fabrication applications. For each representation, we discuss its computational properties (e.g., what types of computations are most efficient with it) and describe how those properties are leveraged in different fabrication applications. For each attribute, we examine how and why it is or is not used with respect to the applications. In doing so, we hope to provide information to help researchers choose geometric representations for future fabrication projects.

Our analysis focuses on relating computational properties of geometric representations to the requirements of fabrication applications. Theoretically, almost every geometric representation is able to specify almost any shape and perform any geometric operation if given infinite storage and compute time. For example, a voxel grid can represent any 3D function at any resolution with infinitely small voxels. However, in practice, there are significant trade-offs in the accuracy, or *conciseness*, and computational efficiency of different representations. Some require more storage for the same level of accuracy (e.g., voxels usually require more storage than irregular grids). Some representations naturally represent different *ranges of shapes* (e.g., triangle meshes would describe a cube more easily than a sphere). Others enable more efficient computation of *Boolean operations* (e.g., voxels allow immediate volumetric intersection computations), which are important for fabrication-aware design. The ability to compute *differential geometric attributes*, such as curvature, principal directions, or vector fields, is also an important factor in choosing a representation (e.g. algebraic representations provide analytic derivation of them). Similarly, often times a design needs to specify other, *non-geometric attributes* on the object (such as elasticity coefficients), and a representation should support intuitive specification, and interpolation to unspecified regions. Lastly, another important property in the context of fabrication is the ability to perform *physical simulation*. Table 10c indicates how well each discussed representation performs in these aspects.

In the following parts, we consider the trade-offs in properties provided by different geometric representations and discuss how they affect their suitability for different fabrication applications. Table 10a provides a summary of the ways in which different geometric representations (columns) have been used for different fabrication applications (rows) in work covered in this survey. The cells indicate the ratio of surveyed papers that use a given geometric representation for each fabrication application (i.e., the number of papers that use the representation divided by the total number of papers for the given application). This ratio is both explicitly written in each cell and is depicted by its color intensity (redder is higher). Row sizes indicate the quantity of papers discussing a solution to the corresponding problems. Table 10b further illustrates attributes that were assigned or computed over the objects, displayed in the same manner of ratios of solutions to the corresponding problems.

The remainder of this section discusses columns of these tables. We will briefly introduce each representation (Sections 8.1-8.4) or attribute (Section 8.5) type, discuss its properties, and analyze how and why it is used in different fabrication applications.

8.1. Solid Representations

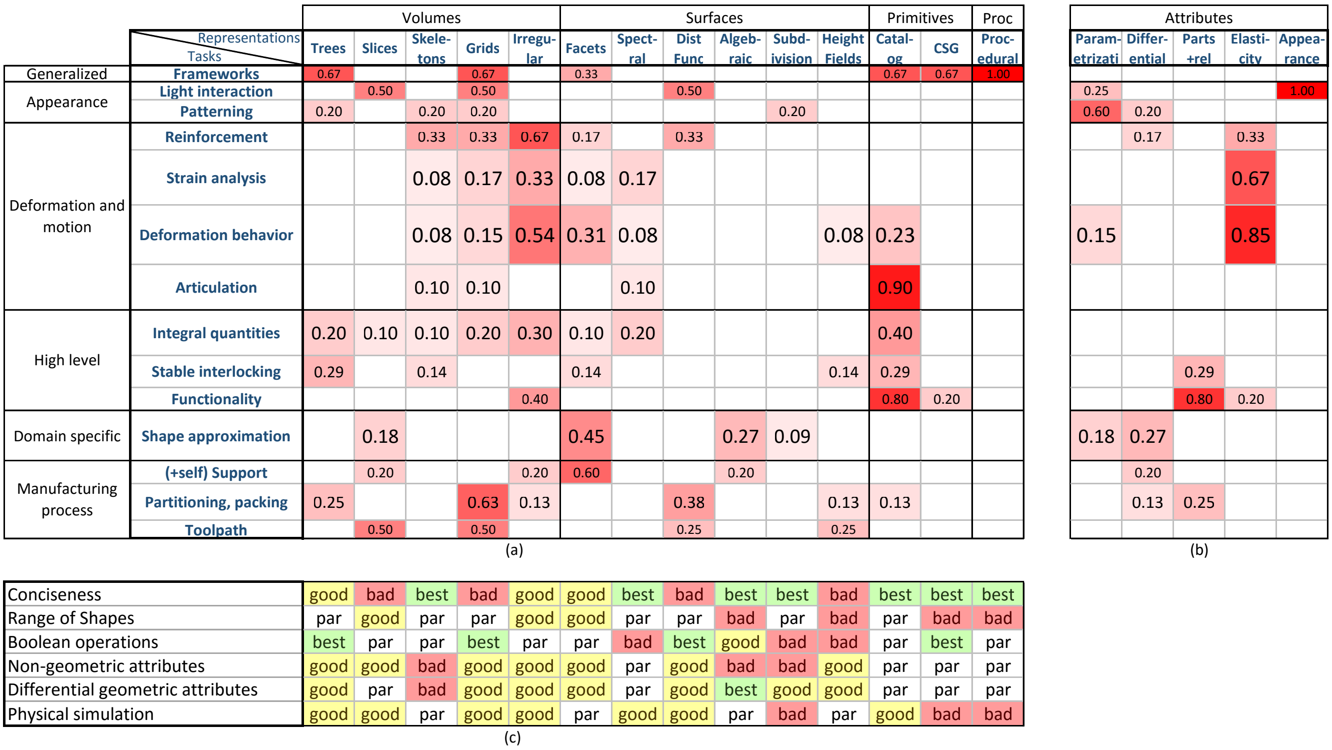
Irregular: Irregular mesh representations offer a great balance between level of detail and accuracy, linearly approximating all regions which are not explicitly specified. For this reason, it is the most common representation for solids in this survey. On the other hand, it is less convenient for data interpolation or Boolean operations, which benefit from regularity. This representation includes several flavors. Its simplest and most common form consists of tetrahedra [SVB*12, BKS*12, LBRM12, ZPZ13, WWY*13, STC*13, LSZ*14, BWBSH14, CZXZ14, PTP*14, VGB*14b, DLL*15, SBR*15, PZM*15, XLCB15, CBNJ*15]. Tetrahedra, lacking regularity and structure, are almost exclusively employed for physical simulation of general objects, with FEM or simpler, more direct forces based methods. Cuboids simplify these computations, on the account of expressiveness, and are especially fitting for furniture design, since most parts are cuboid shaped planks anyway [UIM12, KLY*14]. General polyhedra are only in used when an unstructured partitioning of the space into cells occurs, such as one created by a Voronoi decomposition [LSZ*14, MDL16, PBSH13].

Grids: Dividing the volume into regular cells is perhaps the most intuitive way to partition a domain. In this scenario, each cell may contain the density of the scanned material or many scalars, forming a data tensor, for the case of spectroscopy or Doppler scans [CWIM*03, TVK*12]. In the context of fabrication, grids are also a natural choice since in many cases they match the manufacturing technology, in which each voxel in space can independently be void or assigned a specific material. The regularity of this representation facilitates integration, interpolation of attributes, tiling, frequency analysis, derivation and many other optimization based applications. For these reasons, it is found in computational fabrication work almost as frequently as irregular meshes. It is used for printer level descriptions [LBRM12, HBA13, HL14, VGB*14b, LSZ*14, PRM14, BAU15, SARW*15, YCL*15], multi-grid solving [WDW15, DLL*15], to facilitate tiling [BBO*10, SBR*15, LLMZ16], or other high-level optimization goals [LOMI11, PWLSH13, ZSMS14]. On the other hand, regular grids are not adaptive, and hence demand a lot of memory as the required resolution increases. In addition, they cannot feasibly represent smooth shapes and introduce local artifacts, making them inappropriate for geometric operations and physical simulation.

Trees: Recursive spatial partitioning through tree structures is an efficient way to reduce the burdening memory consumption requirements of grids. This representation suffers from the same disadvantages that grids do, and pays for its conciseness with more involved neighboring queries. In the context of fabrication, Binary Space Partitioning (BSP) trees are naturally used for object partitioning [HBA13, LBRM12], but also for assemblability analysis of planar structure designs [HBA12]. Octrees are primarily used as multi-resolution versions of 3D grids. Their adaptivity is exploited to reduce memory usage of large homogeneous regions [BWBSH14]. They are also used [DLL*15] to propagate parameterization information across levels for a multi-grid solver. As can be seen, this representation is very sparsely used, in spite of its clear advantage over grids. As fabrication resolution continue to improve, these data structures are likely to be required more frequently, due to increasing memory loads.

Skeletons: Skeletons are thin structures that are typically equidistant to a given shape's boundaries. The skeleton usually emphasizes geometrical and topological properties of the shape [TDS*16]. In the context of this survey, two versions of skeletons have been used: rods and beams, and medial axis.

Rods and beams can be represented as a center-curve, and twisting angles. This representation facilitates physical simulation by dramatically reducing dimensionality, and hence is preferable whenever it applies, as can be seen in reported publica-



		Volumes					Surfaces						Primitives		Proc
	Representations / Tasks	Trees	Slices	Skeletons	Grids	Irregular	Facets	Spectral	Dist Func	Algebraic	Subdivision	Height Fields	Catalog	CSG	Procedural
Generalized	Frameworks	0.67			0.67		0.33						0.67	0.67	1.00
Appearance	Light interaction		0.50		0.50				0.50						
	Patterning	0.20		0.20	0.20						0.20				
Deformation and motion	Reinforcement			0.33	0.33	0.67	0.17		0.33						
	Strain analysis			0.08	0.17	0.33	0.08	0.17							
	Deformation behavior			0.08	0.15	0.54	0.31	0.08				0.08	0.23		
	Articulation			0.10	0.10			0.10					0.90		
High level	Integral quantities	0.20	0.10	0.10	0.20	0.30	0.10	0.20					0.40		
	Stable interlocking	0.29		0.14			0.14					0.14	0.29		
	Functionality					0.40							0.80	0.20	
Domain specific	Shape approximation		0.18				0.45			0.27	0.09				
Manufacturing process	(+self) Support		0.20			0.20	0.60			0.20					
	Partitioning, packing	0.25			0.63	0.13			0.38			0.13	0.13		
	Toolpath		0.50		0.50				0.25			0.25			

(a)

		Attributes				
		Parametrization	Differential	Parts +rel	Elasticity	Appearance
Generalized	Frameworks					
Appearance	Light interaction	0.25				1.00
	Patterning	0.60	0.20			
Deformation and motion	Reinforcement		0.17		0.33	
	Strain analysis				0.67	
	Deformation behavior	0.15			0.85	
	Articulation					
High level	Integral quantities					
	Stable interlocking			0.29		
	Functionality			0.80	0.20	
Domain specific	Shape approximation	0.18	0.27			
Manufacturing process	(+self) Support		0.20			
	Partitioning, packing		0.13	0.25		
	Toolpath					

(b)

Conciseness	good	bad	best	bad	good	good	best	bad	best	best	bad	best	best	best
Range of Shapes	par	good	par	par	good	good	par	par	bad	par	bad	par	bad	bad
Boolean operations	best	par	par	best	par	par	bad	best	good	bad	bad	par	best	par
Non-geometric attributes	good	good	bad	good	good	good	par	good	bad	bad	good	par	par	par
Differential geometric attributes	good	par	bad	good	good	good	par	good	best	good	good	par	par	par
Physical simulation	good	good	par	good	good	par	good	good	par	bad	par	good	bad	bad

(c)

Figure 10: *A breakdown of representation (a) and attribute (b) types that are employed by the covered work, according to the problems they are used to solve. Color intensities, and cell values, indicate the ratio of solutions that use a specific representation or attribute out of all solutions for the relevant problems. Note that a certain solution might employ more than one representation, and hence the ratios do not sum up to* 1.0*. Row (and font) size indicates the number of solutions proposed for the same problem set. (c): Performance of each representation with respect to key properties required by computational fabrication related applications.*

tions [WWY*13, ZCT16]. Considering the elongated shape's cross-section has been shown to further provide additional flexibility during optimization [PTC*15].

The medial axis is a well studied representation, consisting of a skeleton and offsets from it. This representation does not directly lend itself to most of the operations required in computational fabrication applications, and therefore is not commonly employed. It can, however, be used to reduce dimensions and dictate topology. In some applications, the medial axis is employed to perform geometry processing operations, such as deformation [CZX*16], and identification of thin parts [SVB*12]. In others, it is elegantly used to prevent self collisions during object deformation or thickening [BBJP12, MAB*15, LFL09].

Slices: Many 3D printing technologies perform the fabrication process by depositing the material layer by layer. This requires determining the shape's boundary on each layer. For maximum quality, the layer thickness is determined by the manufacturing resolution. These needs give rise to an application specific representation by *slices*. The shape is divided into a, typically regularly distributed, set of parallel planes, aligned with the printing direction. On each slice, the shape is typically represented by closed curves (or *contours*), or a 2D grid. This representation does not offer accuracy, conciseness or ease of processing, and is usable strictly for guiding the printing process, as can be deduced by its usage [RCM*14, HL14, MIG*14, DHL14, WCT*15, BAU15, ZGH*16].

8.2. Surface Representations

Facets: Facets based representations, or polygonal meshes, are nearly ubiquitous in computer graphics applications. Most rendering hardware, for example, is designed to handle triangles alone. Therefore, triangle meshes are the form of input to almost all the work mentioned in this survey, excluding architectural applications. A significant number of papers, however, does not require the properties of this specific representation, and use triangle meshes simply out of convenience. In our analysis, we tried to distinguish this type of usage from algorithms that explicitly require a facet based representation, and reported only the latter in Figure 10.

Much like their volumetric counterparts, polygonal meshes offer a good balance between level of detail and accuracy. This facilitates most of the discussed computational requirements: linear interpolation of attributes specified on vertices, physical simulation of linear elements and marking semantic part association and relationships on the elements themselves. For the same reasons, geometric features are more difficult to derive with this representation, however it is so common that many tools have been developed to this end.

For these reasons, it is the most common reported surface representation.

This representation also includes several forms, of which triangles, much like tetrahedra, is the simplest and most common one [MI07, SS10, STBG12, VGB14a, DPW*14, STK*14, CZX*16, CS16], used for its ease in surface FEM or membrane simulations. Quadrilaterals are preferred for some applications, especially when nearly flat or rectangular, since they match principal directions of the shape and most sampling patterns more naturally. In the context of architectural fabrication, quadrilateral production is often preferred since it facilitates mold reuse [FLHCO10, LPW*06, PBSH13]. Hexagonal polygons were shown to produce specific mechanical advantages, such as compliance to stretch and shear [PTC*15], or extra strength in grid-shell structures [PTP*14]. More elaborate general polygons were used where structure was not a concern [UKSI14, CS16], e.g. to guide a laser-cutting operation.

Spectral: Spectral mesh processing is a powerful tool for applications such as filtering, shape matching, remeshing, segmentation, parametrization and many others. In essence, it is an extension of the classical Fourier transform to irregular grids, and typically involves an eigendecomposition of a linear operator over the input mesh [ZVKD10]. This representation lends itself to detail and resolution reduction, due to direct control over mesh frequencies. On the other hand, it does not allow direct geometry manipulation, including needs such as Boolean operations or physical simulation. Hence, in the context of computational fabrication, it can be used only for specific tasks. All reported usage takes advantage of the unique properties of this representation to substantially reduce computational efforts in non-trivial ways. Modal analysis has been creatively employed to identify weak spots in a design, eliminating the need for explicit FEM simulation, through computing the response over different vibration frequencies [ZPZ13, XLCB15, ZCT16]. Manifold harmonics has been successfully employed for dimension reduction in the context of integral quantities optimization, enabling low-frequency modification (allowing for integral control) while leaving high-frequency details intact [MAB*15]. In addition, spherical harmonics were employed to extend a data-driven aerodynamics model to unmeasured directions [MUB15].

Dist func: Representing a shape by marking the space with the distance from it has several advantages. For example, Boolean operations are most natural under this scheme, since they can be performed locally and directly on the functions. Perhaps the biggest advantage, which was exploited in the context of this survey, is that this representation is agnostic to topology changes. Minimum thickness is a very common requirement in manufacturing processes. Defining the shape as the level-set of a prescribed distance ensures this requirement is fulfilled and allows for direct tessellation regardless of the topology it induces [SVB*12, LBRM12, LSZ*14]. The same advantage has been exploited to guide a partitioning optimization [YCL*15]. This representation has also been used to trace out curves along the space, attempting to have them as long as possible, but again with no knowledge of the resulting topology [PRM14, ZGH*16]

Algebraic: Algebraic representation is a different form of implicitly describing a surface, by analytically defining relationships between its coordinates or other properties. Most surfaces cannot be represented in such a clean manner, however it is highly desirable to do so if possible. This representation is concise, elegant, and most importantly can be analytically derived, which has been leveraged in fabrication applications to significantly improve gradient based optimization [EKS*10, MIB15]. Special constructs like Chebyshev nets [GSFD*14] and discrete conformal geometry [KCD*16] have been elegantly adapted to model specific material behavior, facilitating editing operations otherwise unfeasible. These examples demonstrate how such representations should be sought out and used whenever viable.

Height Fields: Height fields typically represent a surface using a regular planar grid, where each node stores the respective distance of the surface from the plane, in the normal direction. Being regularly sampled, this representation does not offer conciseness. In addition, it cannot directly represent all surfaces, since it stores only one scalar per node. For these reasons, this representation is useful only to facilitate optimization of said height [SARW*15, VGB*14b, BKS*12, LFL09].

Subdivision: Subdivision surfaces are smooth free-form surfaces which are generated using recursive rules [Cas12]. This formalism allows the definition and editing of smooth surfaces in a concise manner through the use of a control mesh. While this is very effective for animation and CAD applications, subdivision surfaces lack utility when physical simulation or geometry processing operations are concerned. For this reason, this representation is unfortunately mostly avoided in the fabrication-aware context, even though it is very common to computer graphics. Liu et al. [LPW*06] have adapted their proposed method to integrate fabrication-aware design within subdivision systems in architectural applications, potentially creating a powerful design tool. Zehnder et al. [ZCT16] have exploited the inherent smoothness of this concise representation to guide the elastic deformation of curves over the represented surface. Recently, basic external calculus operators for subdivision surfaces have been developed [dGDMD16]. This allows for a wide variety of geometric operations to be performed on the control mesh, significantly reducing computation time for many applications. Hopefully this approach, and others like it, will hasten the integration of fabrication-awareness in subdivision based design.

Parametric: Parametric surfaces, such as NURBS, or other splines [SZBN03, Pro05, WHL*08] offer great ease in modeling and editing smooth surfaces. Similar to subdivision, they are extremely common in CAD and architectural systems, but lack usability for fabrication applications. For this reason, parametric surfaces are seen in this survey only as input in architectural application, and are almost immediately tessellated. Recent advances in isogeometric analysis, mentioned in Section 6.2, enable physical simulation on NURBS based representation, and might facilitate the use of these representations in a fabrication-aware context.

8.3. Primitive-based Representations

CSG: In Constructive solid geometry (or CSG), relatively simple primitives are combined by a tree of Boolean and transformation operations that are included directly in the representation. The attractive properties of CSG include conciseness and convenient Boolean properties. Perhaps the biggest advantage of this representation is the powerful control over the shape through high-level parameters, defined on leaves and internal tree nodes. This, along with simple data structures and elegant recursive algorithms makes this representation arguably the most popular one in CAD applications. Unfortunately this flexible representation is not popular in the computer graphics community since it is cumbersome for any geometric procedure or simulation, and hence is not commonly encountered in this survey. Some publications were able to leverage the parametric nature of this representation, increasing flexibility of pre-defined library primitives [KSS*15, SSM15]. It is possible that this representation will gain popularity also within the field of computer graphics, thanks to its ease of use for designers, as evidenced by generalized frameworks, which do accept it as input [VWRKM13, SSM15].

Catalog: While CSG representations usually consist of simplistic primitives, the use of elaborate pre-defined or pre-computed elements is much more frequent in this report. The classic use-case of predefined elements is designing with already manufactured, off-the-shelf parts [LOMI11, SSL*14]. The more interesting use is constructing a library of parameterized elements, or *database*. This has been repeatedly demonstrated to be effective in constraining an otherwise unfeasible design space, for man and machine alike. It seems that to date there is no automatic way to synthesize articulated motion without manual definitions of joints [CCA*12, BBJP12, BCT*15b, FSY*15, KLY*14, KSS*15, LBRM12, MTN*15, TCG*14, ZSMS14] and other mechanisms such as gears and pulleys [CLM*13, CTN*13, ZXS*12], employed for specific pre-assigned cases. Other interdependent mechanisms such as tensegrities and burr puzzle cores also seem to elude automatic generation, and must be pre-configured to allow their design [GCMT14, XLF*11]. Another approach starts by perturbing and measuring the response of fundamental elements, allowing for more elaborate optimization in a later step, which instantiates and modifies them. This effectively constrains the design space, making it feasible for optimization and human manageability [MUB15, SBR*15, PZM*15, LLMZ16].

8.4. Procedural Representations

Procedural: Defining shapes programmatically can be a powerful tool. Potentially, extremely complex structures can be represented by a few lines of code, if they incorporate some reasoning. As manufacturing technologies evolve, designs will probably be spanned over several orders of magnitude, from process resolution through micro-structures to complete designs. Explicitly describing these details is obviously unfeasible. For example, a small cube manufactured by weaved wires even today requires gigabytes of storage to be explicitly represented. Hence, procedural based representations, where parts of the shape can be generated on-demand, are likely to become popular. To date, however, such representations are not in use, due to the overhead they impose, and the current capability to describe most designs in full. Generalized frameworks [VWRKM13, CLD*13, SSM15] do offer a programmatic interface to describe shapes, enabling stream-lining information to the manufacturing device, while bounding memory usage.

8.5. Attributes

Throughout the different solutions reported in this survey, different attributes are assigned, propagated, interpolated and manipulated over the designed objects. The usage of some of these attributes is as expected. For example, **appearance** properties such as BRDFs, translucency profiles or colors, are assigned to objects as targets or measurements, and used in optimization processes strictly for plenoptic applications [RCM*14, BAU15, PRJ*13, PRM14]. Similarly, **elasticity** properties, such as Poisson's ratio, Young's modulus or parameters for other physical models (continuum mechanics, Hart-Smith, etc.), are used solely for elastic physical simulation [BBO*10, BKS*12, LBRM12, STBG12, SVB*12, STC*13, CZXZ14, LSZ*14, DLL*15, GCMT14, KSS*15, CBNJ*15, MDLW15, SBR*15, PZM*15, PTC*15, XLCB15, WDW15, MDL16, CZX*16]. Interestingly, not all solutions for tasks that seemingly mandate elasticity actually use it. Some strain analysis solutions avoid direct elasticity simulation through modal analysis [ZPZ13, ZCT16], or geometric properties such as cross-section dimensions [US13]. Even some applications aimed at controlling deformation behavior elude physical simulation by exploiting geometric observations, in order to facilitate real-time response [MI07, STK*14].

Attributing semantic region assignments, or **parts**, aids in aesthetic shape decomposition, since important features are not disconnected [XLF*11, VGB*14b]. Evidently, it also aids in understanding and specifying object functionality, since semantically similar regions also probably have similar functional goals [LOMI11]. Further specifying **relationships** between parts (e.g., one part should fit in another, cover another, or support another) has been demonstrated to be very beneficial in the design of functional objects, since it allows for many automatic functionality preserving adjustments to be done during the design process [LOMI11, UIM12, KLY*14].

Differential features are also computed and assigned to surfaces. Most geometric attributes, such as curvatures and local shape features, are used to guide semantic shape partitioning or approximation, as can be expected [HBA12, PBSH13, YCL*15]. Vector fields are a well studied tool that provides global properties, such as smoothness or singularities minimization, to functions defined over a surface, while still conforming to local features and principal directions of the shape. This property is nicely leveraged to improve structural stability in the architectural context, where vector fields smoothly interpolate between local stress directions [PTP*14, PBSH13]. They are also elegantly used to trace curves on a surface, being as long and as smooth as possible on one hand, but conform to principal directions (or other directions, prescribed by the user) on the other, in the context of planar structures [CPMS14]. This unique property is likely to be exploited more in the future.

The mapping of a 2D plane to a surface, or **parameterization**, is also employed in various ways. In a classical manner, it is used for texture mapping, in the context of pattern synthesis [DLL*15], and for quadrangulation [LFL09]. Such a mapping is also used to dictate light routing through the volume [PRM14]. Another use that is unique to fabrication, is for developable patches. Developable patches are important to mimic the properties of materials which can be bent, cut or folded, but not stretched. Manufacturing processes which deform such planar sheets usually involve fabric or metal. For these application, both the deformed 3D shape and the sheets from which it is created are relevant to the designer, and are typically directly editable. Hence, maintaining a good mapping from one to the other during the design process is an important task, addressed through parametrization several times in this survey [GSFD*14, KCD*16, MI07, STK*14].

9. Discussion

Computational manufacturing, and especially additive manufacturing, is still a very young field. Nevertheless, it has already revolutionized the way objects are prototyped, designed, and even manufactured. As can be seen in this survey, this field gives rise to an abundance of novel or revised applications, most of which have probably not been conceived yet. This leap forward has left a gap with respect to design and representation technologies and methodologies. To fully realize the potential of new manufacturing technologies, a fundamental change in design concepts will probably need to happen. Traditional design and representation approaches assume smoothness wherever not explicitly specified otherwise (e.g the linear interpolation of triangles). This is reasonable when added details induce added cost. This, of course, is not the case for the aforementioned technologies, where practically arbitrary details can be produced with no tooling. Therefore, it is likely that hierarchical or programmatic representations, seen in recent publications, will become more popular in the future.

One of the biggest problems of designing in great detail is the limited complexity graspable by humans. Therefore, exploring the full design space is not a feasible approach. A possible solution is designing through goals. Ideally, specifying the desired objectives, their trade-offs and constraints, should be sufficient to synthesize the desired design. In this scenario, the designer will have many objectives to chose from, and the difficult task would be to balance them correctly. This will alleviate the need for the designer to grasp all the necessary details of a valid design.

For such a system to be useful, many objectives will probably need to be developed. In this survey, some objectives have been introduced, with methods to satisfy them (e.g. articulation by description, deformation behavior, balance, aerodynamics etc.). These objectives can be dictated by professionals, or deduced from large collections. Some of these objectives are more difficult to manage. Aesthetics, for example, is a major concern for almost any design, but a very elusive one. While this mainly concerns perceptual studies, some quantification for it has been attempted for applications such as integrity control, shape approximation, articulation, and architecture. These were addressed by heuristics, measuring symmetries, saliency, smoothness, resemblance, etc. A pressing issue would be to further explore these quantifications, enabling better guidance to optimization processes.

Another such objective is motion. This objective can be rather intuitively specified through motion curves or designated constructs such as rigs and kinematic chains. Realizing it, however, is more challenging. Throughout this survey, motion was realized through the use of primitives, such as joints, gears, and pulleys, which were manually pre-configured. In the case of deformation control, primitives were also employed, to provide more flexibility. In some approaches, however, these were automatically generated and explored, rather than manually defined beforehand. The notion of exploiting reusable motion and deformation primitives is a promising one, especially as manufacturing resolutions increase. It would be interesting to investigate whether this concept can be applied to other objectives, and whether primitives could be automatically evolved for them. Note that solutions to these problems would also have to address practical aspects, such as tiling the elements in a non-restrictive manner, representing them efficiently, and simulating a design with many instantiations.

As the number of introduced objectives increases, one of the difficult tasks that designers will have to face is determining their trade-offs. To date however, most publications address and solve a specific goal, rather than combine multiple objectives and balance them. Problems relating more objectives, and their combinations, will probably be tackled as the field matures.

A tightly related question is that of control. Exposing too much control risks overwhelming the designer, but restricting it also risks the design's expressiveness. Typically, novice users require as much automation as possible, while experts would benefit from even direct control at times. Most solutions presented in computer graphics in general, and this survey specifically, are inclined for maximal automation, and hence are more fitting for novice users. On the other hand, CAD methods typically aim for professionals, and hence offer more control, but lack high-level optimizations. An open avenue of research is bridging this gap, allowing different levels of control, suited for both novice and professional users.

In addition, some manufacturing technologies were not addressed at all by the computer graphics community. For example, some of the sturdiest designs are those manufactured with metal, and yet metal sintering is completely disregarded. Composite materials also present phenomenal strength-to-weight ratios. In these cases, brittle yet strong reinforcement fibers are embedded in the object, making it significantly more resistant. Fiber-laying strategies likely stand to benefit from geometry processing techniques developed in the realm of computer graphics, and hence also pose an interesting research avenue. Manufacturing by weaving poses great representational challenges, since enormous numbers of features can be concentrated in very small objects. None of these technologies have been addressed by the graphics community so far, but as their popularity grows, they are likely to be the subjects of research in the near future.

In conclusion, the fields of computational fabrication and fabrication-aware design are still in the midst of an explosive evolution. Many new and exciting applications and approaches are still waiting to be discovered. The various new applications discussed in this survey only scratch the surface of this manufacturing and design potential, as the future will probably tell.

Biographies

Amit H. Bermano is a postdoctoral researcher in the Graphics Group at Princeton University. He obtained his M.Sc at the Technion, Israel and his doctoral degree at ETH Zurich. Before Joining Princeton University, he was a postdoctoral researcher at Disney

Research, Zurich. His research interests include computational fabrication, geometry processing, and facial animation.

Thomas Funkhouser is a Professor in the Department of Computer Science at Princeton University. He obtained his B.S. in biological sciences from Stanford University in 1983, M.S. in computer science from UCLA in 1989, and Ph.D. in computer science from UC Berkeley in 1993. His research interests include interactive geometric modeling, computer graphics, and computer vision.

Szymon Rusinkiewicz is a Professor in the Department of Computer Science at Princeton University. He obtained his B.S. in physics from MIT in 1995, and his Ph.D. in computer science from Stanford in 2001. His research focuses on the interface between computers and the visual and tangible world: acquisition, representation, analysis, and fabrication of 3D shape, motion, surface appearance, and scattering

Acknowledgments

We thank Justin Solomon, Bill Regli, and Jan Vandenbrande for many helpful conversations regarding this work, as well as Sema Berkiten for helping in summarizing some of the papers. This work was partially funded by DARPA agreement no. HR0011-15-2-0040.

References

[BAU15] Brunton A., Arikan C. A., Urban P.: Pushing the limits of 3d color printing: Error diffusion with translucent materials. *CoRR abs/1506.02400* (2015). URL: http://arxiv.org/abs/1506.02400. 3, 17, 18, 20

[BBAM12] Bermano A., Baran I., Alexa M., Matusk W.: Shadowpix: Multiple images from self shadowing. *Comput. Graph. Forum 31*, 2pt3 (May 2012), 593–602. URL: http://dx.doi.org/10.1111/j.1467-8659.2012.03038.x, doi:10.1111/j.1467-8659.2012.03038.x. 3

[BBJP12] Bächer M., Bickel B., James D. L., Pfister H.: Fabricating articulated characters from skinned meshes. *ACM Trans. Graph. 31*, 4 (July 2012), 47:1–47:9. URL: http://doi.acm.org/10.1145/2185520.2185543, doi:10.1145/2185520.2185543. 8, 18, 20

[BBO*10] Bickel B., Bächer M., Otaduy M. A., Lee H. R., Pfister H., Gross M., Matusik W.: Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph. 29*, 4 (July 2010), 63:1–63:10. URL: http://doi.acm.org/10.1145/1778765.1778800, doi:10.1145/1778765.1778800. 6, 7, 17

[BCT15a] Bächer M., Coros S., Thomaszewski B.: Linkedit: Interactive linkage editing using symbolic kinematics. *ACM Trans. Graph. 34*, 4 (July 2015), 99:1–99:8. URL: http://doi.acm.org/10.1145/2766985, doi:10.1145/2766985. 8

[BCT*15b] Bharaj G., Coros S., Thomaszewski B., Tompkin J., Bickel B., Pfister H.: Computational design of walking automata. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (New York, NY, USA, 2015), SCA '15, ACM, pp. 93–100. URL: http://doi.acm.org/10.1145/2786784.2786803, doi:10.1145/2786784.2786803. 8, 20

[BKB*12] Baran I., Keller P., Bradley D., Coros S., Jarosz W., Nowrouzezahrai D., Gross M.: Manufacturing layered attenuators for multiple prescribed shadow images. *Computer Graphics Forum (Proceedings of Eurographics 2012) 31*, 2 (May 2012), 603–610. 3

[BKS*12] Bickel B., Kaufmann P., Skouras M., Thomaszewski B., Bradley D., Beeler T., Jackson P., Marschner S., Matusik W., Gross M.: Physical face cloning. *ACM Trans. Graph. 31*, 4 (July 2012), 118:1–118:10. URL: http://doi.acm.org/10.1145/2185520.2185614, doi:10.1145/2185520.2185614. 6, 17, 19, 20

[Blo09] Block P.: *Thrust network analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 16

[BO07] Block P., Ochsendorf J.: Thrust network analysis: A new methodology for three-dimensional equilibrium. *Journal of the International Association for Shell and Spatial Structures 48*, 3 (December 2007), 167–173. 16

[BWBSH14] Bächer M., Whiting E., Bickel B., Sorkine-Hornung O.: Spin-it: Optimizing moment of inertia for spinnable objects. *ACM Trans. Graph. 33*, 4 (July 2014), 96:1–96:10. URL: http://doi.acm.org/10.1145/2601097.2601157, doi:10.1145/2601097.2601157. 9, 17

[Cas12] Cashman T. J.: Beyond catmull-clark? a survey of advances in subdivision surface methods. *Computer Graphics Forum 31*, 1 (2012), 42–61. URL: http://dx.doi.org/10.1111/j.1467-8659.2011.02083.x, doi:10.1111/j.1467-8659.2011.02083.x. 19

[CBNJ*15] Christiansen A. N., Bærentzen J. A., Nobel-Jørgensen M., Aage N., Sigmund O.: Combined shape and topology optimization of 3d structures. *Computers & Graphics 46* (2015), 25 – 35. Shape Modeling International 2014. URL: http://www.sciencedirect.com/science/article/pii/S0097849314001095, doi:http://dx.doi.org/10.1016/j.cag.2014.09.021. 6, 20

[CCA*12] Calì J., Calian D. A., Amati C., Kleinberger R., Steed A., Kautz J., Weyrich T.: 3d-printing of non-assembly, articulated models. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 130:1–130:8. URL: http://doi.acm.org/10.1145/2366145.2366149, doi:10.1145/2366145.2366149. 8, 20

[CFG*15] Chang A. X., Funkhouser T., Guibas L., Hanrahan P., Huang Q., Li Z., Savarese S., Savva M., Song S., Su H., Xiao J., Yi L., Yu F.: *ShapeNet: An Information-Rich 3D Model Repository*. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 11

[CHB09] Cottrell J. A., Hughes T. J., Bazilevs Y.: *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009. 13

[CJR95] Chalasani K., Jones L., Roscoe L.: Support generation for fused deposition modeling. In *Proceedings of Solid Freeform Fabrication Symposium, University of Texas, Austin, TX, Aug* (1995), pp. 229–241. 16

[CLD*13] Chen D., Levin D. I. W., Didyk P., Sitthi-Amorn P., Matusik W.: Spec2Fab: A reducer-tuner model for translating specifications to 3D prints. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2013, Anaheim, CA) 32*, 4 (2013), 135:1–135:10. 2, 3, 20

[CLM*13] Ceylan D., Li W., Mitra N. J., Agrawala M., Pauly M.: Designing and fabricating mechanical automata from mocap sequences. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 186:1–186:11. URL: http://doi.acm.org/10.1145/2508363.2508400, doi:10.1145/2508363.2508400. 8, 20

[CMWA14] Carter L. N., Martin C., Withers P. J., Attallah M. M.: The influence of the laser scan strategy on grain structure and cracking behaviour in {SLM} powder-bed fabricated nickel superalloy. *Journal of Alloys and Compounds 615* (2014), 338 – 347. URL: http://www.sciencedirect.com/science/article/pii/S092583881401528X, doi:http://dx.doi.org/10.1016/j.jallcom.2014.06.172. 14

[CPMS14] Cignoni P., Pietroni N., Malomo L., Scopigno R.: Field-aligned mesh joinery. *ACM Trans. Graph. 33*, 1 (Feb. 2014), 11:1–11:12. URL: http://doi.acm.org/10.1145/2537852, doi:10.1145/2537852. 12, 20

[CS16] Chen L., Sass L.: Fresh press modeler: A generative system for physically based low fidelity prototyping. *Computers & Graphics 54* (2016), 157 – 165. Special Issue on CAD/Graphics 2015. URL: http://www.sciencedirect.com/science/article/pii/S0097849315001065, doi:http://dx.doi.org/10.1016/j.cag.2015.07.003. 13, 19

[CSaLM13] Chen D., Sitthi-amorn P., Lan J. T., Matusik W.: Computing and fabricating multiplanar models. In *Computer graphics forum* (2013), vol. 32, Wiley Online Library, pp. 305–315. 13

[CTN*13] Coros S., Thomaszewski B., Noris G., Sueda S., Forberg M., Sumner R. W., Matusik W., Bickel B.: Computational design of mechanical characters. *ACM Trans. Graph. 32*, 4 (July 2013), 83:1–83:12. URL: http://doi.acm.org/10.1145/2461912.2461953, doi:10.1145/2461912.2461953. 8, 20

[CWIM*03] Chan I., Wells III W., Mulkern R. V., Haker S., Zhang J., Zou K. H., Maier S. E., Tempany C. M.: Detection of prostate cancer by integration of line-scan diffusion, t2-mapping and t2-weighted magnetic resonance imaging; a multichannel statistical classifier. *Medical physics 30*, 9 (2003), 2390–2398. 17

[CZL*15] Chen X., Zhang H., Lin J., Hu R., Lu L., Huang Q., Benes B., Cohen-Or D., Chen B.: Dapper: Decompose-and-pack for 3d printing. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 213:1–213:12. URL: http://doi.acm.org/10.1145/2816795.2818087, doi:10.1145/2816795.2818087. 15

[CZX*16] Chen W., Zhang X., Xin S., Xia Y., Lefebvre S., Wang W.: Synthesis of filigrees for digital fabrication. *ACM Trans. Graph. 35*, 4 (July 2016), 98:1–98:13. URL: http://doi.acm.org/10.1145/2897824.2925911, doi:10.1145/2897824.2925911. 4, 5, 18, 19

[CZXZ14] Chen X., Zheng C., Xu W., Zhou K.: An asymptotic numerical method for inverse elastic shape design. *ACM Trans. Graph. 33*, 4 (July 2014), 95:1–95:11. URL: http://doi.acm.org/10.1145/2601097.2601189, doi:10.1145/2601097.2601189. 7, 17, 20

[DG14] Deaton J. D., Grandhi R. V.: A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization 49*, 1 (2014), 1–38. URL: http://dx.doi.org/10.1007/s00158-013-0956-z, doi:10.1007/s00158-013-0956-z. 2, 5

[dGAOD13] de Goes F., Alliez P., Owhadi H., Desbrun M.: On the equilibrium of simplicial masonry structures. *ACM Trans. Graph. 32*, 4 (July 2013), 93:1–93:10. URL: http://doi.acm.org/10.1145/2461912.2461932, doi:10.1145/2461912.2461932. 16

[dGDMD16] de Goes F., Desbrun M., Meyer M., DeRose T.: Subdivision exterior calculus for geometry processing. *ACM Trans. Graph. 35*, 4 (July 2016), 133:1–133:11. URL: http://doi.acm.org/10.1145/2897824.2925880, doi:10.1145/2897824.2925880. 19

[DHL14] Dumas J., Hergel J., Lefebvre S.: Bridging the gap: Automated steady scaffoldings for 3d printing. *ACM Trans. Graph. 33*, 4 (July 2014), 98:1–98:10. URL: http://doi.acm.org/10.1145/2601097.2601153, doi:10.1145/2601097.2601153. 16, 18

[DLL*15] Dumas J., Lu A., Lefebvre S., Wu J., München T. U., Dick C., München T. U.: By-example synthesis of structurally sound patterns. *ACM Trans. Graph. 34*, 4 (July 2015), 137:1–137:12. URL: http://doi.acm.org/10.1145/2766984, doi:10.1145/2766984. 4, 17, 20

[DPW*14] Deuss M., Panozzo D., Whiting E., Liu Y., Block P., Sorkine-Hornung O., Pauly M.: Assembling self-supporting structures. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 214:1–214:10. URL: http://doi.acm.org/10.1145/2661229.2661266, doi:10.1145/2661229.2661266. 16, 19

[EKS*10] Eigensatz M., Kilian M., Schiftner A., Mitra N. J., Pottmann H., Pauly M.: Paneling architectural freeform surfaces. *ACM Transactions on Graphics 29*, 3 (2010), to appear. 13, 19

[FLHCO10] Fu C.-W., Lai C.-F., He Y., Cohen-Or D.: K-set tilable surfaces. *ACM Transactions on Graphics (SIGGRAPH 2010 issue) 29*, 3 (August 2010), 44:1–44:6. 13, 19

[FSY*15] Fu C.-W., Song P., Yan X., Yang L. W., Jayaraman P. K., Cohen-Or D.: Computational interlocking furniture assembly. *ACM Trans. Graph. 34*, 4 (July 2015), 91:1–91:11. URL: http://doi.acm.org/10.1145/2766892, doi:10.1145/2766892. 11, 20

[GCMT14] Gauge D., Coros S., Mani S., Thomaszewski B.: Interactive design of modular tensegrity characters. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2014), SCA '14, Eurographics Association, pp. 131–138. URL: http://dl.acm.org/citation.cfm?id=2849517.2849539. 8, 20

[GRS14] Gibson I., Rosen D., Stucker B.: *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. SpringerLink : Bücher. Springer New York, 2014. URL: https://books.google.com/books?id=OPGbBQAAQBAJ. 14

[GSFD*14] Garg A., Sageman-Furnas A. O., Deng B., Yue Y., Grinspun E., Pauly M., Wardetzky M.: Wire mesh design. *ACM Trans. Graph. 33*, 4 (July 2014), 66:1–66:12. URL: http://doi.acm.org/10.1145/2601097.2601106, doi:10.1145/2601097.2601106. 12, 13, 19, 21

[GZR*15] Gao W., Zhang Y., Ramanujan D., Ramani K., Chen Y., Williams C. B., Wang C. C., Shin Y. C., Zhang S., Zavattieri P. D.: The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design 69* (2015), 65 – 89. URL: http://www.sciencedirect.com/science/article/pii/S0010448515000469, doi:http://dx.doi.org/10.1016/j.cad.2015.04.001. 1

[HBA12] Hildebrand K., Bickel B., Alexa M.: crdbrd: Shape fabrication by sliding planar slices. *Computer Graphics Forum 31*, 2pt3 (2012), 583–592. URL: http://dx.doi.org/10.1111/j.1467-8659.2012.03037.x, doi:10.1111/j.1467-8659.2012.03037.x. 12, 17, 20

[HBA13] Hildebrand K., Bickel B., Alexa M.: Orthogonal slicing for additive manufacturing. *Computers & Graphics 37*, 6 (2013), 669 – 675. Shape Modeling International (SMI) Conference 2013. URL: http://www.sciencedirect.com/science/article/pii/S009784931300085X, doi:http://dx.doi.org/10.1016/j.cag.2013.05.011. 15, 17

[HHY*13] Hussein A., Hao L., Yan C., Everson R., Young P.: Advanced lattice support structures for metal additive manufacturing. *Journal of Materials Processing Technology 213*, 7 (2013), 1019–1026. 16

[HIH*13] Hullin M. B., Ihrke I., Heidrich W., Weyrich T., Damberg G., Fuchs M.: Computational fabrication and display of material appearance. *Eurographics State-of-the-Art Reports (STAR)* (May 2013), 17 pages. 2, 3, 4

[HL14] Hergel J., Lefebvre S.: Clean color: Improving multi-filament 3d prints. *Comput. Graph. Forum 33*, 2 (May 2014), 469–478. URL: http://dx.doi.org/10.1111/cgf.12318, doi:10.1111/cgf.12318. 3, 17, 18

[HLZCO14] Hu R., Li H., Zhang H., Cohen-Or D.: Approximate pyramidal shape decomposition. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 213:1–213:12. URL: http://doi.acm.org/10.1145/2661229.2661244, doi:10.1145/2661229.2661244. 15

[HYML09] Huang X., Ye C., Mo J., Liu H.: Slice data based support generation algorithm for fused deposition modeling. *Tsinghua Science & Technology 14* (2009), 223–228. 16

[JBAG12] JHABVALA J., BOILLAT E., ANDRÉ C., GLARDON R.: An innovative method to build support structures with a pulsed laser in the selective laser melting process. *The International Journal of Advanced Manufacturing Technology 59*, 1-4 (2012), 137–142. 16

[JDKL14] JACOBSON A., DENG Z., KAVAN L., LEWIS J.: Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses* (2014). 8

[JHF15] JIN Y.-A., HE Y., FU J.-Z.: Support generation for additive manufacturing based on sliced data. *The International Journal of Advanced Manufacturing Technology 80*, 9-12 (2015), 2041–2052. 16

[Kal15] KALIDINDI S. R.: Data science and cyberinfrastructure: critical enablers for accelerated development of hierarchical materials. *International Materials Reviews 60*, 3 (2015), 150–168. 6

[KCD*16] KONAKOVIĆ M., CRANE K., DENG B., BOUAZIZ S., PIKER D., PAULY M.: Beyond developable: Computational design and fabrication with auxetic materials. *ACM Trans. Graph. 35*, 4 (July 2016), 89:1–89:11. URL: http://doi.acm.org/10.1145/2897824.2925944, doi:10.1145/2897824.2925944. 12, 19, 21

[KLY*14] KOO B., LI W., YAO J., AGRAWALA M., MITRA N. J.: Creating works-like prototypes of mechanical objects. *ACM Transactions on Graphics (Special issue of SIGGRAPH Asia 2014)* (2014). 10, 17, 20

[KMOD09] KHAREVYCH L., MULLEN P., OWHADI H., DESBRUN M.: Numerical coarsening of inhomogeneous elastic materials. *ACM Trans. Graph. 28*, 3 (July 2009), 51:1–51:8. URL: http://doi.acm.org/10.1145/1531326.1531357, doi:10.1145/1531326.1531357. 6

[KSS*15] KOYAMA Y., SUEDA S., STEINHARDT E., IGARASHI T., SHAMIR A., MATUSIK W.: Autoconnect: Computational design of 3d-printable connectors. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 231:1–231:11. URL: http://doi.acm.org/10.1145/2816795.2818060, doi:10.1145/2816795.2818060. 10, 11, 20

[LBRM12] LUO L., BARAN I., RUSINKIEWICZ S., MATUSIK W.: Chopper: Partitioning models into 3D-printable parts. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (Dec. 2012). 15, 17, 19, 20

[LFL09] LO K.-Y., FU C.-W., LI H.: 3d polyomino puzzle. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 157:1–157:8. URL: http://doi.acm.org/10.1145/1618452.1618503, doi:10.1145/1618452.1618503. 11, 18, 19, 20

[LHAZ15] LI H., HU R., ALHASHIM I., ZHANG H.: Foldabilizing furniture. *ACM Transactions on Graphics, (Proc. of SIGGRAPH 2015) 34*, 4 (2015). 10

[LJGH11] LI X.-Y., JU T., GU Y., HU S.-M.: A geometric study of v-style pop-ups: Theories and algorithms. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 98:1–98:10. URL: http://doi.acm.org/10.1145/1964921.1964993, doi:10.1145/1964921.1964993. 12

[LLMZ16] LI D., LEVIN D. I., MATUSIK W., ZHENG C.: Acoustic voxels: Computational optimization of modular acoustic filters. *ACM Trans. Graph. 35*, 4 (2016). doi:10.1145/2897824.2925960. 10, 17, 20

[LM16] LIU J., MA Y.: A survey of manufacturing oriented topology optimization methods. *Advances in Engineering Software 100* (2016), 161 – 175. URL: //www.sciencedirect.com/science/article/pii/S0965997816302174, doi:http://dx.doi.org/10.1016/j.advengsoft.2016.07.017. 5

[LOMI11] LAU M., OHGAWARA A., MITANI J., IGARASHI T.: Converting 3d furniture models to fabricatable parts and connectors. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 85:1–85:6. URL: http://doi.acm.org/10.1145/1964921.1964980, doi:10.1145/1964921.1964980. 10, 17, 20

[LPW*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W.: Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. 25*, 3 (July 2006), 681–689. URL: http://doi.acm.org/10.1145/1141911.1141941, doi:10.1145/1141911.1141941. 13, 19

[LSH*10] LI X.-Y., SHEN C.-H., HUANG S.-S., JU T., HU S.-M.: Popup: automatic paper architectures from 3d models. *ACM Transactions on Graphics 29*, 4 (2010), 111:1–9. 12

[LSZ*14] LU L., SHARF A., ZHAO H., WEI Y., FAN Q., CHEN X., SAVOYE Y., TU C., COHEN-OR D., CHEN B.: Build-to-last: Strength to weight 3d printed objects. *ACM Trans. Graph. 33*, 4 (July 2014), 97:1–97:10. URL: http://doi.acm.org/10.1145/2601097.2601168, doi:10.1145/2601097.2601168. 5, 17, 19

[MAB*15] MUSIALSKI P., AUZINGER T., BIRSAK M., WIMMER M., KOBBELT L.: Reduced-order shape optimization using offset surfaces. *ACM Trans. Graph. 34*, 4 (July 2015), 102:1–102:9. URL: http://doi.acm.org/10.1145/2766955, doi:10.1145/2766955. 10, 18, 19

[MDL16] MARTÍNEZ J., DUMAS J., LEFEBVRE S.: Procedural voronoi foams for additive manufacturing. *ACM Trans. Graph. 35*, 4 (July 2016), 44:1–44:12. URL: http://doi.acm.org/10.1145/2897824.2925922, doi:10.1145/2897824.2925922. 7, 17, 20

[MDLW15] MARTÍNEZ J., DUMAS J., LEFEBVRE S., WEI L.-Y.: Structure and appearance optimization for controllable shape design. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 229:1–229:11. URL: http://doi.acm.org/10.1145/2816795.2818101, doi:10.1145/2816795.2818101. 4

[MGE07] MASSARWI F., GOTSMAN C., ELBER G.: Papercraft models using generalized cylinders. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on* (2007), IEEE, pp. 148–157. 12

[MI07] MORI Y., IGARASHI T.: Plushie: An interactive design system for plush toys. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. URL: http://doi.acm.org/10.1145/1275808.1276433, doi:10.1145/1275808.1276433. 12, 19, 20, 21

[MIB15] MIKI M., IGARASHI T., BLOCK P.: Parametric self-supporting surfaces via direct computation of airy stress functions. *ACM Trans. Graph. 34*, 4 (July 2015), 89:1–89:12. URL: http://doi.acm.org/10.1145/2766888, doi:10.1145/2766888. 16, 19

[MIG*14] MUELLER S., IM S., GUREVICH S., TEIBRICH A., PFISTERER L., GUIMBRETIÈRE F., BAUDISCH P.: Wireprint: 3d printed previews for fast prototyping. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2014), UIST '14, ACM, pp. 273–280. URL: http://doi.acm.org/10.1145/2642918.2647359, doi:10.1145/2642918.2647359. 14, 18

[MLB16] MIGUEL E., LEPOUTRE M., BICKEL B.: Computational design of stable planar-rod structures. *ACM Trans. Graph. 35*, 4 (July 2016), 86:1–86:11. URL: http://doi.acm.org/10.1145/2897824.2925978, doi:10.1145/2897824.2925978. 11, 12

[MP09] MITRA N. J., PAULY M.: Shadow art. *ACM Transactions on Graphics 28*, 5 (2009). to appear. 3

[MS04] MITANI J., SUZUKI H.: Making papercraft toys from meshes using strip-based approximate unfolding. In *ACM transactions on graphics (TOG)* (2004), vol. 23, ACM, pp. 259–263. 12

[MTN*15] MEGARO V., THOMASZEWSKI B., NITTI M., HILLIGES O., GROSS M., COROS S.: Interactive design of 3d-printable robotic creatures. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 216:1–216:9. URL: http://doi.acm.org/10.1145/2816795.2818137, doi:10.1145/2816795.2818137. 9, 20

[MUB15] MARTIN T., UMETANI N., BICKEL B.: Omniad: Data-driven omni-directional aerodynamics. *ACM Trans. Graph. 34*, 4 (July 2015), 113:1–113:12. URL: http://doi.acm.org/10.1145/2766919, doi:10.1145/2766919. 10, 19, 20

[MUS14] MCCRAE J., UMETANI N., SINGH K.: Flatfitfab: Interactive modeling with planar sections. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2014), UIST '14, ACM, pp. 13–22. URL: http://doi.acm.org/10.1145/2642918.2647388, doi:10.1145/2642918.2647388. 12, 13

[NMK*06] Nealen A., Müller M., Keiser R., Boxerman E., Carlson M.: Physically based deformable models in computer graphics. In *Computer graphics forum* (2006), vol. 25, Wiley Online Library, pp. 809–836. 5

[PBSH13] Panozzo D., Block P., Sorkine-Hornung O.: Designing unreinforced masonry models. *ACM Trans. Graph. 32*, 4 (July 2013), 91:1–91:12. URL: http://doi.acm.org/10.1145/2461912.2461958, doi:10.1145/2461912.2461958. 16, 17, 19, 20

[PDP*15] Panozzo D., Diamanti O., Paris S., Tarini M., Sorkine E., Sorkine-Hornung O.: Texture mapping real-world objects with hydrographics. *Computer Graphics Forum (proceedings of EUROGRAPHICS Symposium on Geometry Processing) 34*, 5 (2015), 65–75. 3

[PHN*12] Papas M., Houit T., Nowrouzezahrai D., Gross M., Jarosz W.: The magic lens: Refractive steganography. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 31*, 6 (Nov. 2012). doi:10.1145/2366145.2366205. 3

[PJJ*11] Papas M., Jarosz W., Jakob W., Rusinkiewicz S., Matusik W., Weyrich T.: Goal-based caustics. *Computer Graphics Forum (Proc. Eurographics) 30*, 2 (Apr. 2011). 3

[Pot13] Pottmann H.: Architectural geometry and fabrication-aware design. *Nexus Network Journal 15*, 2 (2013), 195–208. URL: http://dx.doi.org/10.1007/s00004-013-0149-5, doi:10.1007/s00004-013-0149-5. 13

[PRJ*13] Papas M., Regg C., Jarosz W., Bickel B., Jackson P., Matusik W., Marschner S., Gross M.: Fabricating translucent materials using continuous pigment mixtures. *ACM Trans. Graph. 32*, 4 (July 2013), 146:1–146:12. URL: http://doi.acm.org/10.1145/2461912.2461974, doi:10.1145/2461912.2461974. 3, 20

[PRM14] Pereira T., Rusinkiewicz S., Matusik W.: Computational light routing: 3D printed fiber optics for sensing and display. *ACM Transactions on Graphics 33*, 3 (May 2014). 4, 17, 19, 20

[Pro05] Provatidis C. G.: Analysis of box-like structures using 3-d coons' interpolation. *Communications in Numerical Methods in Engineering 21*, 8 (2005), 443–456. URL: http://dx.doi.org/10.1002/cnm.762, doi:10.1002/cnm.762. 19

[PTC*15] Pérez J., Thomaszewski B., Coros S., Bickel B., Canabal J. A., Sumner R., Otaduy M. A.: Design and fabrication of flexible rod meshes. *ACM Trans. Graph. 34*, 4 (July 2015), 138:1–138:12. URL: http://doi.acm.org/10.1145/2766998, doi:10.1145/2766998. 7, 18, 19, 20

[PTP*14] Pietroni N., Tonelli D., Puppo E., Froli M., Scopigno R., Cignoni P.: Voronoi grid-shell structures. *CoRR abs/1408.6591* (2014). URL: http://arxiv.org/abs/1408.6591. 13, 17, 19, 20

[PWLSH13] Prévost R., Whiting E., Lefebvre S., Sorkine-Hornung O.: Make It Stand: Balancing shapes for 3D fabrication. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH) 32*, 4 (2013), 81:1–81:10. 9, 17

[PZM*15] Panetta J., Zhou Q., Malomo L., Pietroni N., Cignoni P., Zorin D.: Elastic textures for additive fabrication. *ACM Trans. on Graphics - Siggraph 2015 34*, 4 (aug 2015), 12. Julian Panetta and Quingnan Zhou are Joint first authors. URL: http://vcg.isti.cnr.it/Publications/2015/PZMPCZ15. 6, 17, 20

[RCM*14] Reiner T., Carr N., Měch R., Šťava O., Dachsbacher C., Miller G.: Dual-color mixing for fused deposition modeling printers. *Computer Graphics Forum 33*, 2 (2014), 479–486. URL: http://dx.doi.org/10.1111/cgf.12319, doi:10.1111/cgf.12319. 3, 18, 20

[SARW*15] Sitthi-Amorn P., Ramos J. E., Wangy Y., Kwan J., Lan J., Wang W., Matusik W.: Multifab: A machine vision assisted platform for multi-material 3d printing. *ACM Trans. Graph. 34*, 4 (July 2015), 129:1–129:11. URL: http://doi.acm.org/10.1145/2766962, doi:10.1145/2766962. 14, 17, 19

[SBR*15] Schumacher C., Bickel B., Rys J., Marschner S., Daraio C., Gross M.: Microstructures to control elasticity in 3d printing. *ACM Trans. Graph. 34*, 4 (July 2015), 136:1–136:13. URL: http://doi.acm.org/10.1145/2766926, doi:10.1145/2766926. 7, 17, 20

[SCP*14] Scopigno R., Cignoni P., Pietroni N., Callieri M., Dellepiane M.: Digital Fabrication Technologies for Cultural Heritage (STAR). In *Eurographics Workshop on Graphics and Cultural Heritage* (2014), Klein R., Santos P., (Eds.), The Eurographics Association. doi:10.2312/gch.20141306. 2

[SDW*16] Song P., Deng B., Wang Z., Dong Z., Li W., Fu C.-W., Liu L.: CofiFab: Coarse-to-fine fabrication of large 3d objects. *ACM Transactions on Graphics (SIGGRAPH 2016) 35*, 4 (2016). Article 45. 15

[Séq12] Séquin C. H.: Prototyping dissection puzzles with layered manufacturing. In *Fabrication and Sculpture Track, Shape Modeling International* (2012). 11

[SFCO12] Song P., Fu C.-W., Cohen-Or D.: Recursive interlocking puzzles. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 128:1–128:10. URL: http://doi.acm.org/10.1145/2366145.2366147, doi:10.1145/2366145.2366147. 11

[SP13] Schwartzburg Y., Pauly M.: Fabrication-aware design with intersecting planar pieces. *Computer Graphics Forum (Proceedings of Eurographics 2013) 32*, 2 (2013), 317–326. 12

[SPG*16] Schüller C., Panozzo D., Grundhöfer A., Zimmer H., Sorkine E., Sorkine-Hornung O.: Computational thermoforming. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 35*, 4 (2016). 3

[SS10] Singh M., Schaefer S.: Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph. 29*, 4 (July 2010), 46:1–46:7. URL: http://doi.acm.org/10.1145/1778765.1778783, doi:10.1145/1778765.1778783. 13, 19

[SSL*14] Schulz A., Shamir A., Levin D. I. W., Sitthi-Amorn P., Matusik W.: Design and fabrication by example. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2014) 33*, 4 (2014). 10, 20

[SSM15] Shugrina M., Shamir A., Matusik W.: Fab forms: Customizable objects for fabrication with validity and geometry caching. *ACM Trans. Graph. 34*, 4 (2015). 9, 20

[STBG12] Skouras M., Thomaszewski B., Bickel B., Gross M.: Computational design of rubber balloons. *Comput. Graphics Forum (Proc. Eurographics)* (2012). 7, 19, 20

[STC*13] Skouras M., Thomaszewski B., Coros S., Bickel B., Gross M.: Computational design of actuated deformable characters. *ACM Trans. Graph. 32*, 4 (July 2013), 82:1–82:10. URL: http://doi.acm.org/10.1145/2461912.2461979, doi:10.1145/2461912.2461979. 7

[STK*14] Skouras M., Thomaszewski B., Kaufmann P., Garg A., Bickel B., Grinspun E., Gross M.: Designing inflatable structures. *ACM Trans. Graph. 33*, 4 (July 2014), 63:1–63:10. URL: http://doi.acm.org/10.1145/2601097.2601166, doi:10.1145/2601097.2601166. 12, 19, 20, 21

[STTP14] Schwartzburg Y., Testuz R., Tagliasacchi A., Pauly M.: High-contrast computational caustic design. *ACM Trans. Graph. 33*, 4 (July 2014), 74:1–74:11. URL: http://doi.acm.org/10.1145/2601097.2601200, doi:10.1145/2601097.2601200. 3

[Stu98] Sturman D.: The state of computer animation. *SIGGRAPH Comput. Graph. 32*, 1 (Feb. 1998), 57–61. URL: http://doi.acm.org/10.1145/279389.279467, doi:10.1145/279389.279467. 8

[SVB*12] Stava O., Vanek J., Benes B., Carr N., Měch R.: Stress relief: Improving structural strength of 3d printable objects. *ACM Trans. Graph. 31*, 4 (July 2012), 48:1–48:11. URL: http://doi.acm.org/10.1145/2185520.2185544, doi:10.1145/2185520.2185544. 5, 17, 18, 19

[SZ15] Sun T., Zheng C.: Computational design of twisty joints and puzzles. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2015) 34*, 4 (Aug. 2015). URL: http://www.cs.columbia.edu/cg/twisty. 11

[SZBN03] Sederberg T. W., Zheng J., Bakenov A., Nasri A.: T-splines and t-nurccs. *ACM Trans. Graph. 22*, 3 (July 2003), 477–484. URL: http://doi.acm.org/10.1145/882262.882295, doi:10.1145/882262.882295. 19

[TCG*14] Thomaszewski B., Coros S., Gauge D., Megaro V., Grinspun E., Gross M.: Computational design of linkage-based characters. *ACM Trans. Graph. 33*, 4 (July 2014), 64:1–64:9. URL: http://doi.acm.org/10.1145/2601097.2601143, doi:10.1145/2601097.2601143. 8, 20

[TDS*16] Tagliasacchi A., Delame T., Spagnuolo M., Amenta N., Telea A.: 3d skeletons: A state-of-the-art report. *Computer Graphics Forum 35*, 2 (2016), 573–597. URL: http://dx.doi.org/10.1111/cgf.12865, doi:10.1111/cgf.12865. 17

[THKM13] Tompkin J., Heinzle S., Kautz J., Matusik W.: Content-adaptive lenticular prints. *ACM Trans. Graph. 32*, 4 (July 2013), 133:1–133:10. URL: http://doi.acm.org/10.1145/2461912.2462011, doi:10.1145/2461912.2462011. 3

[TSG*14] Tang C., Sun X., Gomes A., Wallner J., Pottmann H.: Form-finding with polyhedral meshes made simple. *ACM Trans. Graph. 33*, 4 (2014), 70–1. 16

[TVK*12] Tiwari P., Viswanath S., Kurhanewicz J., Sridhar A., Madabhushi A.: Multimodal wavelet embedding representation for data combination (maweric): integrating magnetic resonance imaging and spectroscopy for prostate cancer detection. *NMR in Biomedicine 25*, 4 (2012), 607–619. 17

[UIM12] Umetani N., Igarashi T., Mitra N. J.: Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph. 31*, 4 (July 2012), 86:1–86:11. URL: http://doi.acm.org/10.1145/2185520.2185582, doi:10.1145/2185520.2185582. 10, 17, 20

[UKSI14] Umetani N., Koyama Y., Schmidt R., Igarashi T.: Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph. 33*, 4 (July 2014), 65:1–65:10. URL: http://doi.acm.org/10.1145/2601097.2601129, doi:10.1145/2601097.2601129. 10, 19

[US13] Umetani N., Schmidt R.: Cross-sectional structural analysis for 3d printing optimization. In *SIGGRAPH Asia 2013 Technical Briefs* (New York, NY, USA, 2013), SA '13, ACM, pp. 5:1–5:4. URL: http://doi.acm.org/10.1145/2542355.2542361, doi:10.1145/2542355.2542361. 5, 20

[VGB14a] Vanek J., Galicia J. A. G., Benes B.: Clever support: Efficient support structure generation for digital fabrication. *Computer Graphics Forum 33*, 5 (2014), 117–125. URL: http://dx.doi.org/10.1111/cgf.12437, doi:10.1111/cgf.12437. 16

[VGB*14b] Vanek J., Galicia J. A. G., Benes B., Mäžch R., Carr N., Stava O., Miller G. S.: Packmerger: A 3d print volume optimizer. *Computer Graphics Forum 33*, 6 (2014), 322–332. URL: http://dx.doi.org/10.1111/cgf.12353, doi:10.1111/cgf.12353. 15, 17, 19, 20

[VHWP12] Vouga E., Höbinger M., Wallner J., Pottmann H.: Design of self-supporting surfaces. *ACM Trans. Graphics* (2012). Proc. SIGGRAPH. 16

[VWRKM13] Vidimče K., Wang S.-P., Ragan-Kelley J., Matusik W.: Openfab: A programmable pipeline for multi-material fabrication. *ACM Transactions on Graphics 32* (July 2013). 2, 20

[WCT*15] Wang W., Chao H., Tong J., Yang Z., Tong X., Li H., Liu X., Liu L.: Saliency-preserving slicing optimization for effective 3d printing. *Computer Graphics Forum 34*, 6 (2015), 148–160. URL: http://dx.doi.org/10.1111/cgf.12527, doi:10.1111/cgf.12527. 15, 18

[WDW15] Wu J., Dick C., Westermann R.: A system for high-resolution topology optimization. *Visualization and Computer Graphics, IEEE Transactions on PP*, 99 (2015), 1–1. doi:10.1109/TVCG.2015.2502588. 6, 17

[WHL*08] Wang H., He Y., Li X., Gu X., Qin H.: Polycube splines. *Computer-Aided Design 40*, 6 (2008), 721–733. 19

[WPGM16] Wu R., Peng H., Guimbretière F., Marschner S.: Printing arbitrary meshes with a 5dof wireframe printer. *ACM Trans. Graph. 35*, 4 (July 2016), 101:1–101:9. URL: http://doi.acm.org/10.1145/2897824.2925966, doi:10.1145/2897824.2925966. 14

[WWY*13] Wang W., Wang T. Y., Yang Z., Liu L., Tong X., Tong W., Deng J., Chen F., Liu X.: Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 177:1–177:10. URL: http://doi.acm.org/10.1145/2508363.2508382, doi:10.1145/2508363.2508382. 14, 17, 18

[XLCB15] Xu H., Li Y., Chen Y., Barbič J.: Interactive material design using model reduction. *ACM Trans. Graph. 34*, 2 (Mar. 2015), 18:1–18:14. URL: http://doi.acm.org/10.1145/2699648, doi:10.1145/2699648. 6, 17, 19, 20

[XLF*11] Xin S., Lai C.-F., Fu C.-W., Wong T.-T., He Y., Cohen-Or D.: Making burr puzzles from 3d models. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 97:1–97:8. URL: http://doi.acm.org/10.1145/1964921.1964992, doi:10.1145/1964921.1964992. 11, 20

[YCL*15] Yao M., Chen Z., Luo L., Wang R., Wang H.: Level-set-based partitioning and packing optimization of a printable model. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 214:1–214:11. URL: http://doi.acm.org/10.1145/2816795.2818064, doi:10.1145/2816795.2818064. 15, 17, 19, 20

[YIC*12] Yue Y., Iwasaki K., Chen B.-Y., Dobashi Y., Nishita T.: Pixel art with refracted light by rearrangeable sticks. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 575–582. 3

[YZ15] Yang S., Zhao Y. F.: Additive manufacturing-enabled design theory and methodology: a critical review. *The International Journal of Advanced Manufacturing Technology 80*, 1 (2015), 327–342. URL: http://dx.doi.org/10.1007/s00170-015-6994-5, doi:10.1007/s00170-015-6994-5. 1

[ZCT16] Zehnder J., Coros S., Thomaszewski B.: Designing structurally-sound ornamental curve networks. *ACM Trans. Graph. 35*, 4 (July 2016), 99:1–99:10. URL: http://doi.acm.org/10.1145/2897824.2925888, doi:10.1145/2897824.2925888. 4, 18, 19, 20

[ZGH*16] Zhao H., Gu F., Huang Q.-X., Garcia J., Chen Y., Tu C., Benes B., Zhang H., Cohen-Or D., Chen B.: Connected fermat spirals for layered fabrication. *ACM Trans. Graph. 35*, 4 (July 2016), 100:1–100:10. URL: http://doi.acm.org/10.1145/2897824.2925958, doi:10.1145/2897824.2925958. 14, 18, 19

[ZJL14] Zhou S., Jiang C., Lefebvre S.: Topology-constrained synthesis of vector patterns. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 215:1–215:11. URL: http://doi.acm.org/10.1145/2661229.2661238, doi:10.1145/2661229.2661238. 4

[ZLW*16] Zhao H., Lu L., Wei Y., Lischinski D., Sharf A., Cohen-Or D., Chen B.: Printed perforated lampshades for continuous projective images. *ACM Trans. Graph. 35*, 5 (June 2016), 154:1–154:11. URL: http://doi.acm.org/10.1145/2907049, doi:10.1145/2907049. 3

[ZPZ13] ZHOU Q., PANETTA J., ZORIN D.: Worst-case structural analysis. *ACM Trans. Graph. 32*, 4 (July 2013), 137:1–137:12. URL: http://doi.acm.org/10.1145/2461912.2461967, doi:10.1145/2461912.2461967. 5, 17, 19, 20

[ZSMS14] ZHOU Y., SUEDA S., MATUSIK W., SHAMIR A.: Boxelization: Folding 3d objects into boxes. *ACM Trans. Graph. 33*, 4 (July 2014), 71:1–71:8. URL: http://doi.acm.org/10.1145/2601097.2601173, doi:10.1145/2601097.2601173. 10, 17

[ZVKD10] ZHANG H., VAN KAICK O., DYER R.: Spectral mesh processing. *Computer Graphics Forum 29*, 6 (2010), 1865–1894. URL: http://dx.doi.org/10.1111/j.1467-8659.2010.01655.x, doi:10.1111/j.1467-8659.2010.01655.x. 19

[ZXS*12] ZHU L., XU W., SNYDER J., LIU Y., WANG G., GUO B.: Motion-guided mechanical toy modeling. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 127:1–127:10. URL: http://doi.acm.org/10.1145/2366145.2366146, doi:10.1145/2366145.2366146. 8, 20

[ZYZZ15] ZHANG Y., YIN C., ZHENG C., ZHOU K.: Computational hydrographic printing. *ACM Trans. Graph. 34*, 4 (July 2015), 131:1–131:11. URL: http://doi.acm.org/10.1145/2766932, doi:10.1145/2766932. 3