Medley: A Library of Embeddables to Explore Rich Material Properties for 3D Printed Objects

Xiang 'Anthony' Chen
HCI Institute
Carnegie Mellon University
xiangchen@acm.org

Stelian Coros
Computer Science Department
ETH Zurich
scoros@inf.ethz.ch

Scott E. Hudson
HCI Institute
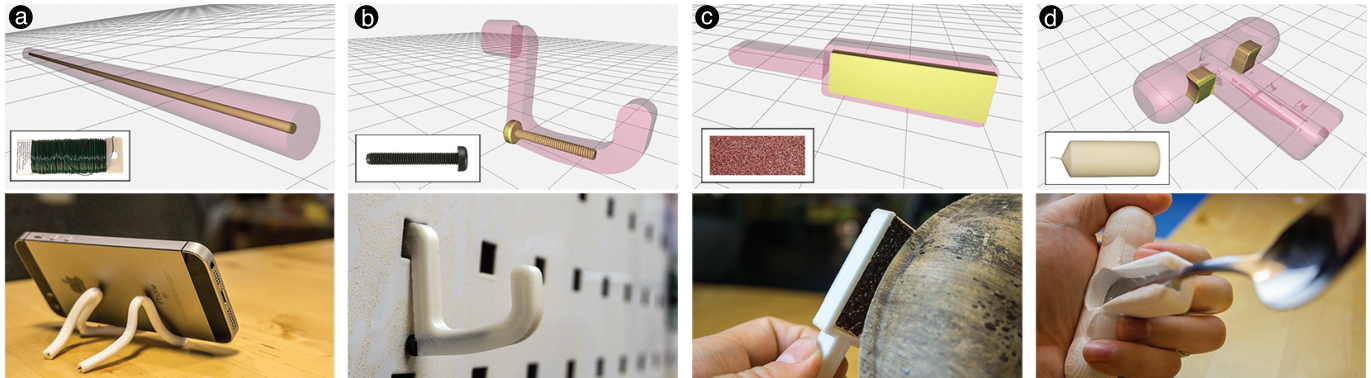Carnegie Mellon University
scott.hudson@cs.cmu.edu

Figure 1. We present a library of *embeddables*—everyday objects that can be cut, worked and embedded into 3D printable designs to augment their material properties. For example, inserting a florists' wire makes elastic TPU-based objects stay in shape (plastic deformation) (a); a screw is used to reinforce a pegboard hook (b); a piece of sand paper becomes the surface of a printed sander (c); the malleability of wax makes it possible to further personalize assistive devices even after they are printed (d).

ABSTRACT

In our everyday life, we interact with and benefit from objects with a wide range of material properties. In contrast, personal fabrication machines (e.g., desktop 3D printers) currently only support a much smaller set of materials. Our goal is to close the gap between current limitations and the future of multi-material printing by enabling people to explore the reuse of material from everyday objects into their custom designs. To achieve this, we develop a library of *embeddables*–everyday objects that can be cut, worked and embedded into 3D print-able designs. We describe a design space that characterizes the geometric and material properties of embeddables. We then develop Medley—a design tool whereby users can import a 3D model, search for embeddables with desired material properties, and interactively edit and integrate their geometry to fit into the original design. Medley also supports the final fabrication and embedding process, including instructions for carving or cutting the objects, and generating optimal paths for inserting embeddables. To validate the expressiveness of our library, we showcase numerous examples augmented by embeddables that go beyond the objects' original printed materials.

ACM Classification Keywords

H.5.2. Information interfaces and presentation: User Interfaces-Input devices and strategies

Author Keywords

Embeddables; multi-material printing; fabrication

INTRODUCTION

In our everyday life, we are surrounded by objects with a wide range of material properties. Consider gloves as an example. They can be made of leather for insulation against cold (Figure 2a), or silicone for protection against heat (Figure 2b), or nitrile for elasticity (Figure 2c), or rubber for better grip and abrasion resistance (Figure 2d), or even infused with copper to provide support for hands and joints (Figure 2e).

In contrast, despite the rapid development in multi-material printing, personal fabrication machines (e.g., desktop 3D printers) currently only support a much smaller set of materials. This limits people's creativity in exploring how their designs might incorporate and benefit from various material properties. It also limits the application of fabricated objects, preventing them from being adopted for a wider context of usage.
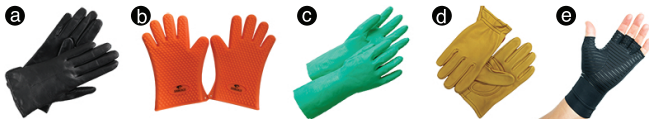
Figure 2. Everyday objects are made of various materials, e.g., for gloves, we have leather (a), silicone (b), nitrile (c), rubber (d) and even infused copper (e).

To overcome the current limitations of material variety, one approach is to design *metamaterials*—a composition of material(s) that exhibits behaviors beyond their natural properties by means of carefully constructed spatial variations in the material(s). Prior work has explored metamaterial designs for elasticity [12, 17], acoustics [11], and mechanisms [5]. To incorporate an even larger set of properties, another approach is to embed alternate materials, such as conductive paint for interactivity [14] or state-changeable material for custom deformability [4]. Though promising, prior work tends to focus on one or a few specific properties, rather than an approach that affords the exploration of a holistic set.

The goal of our research is to leverage a repertoire of material richness in everyday objects to enable the exploration of various material properties in people's custom designs. To achieve this, we develop a library of *embeddables*—everyday objects that can be cut, worked and embedded into 3D printable designs. For example, as shown in Figure 1a, by inserting a florists' wire we can make an elastic object's stay in shape (plastic deformation), such as making a phone holder from a 'noodle' printed in TPU (Thermoplastic Polyurethane); embedding an M3 20mm screw strengthens a pegboard hook along its overhanging structure (Figure 1b); embedding a piece of sand paper adds roughness and creates a simple sander (Figure 1c); embeddables can also be used for assistive devices: embedding wax into an a spoon handle[1] allows users to sand it to a custom shape where the fingers can grip comfortably (Figure 1d).

To enable the use of these embeddables into fabricated designs, we first contribute a systematic categorization of both their geometric and material properties. Specifically, we construct a design space (Figure 3) composed of two dimensions: in terms of geometry, we define *degree of workability* (DOW) to characterize how different embeddables can be worked into user-defined shapes; in terms of material, we introduce an extensible taxonomy of material properties relevant to personal fabrication. Combined, these two dimensions produce a large space of examples making use of embeddables in various usage scenarios. Further, it also serves to unify a number of projects from prior work.

Building upon the design space, we then contribute Medley–a tool for the computational design and fabrication of embeddables into users' custom designs. With Medley, users can import a 3D model, search for embeddables with desired material properties, and interactively edit and integrate their geometry to fit into the original design. Further, Medley also supports the fabrication and embedding process, including instructions

for carving or cutting the objects, and generating optimal paths for inserting them.

Contribution

To validate the expressiveness of our approach, we showcase a series of examples augmented by embeddables that go beyond the objects' original printed materials. Our main contribution is a library with computational support for the exploration of material properties to augment personal fabrication, closing the gap between its current limitations and the future of multi-material printing.

In the remainder of this paper, we first review related work. Next, we present the design space underpinning the construction of our embeddable library as well as the computational design process. We then demonstrate a wide range of printed results, and describe the technical details, which include specifying the geometry of embeddables, finding optimal insertion paths, and generating instructions and templates to guide the embedding process. We close by discussing current limitations and future work.

RELATED WORK

Our work is inspired by three areas of research: multi-material printing methods and machines, techniques to embed elements into 3D printed objects, and tools for designing objects' material properties.

Multi-material Printing Methods and Machines

Multi-material printing has become a recurring theme in the recent development of additive manufacturing, such as Stratasys' PolyJet printers [20], MarkForged's carbon fiber composite printer [10], and Mosaic's Palette+ printer that combines multiple filaments into one single strand [9].

Research in personal fabrication has also demonstrated custom platforms that support printing with multiple materials. For example, MultiFab uses a machine vision system to scan existing objects or printed layers, thus enabling it to apply multiple materials spatially based on the scanned results, such as printing objects with detailed texture, or meta-materials made of multiple materials [18]. xPrint provides an integrated design and fabrication platform for parametrically depositing a wide range of liquid and gel materials to achieve different effects such as shape changing objects [24]. Foldem stacks sheets of multiple materials and then uses a laser cutter to ablate these layers selectively to enable different material properties across the design domain [25].

Despite these recent developments, fully multi-material printing machines remain fairly rare and high-cost compared to standard single- or dual- material desktop printers. Further, even with access to such printers, users still need to explore the design space of materials, and how to incorporate them into a design. To support the first task, prior work has demonstrated a rapid prototyping approach by embedding a range of elements into a single-material design. To support the second task, researchers have come up with computational design tools that provide a vocabulary for users to specify material distribution and layout in their design. Below we review these two bodies of work.

---

[1]Adopted from 'The Andrew Fork': https://www.thingiverse.com/thing:1816581

Embedding Elements into 3D Printed Objects

Various elements of different material properties have been embedded into fabricated objects. Printed Optics embeds elements to customize the optical properties of an object [26]. Acoustruments embeds tunnels in printed device casings to enhance interactivity by sensing acoustic signals from user input [7]. InfraStructs uses multiple materials and AirCode uses air pockets to embed encoded information in a printed object [27, 8]. PipeDream embeds conductive material routed through the interior of an object to enable various kinds of interactivity over the surface [14]. Liquido can capcitively sense tilting and motion of printed objects, which is enabled by a fabrication platform that can embed liquids during the printing process [16]. Rather than static material, HotFlex proposes an approach to dynamically modify the state (shape) of 3D printed objects after they are printed, which is achieved by embedding low melting point material whose stiffness can be controlled by computationally generating heat from embedded resistors [4].

All this work tends to focus on embedding one particular kind of material, or enhancing one type of material property. Our research takes a different angle, as we propose a design space and a material library that encompass a wide range of material properties, which are made possible by embedding materials upcycled from everyday objects. To further support the embedding process, we also provide a computational design tool. Below we review prior art on tool support for multi-material design.

Tools to Design Objects' Material Properties

The emergent multi-material printers also demand computational support for end users to perform material design. Bickel et al. design and fabricate objects with desired deformation behavior by first measuring example deformation, based on which generated layers of microstructures can be used to achieve the same type of behavior [1]. OpenFab provides a programming pipeline that uses shader-like 'fablets' to procedurally define surface detail and material composition throughout the object volume [23]. Foundry is a tool for specifying hierarchical material properties of a 3D object, using various composable operators that decompose/subdivide the object, remap the coordinates, and assign material compositions to each sub-volume [22].

Though promising, these projects almost exclusively focus on computational methods and processes, rather than creating interactive environments for directly manipulating material designs. To address this, Capricate is a design tool whereby a user can select areas on a 3D object to generate capacitive touch materials [15]. To further promote interactivity, we take a more comprehensive approach: our library of embeddables enable users to explore what materials are available for embedding, which is then supported by a tool with a simple and interactive workflow to specify the geometry of the embedded material as well as generating optimal insertion paths and templates for the crafting process. Below we first describe the conceptual backbone of our work–a design space that characterizes material properties of everyday objects and how they can be worked into 3D designs.

DESIGN SPACE

To design and fabricate objects with embeddables, it is important to understand *(i)* material: what properties are available and how to find them in everyday objects, and *(ii)* geometric workability: how to work (e.g., cutting, carving) embeddables into specific geometry that fits into certain parts of the design object, which needs to planned at design time as well as executed in the fabrication process. To structure our understanding of these two questions, we construct a design space as shown in Figure 3.

The rows of the design space correspond to embeddables' different material properties. These properties are collected from a survey of related work on embedding elements into fabricated objects (shown in italics in Figure 3) as well as literature in material science [6]. Our current focus is on a set of material properties that can be found in common everyday objects, and are relevant to personal fabrication (i.e., have been explored in prior work, or can lead to new design directions). Our current list of properties is by no means exhaustive; rather they are meant to be extensible and more properties can be included under this framework.

The columns of the design space describe the embeddables' *degree of workability* (DOW)–how they can be worked into user-defined shape. Notice that DOW is different from the dimension of an embeddable's geometry. For example, as shown below, a wire has DOW 1 (cut to length) yet it can be bent into 2D or 3D geometry.

- *DOW* = 0 are fixed-shaped, atomic objects that will be directly embedded into a design. For example as shown in Figure 1b, a screw is embedded into a peg board hook to increase its flexural strength.

- *DOW* = 1 are wires, tubes or rods of material that can be cut to certain length to create embeddables. Length is the 1 DOW. For example, Figure 1a shows a florists' wire cut to length and inserted into a TPU-based 'noodle' to enable plastic deformation.

- *DOW* = 2 are patches, sheets or plates of material that can be cut to certain area to create embeddables, thus having 2 DOW on the X/Y plane. The other dimension, thickness, is fixed. For example, a piece of sand paper is cut to replace certain area in order to create a printed sander (Figure 1c).

- *DOW* = 3 are bars, blocks or lumps of material that can be cut to arbitrary 3D shape within their size limits. For example, by carving out two pieces of finger-sized wax we can embed them to customize the handle of an assistive utensil holder (Figure 1d).

Each cell in Figure 3 contains one example—either a piece of related work or an embeddable—with the specific material property and DOW. Admittedly each of these is not the only example of a particular cell, or necessarily better than the other examples not shown here. They merely serve to provide a concrete illustration of what embeddables are possible within a given cell, and potentially a starting point for interested readers to explore other similar embeddables. On the other hand, one embeddable might have more than one property

	WORKABILITY	NON-WORKABLE	WORKABLE		
MATERIAL PROPERTIES		DEGREE 0 fixed-shaped, atomic objects	DEGREE 1 wires, tubes, rods, etc.	DEGREE 2 patches, sheets, plates, etc.	DEGREE 3 bars, blocks, lumps, etc.
MECHANICAL	DENSITY	nuts	threaded rod	metal sheet	clay
	FEEL — SOFTNESS	soft button	thread	*Embedding fabric* Rivera at al. [12]	sponge
	FEEL — ROUGHNESS	wood screw	threaded rod	sand paper	scourer
	STRENGTH — TENSILE	nail	fishing line	metal sheet	lumber
	STRENGTH — COMPRESSION	nuts	bike spoke	metal sheet	lumber
	STRENGTH — FLEXURAL	screw	threaded rod	acrylic sheet	lumber
	DEFORMABILITY — ELASTICITY	spring	rubber band	rubber sheet	foam
	DEFORMABILITY — FLEXIBILITY	press-fit rivets	florists' wire	*Hotflex* Groeger et al. [3]	foam
	DEFORMABILITY — PLASTICITY	gorillapod	paper clip	metal sheet	clay
	DURABILITY — WEAR RESISTANCE	screwdriver bits	chicken wire	rubber sheet	shoe sole
	MALLEABILITY — MACHINABILITY	N/A by definition	foam rod	MDF	wax
	CONTACT — FRICTION	ball bearings	steel rod	rubber sheet	sponge
ACOUSTIC		marble	*Accoustruments* Laput et al. [6]	aluminum can	foam earplugs
ELECTRICAL		*Embedded electronics* e.g., Voxel8	*PipeDream* Savage et al. [13]	*Capricate* Schmitz et al. [14]	*Capricate* Schmitz et al. [14]
OPTICAL		marble	*Printed optics* Willis et al. [25]	plastic bottle	matte gel
THERMAL		coin	copper wire	silicon oven mitt	insulation foam

Figure 3. A design space characterizes the material properties of embeddables, as well as their *degree of workability* (DOW)—whether they are embedded as fixed-shape objects (DOW=0), or can be cut to length (1), to a fixed-thickness profile (2), or to a customized 3D form (3).

relevant to fabrication. For example, an embeddable might exhibit a similar set of properties, such as lumber has nice tensile, compressive and flexural strengths; in other cases, one embeddable might present quite different properties, such as kitchen sponge is both soft and has high surface friction due to its porousness.

A LIBRARY OF EMBEDDABLES → RICH MATERIALITY

Based on the design space, we have constructed an extensible library of embeddables. To validate its expressiveness, in this section, we first demonstrate results showing how this library enables the exploration of rich material properties in 3D printed objects. We leave the technical details in the next few sections where we then describe how to computationally design with embeddables from the library.

It is beyond the scope of this paper to include every single cell in Figure 3; rather, our strategy is to create a set of examples that sample broadly across the design space. In so doing, our goal is to validate how this space holistically affords the enabling of rich material properties beyond the small set supported by today's desktop 3D printers.

Below we showcase a series of examples designed using Medley, which validated the richness and expressiveness of our proposed embeddable library.

Deformation As shown earlier in Figure 1a, inserting wires (e.g., florists' wire, a straightened paper clip) can enable elastic material (e.g., TPU-based prints) to perform plastic deformation, i.e., being able to stay in a deformed shape. As another example, we can use this kind of embeddables to create adjustable hinges (Figure 4) for a pair of Harry Potter glasses[2].
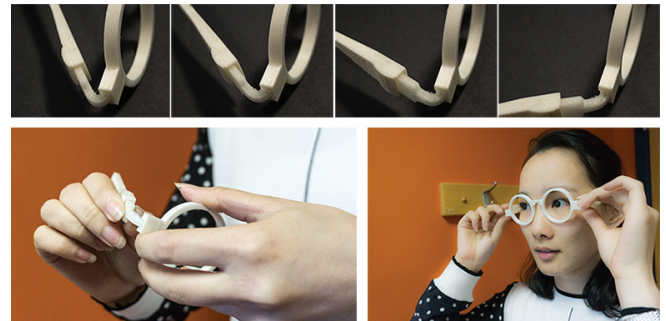
Figure 4. Inserting wire makes adjustable hinges for this pair of Harry Potter glasses. The wire (DOW=1) is drawn and inserted similar to the 'noodle' example shown in Figure 1a.

Density A Bulbasaur phone stand[3] designed for earlier devices is too light to stably hold large smart phones (e.g., iPhone 8 Plus). We embed three nuts to put more weight on its base, leveraging their higher density (Figure 5). Please see our accompanying video figure for a demo.

Softness People with hand injuries or motor impairment might have difficulty holding hand tools. Similar to creating adaptations [3], we can integrate embeddables like a kitchen sponge
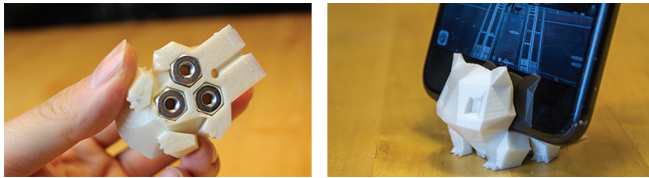
Figure 5. Embedding higher density nuts adds weight to stabilize this printed stand for large smart phones. Medley allows users to draw and 'push' the nuts (DOW=0) into the bottom of the phone stand.
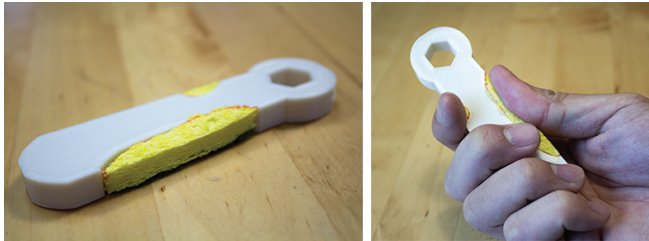
Figure 6. Embedding sponge softens and tightens the grip of this printed wrench. Medley allows users to draw a custom-shape sponge (DOW=3) to replace part of the original wrench handle.

to create a softer grip of a wrench (Figure 6), which also tightens the grip as sponge has higher friction coefficient due to its porousness.

Friction As another example of friction, in Figure 7, a regularly 3D printed door stop would slip on a carpet. Embedding a piece of rubber sheet to replace its bottom increases friction and makes it a functional door stop. Further, rubber also has nice **abrasion resistance**, making the door stop more durable.

Figure 7. Embedding a piece rubber to make a functional door stop. Medley allows the creation of this embeddable (DOW=2) by simple drawing and extruding a line.

Roughness As shown earlier in Figure 1c, embedding a piece of sand paper adds roughness and makes a simple sander.

Flexural strength As shown earlier in Figure 1b, a screw adds flexural strength to a 3D printed peg board hook, as it approximates replacing part of the hook with a stronger material. Embedding a wire straightened from a paper clip into a spoon avoids snapping during breakage, as snapping could be dangerous to the user (Figure 8).

Malleability As shown earlier in Figure 1d, embedding wax (carved from a candle) into an a spoon handle allows users to sand it to a custom shape that the fingers can grip comfortably.

Optical properties As shown in Figure 9, for a 3D printed lamp shade, embeddables (e.g., transparent marbles) can add rich optical properties, e.g., making part of the lamp shade more translucent, refracting light to brighten the upper part that is otherwise a bit dark.

Figure 8. Embedding wire straightened from a paper clip into a spoon avoids snapping, which could be dangerous to the user during breakage. Please also see our accompanying video figure.

Figure 9. Embedding clear marbles into a lamp shade makes it more translucent, and can also refract light to brighten the upper part.

Thermal properties Some embeddables have high heat conductivity, such as coins, which can be used to create simple heat sinks on a printed laptop stand (Figure 10). Others can insulate, such as silicone oven mitts, which can be used to make printed utensils (e.g., tongs) heat resistive (Figure 11).

Having demonstrated a series of embeddable results, the next few sections describe the technical approaches underpinning the design and fabrication process.
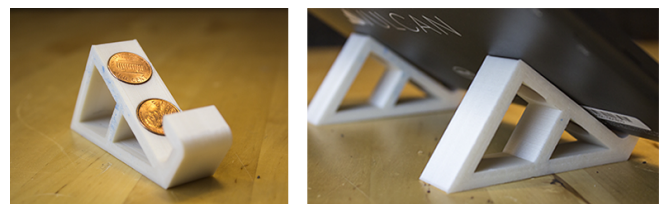
Figure 10. Embedding coins to laptop stands creates a heat sink.

Figure 11. Embedding scraps cut from a silicone oven mitt into a pair of tongs makes it possible to be used with hot objects.

OVERVIEW OF DESIGN AND FABRICATION PROCESS

We develop Medley–a computational tool to support the design and fabrication of objects with embeddables (Figure 12). A design object (e.g., a wrench) can be imported to the tool. As shown in Figure 12a, the library lists each embeddable as a card that contains a thumbnail image, a name, and a short list of material properties. Typing in the search bar finds a specific embeddable, as it dynamically filters embeddables that match the input keywords.

Clicking on an embeddable (e.g., sponge for the wrench) highlights and selects it (Figure 12a), and applies it to the subsequent actions, which we briefly describe here and discuss in further details in the next few sections:

- **Specifying the geometry of embeddables**. Given the DOW of an embeddable, Medley enables the creation of custom embeddable geometry by directly sketching on the design object (e.g., the two cross-sections on the wrench), and manipulating its DOW using simple slider widgets (Figure 12b).

- **Finding optimal paths to insert embeddables**. Once an embeddable is created and placed into the design object, Medley searches for an optimal insertion path with minimal removal of the design object's original volume (e.g., the arrows on the embedded parts on the wrench). Medley supports inserting the embeddable both during and after the printing process (Figure 12c).

- **Generating guides for crafting embeddables**. Finally Medley generates a fabrication-ready 3D model with carved out insertion path, as well as guides to cut embeddables into specified shapes (Figure 12de).

SPECIFYING THE GEOMETRY OF EMBEDDABLES

Once an embeddable is selected, the next step is to specify its geometry *in vivo* on a design object, such as the position and orientation of a fixed-shape embeddable ($DOW = 0$), or the shapes for workable embeddables ($DOW > 0$).

One challenge here is that different DOWs constrains how an embeddable can be created in different ways, which could significantly increase the complexity of the interface. As detailed below, Medley addresses this by providing a unified sketch-based input vocabulary that enables simply drawing on the surface of the design object to initiate the placement of the embeddable, or the creation of its shape. A second challenge is aligning an embeddable to certain parts of the design object. This task could be cumbersome if only using translation/rotation/scaling operations in the Cartesian system, as the X/Y/Z coordinates do not necessarily align with the object's geometry. To address this, Medley leverages the design object's geometry as a constraint to simplify the editing and transformation task, reducing the variables from nine (translating/rotating/scaling with respect to each of X/Y/Z) to at most three (depth, width and thickness). Below we detail our approaches to tackle these two challenges.

DOW = 0 For a fixed-shape embeddable, we pre-compute its principal axis $\mathbf{n_e}$—an axis that represents the global orientation of an object—by first fitting all its vertices to a plane using
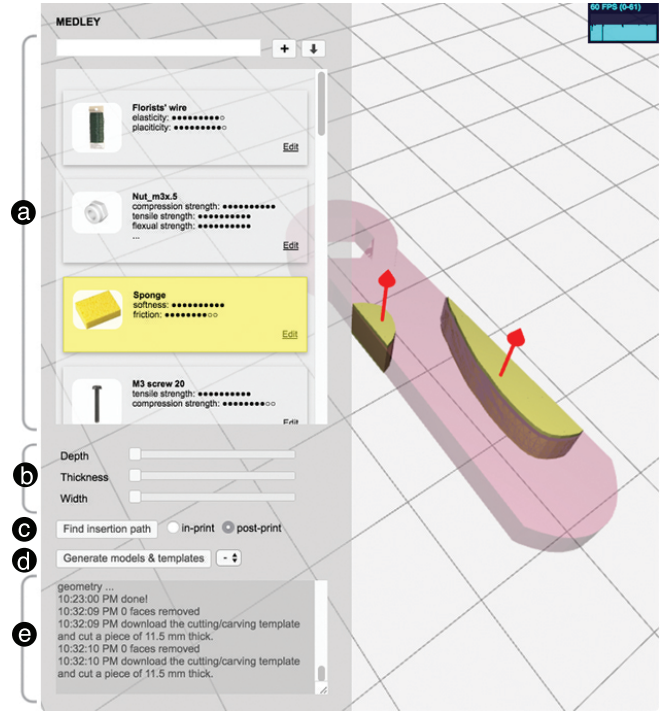
Figure 12. Overview of Medley, the computational tool that supports the library of embeddables, from searching an embeddable by material properties (a), to interactively specifying its geometry and placement (b), to finding an optimal insertion path (c), and to generating fabrication-ready model (d) and instructions for the crafting process (e).
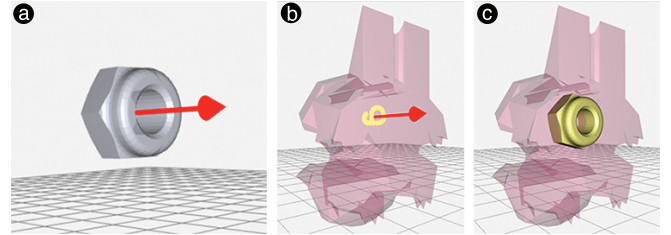
Figure 13. Specifying the geometry of a DOW=0 embeddable (e.g., a nut) by 'drawing' it on a design object (b), which aligns its pre-computed principal axis (a) with that of the sketched points' (bc).

Singular Value Decomposition (SVD) [19], and then taking the plane's normal as $\mathbf{n_e}$ (Figure 13a). As one sketches on the design object, we compute the principal axis $\mathbf{n_s}$ from the sketch points (Figure 13b). First, we convert the sketch points $\{P_0, P_1, ... P_N\}$ to a polyline L. If L is closed (i.e., a forming a polygon), we apply the similar SVD method; if L is open, we compute $\mathbf{n_s}$ as $\frac{1}{N-1}\sum_{i=1}^{N}(P_i - P_0)$ We then position the embeddable at the centroid of the sketch P_c, and orient it by aligning $\mathbf{n_e}$ to $\mathbf{n_s}$ (Figure 13c).

Once the embeddable is aligned and placed on the surface at P_c, we can push it into the design object by adjusting the 'depth' slider. To enable this interaction, we first raycast from P_c along the reverse of its normal, which will hit Q on the other side of the design object. We use P_c and Q as control points: given a depth ratio d read from the slider, we position the embeddable at $(1-d)P_c + dQ$. This approach is applied
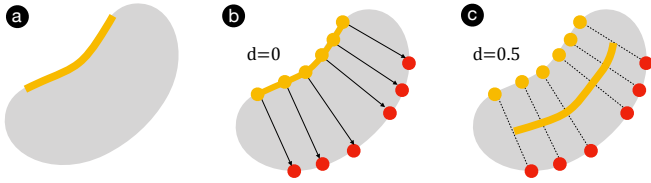
Figure 14. Top view of 'pushing' an embeddable to a design object: after sketching a to create the geometry (a), each sketch point raycasts along the reversed direction of its normal to find another control point (b), and by interpolating pairs of these points we can 'push' in an embeddable, which is driven by a depth value *d* between 0 and 1.

Figure 15. Specifying the geometry of a DOW=1 embeddable (e.g., a wire) by drawing a continuous line (ab) or clicking at discrete points (cd).

to all the embeddables with different DOW's. A more general illustration of the process is shown in Figure 14.

DOW = 1 enables two ways of creating a 'tunnel' for putting in an embeddable: continuously drawing a freeform path (Figure 15ab) and discretely clicking at points that consecutively connect to one another (Figure 15cd). Both actions return a polyline *L*. To make sure the material can be inserted into the tunnel, we straighten *L* so that it does not go under the embeddable's bend radius limit r_e. In particular, we iteratively evaluate each three consecutive vertices on *L*, and fix those whose curving radius is smaller than r_e. We then extrude geometry along the 'straightened' *L* using a circular cross section with the embeddable's radius.

Similar to $DOW = 0$, it is possible to push the embeddable into the design object using the 'depth' slider (Figure 15f). This is implemented by creating two control points for each vertex along *L* and interpolating them with the given depth ratio.

DOW = 2, 3 There are two ways to create geometry for a $DOW = 2$ embeddable (e_2), both of which can be easily extended to a $DOW = 3$ embeddable (e_3):

- Drawing a freeform *open* line *L* (Figure 16), which is similar to the input for $DOW = 1$. However, instead of extruding a 'tunnel', we expand the polyline *L*. First we compute a fitting plane using SVD based on all the sketch points as well as their normals (one such normal is obtained by interpolating the vertex normals of the face that contains a sketch point). Then as shown in Figure 16, we expand *L* bidirectionally along each sketch point's normal (Figure 16b), which creates a plane cutting into the design object (Figure 16c); next, we extrude the plane bidirectionally along its own normal (Figure 16d), thus creating a full geometry representing the embeddable (Figure 16e).

  The amount of these two expansions corresponds to thickness and width, respectively. While $DOW = 2$ embeddables have fixed thickness and can only vary width, $DOW = 3$ embeddables allows for adjustable thickness. Further, both can adjust their depth using the aforementioned control point approach applied on $DOW = 1$.

- Drawing a freeform *closed* line (or pressing 'shift' to automatically close an open line) indicates a polygon *S* rather than a polyline (Figure 17a). To create the embeddable geometry, we first retrieve a set of neighoring faces from the design object within a certain geodesic distance from the center of the polygon (Figure 17b). We then re-tessellate
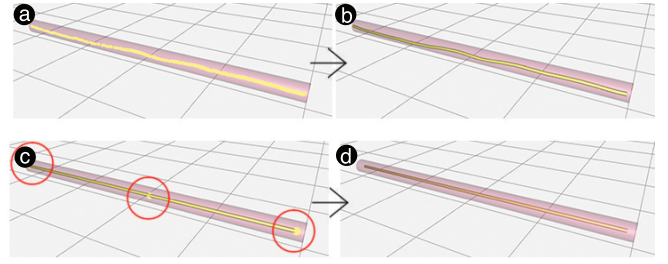
these faces: for a given face *f*, we retain it if its vertices are all in *S*, and remove it if they are all outside of *S*; otherwise we retessellate and include the part of *f* that intersects with *S* (Figure 17cd).

These new tessellated faces create the top side of the embedded geometry (Figure 17e). Next, we again obtain a set of control points from raycasting, which create the bottom side of the embeddable geometry. We then add the lateral area between the two sides to finish the whole geometry.

Depth and thickness are adjustable (with thickness only applicable to $DOW = 3$) using the top and bottom vertices as control points.

To summarize, an embeddable's geometry can be created using a unified technique—sketching on the design object. In other words, selecting an embeddable is similar to choosing a type of paint brush that is subsequently applied to the sketching action. Medley handles the geometric and material details (e.g., DOW, bend radius) while keeping the sketch-based interaction consistently to the minimum across different types of embeddables (e.g., drawing one single stroke). Further, to address the challenge of alignment, Medley uses the design object's geometry as constraint so that the positioning and orienting of an embeddable is in relation to the design object's own geometric and spatial properties, such as using points intersected with the design object as control points to position the depth of an embeddable.

FINDING OPTIMAL PATHS TO INSERT EMBEDDABLES

Once an embeddable's geometry is specified, the next step is to find ways to insert it into the design object. In terms of timing, an embeddable can be inserted while printing the design object (in-print), or after the printing is finished (post-print). We can also choose whether the insertion process allows for deformation of the embeddable. For example, inserting a wire causes it to undergo bending while being inserted; inserting a nail, on the other hand, should require no deformation. In our current system, we only consider deformational insertion for $DOW = 1$ with high stiffness (e.g., metal wires but not threads or strings). Figure 18 summarizes the $2 \times 2 = 4$ ways of inserting embeddables.

Below we provide details for each of these scenarios, in particular how Medley searches for the optimal path by minimizing the amount of the design object that needs to be removed in order to insert the embeddable.
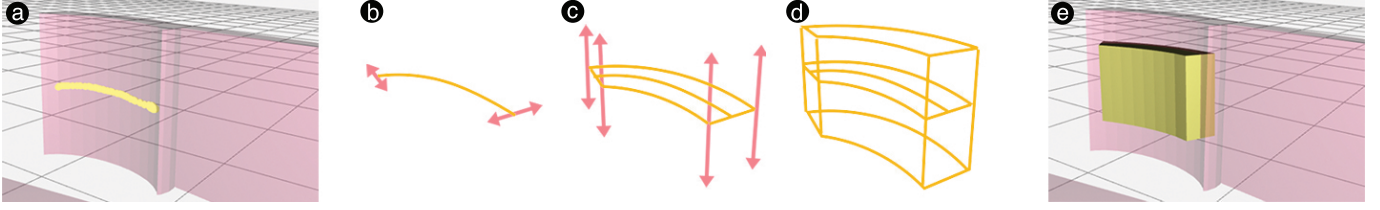
Figure 16. Specifying the geometry of a DOW=2, 3 embeddable by drawing an open freeform line (a), which is then expanded into a 3D shape with controllable thickness and width (b→e).

Figure 17. Specifying the geometry of a DOW=2, 3 embeddable by drawing an (automatically) closed line (a), which is then retessellated to a surface extrudable into a 3D shape (b→e).b

Deformational Insertion of DOW=1 Embeddables

For $DOW = 1$ we can create a polyline 'tunnel' L to contain the embeddable inside the design object. To generate a path for inserting the embeddable, we extend L to the nearest surface. Medley computes such paths for both in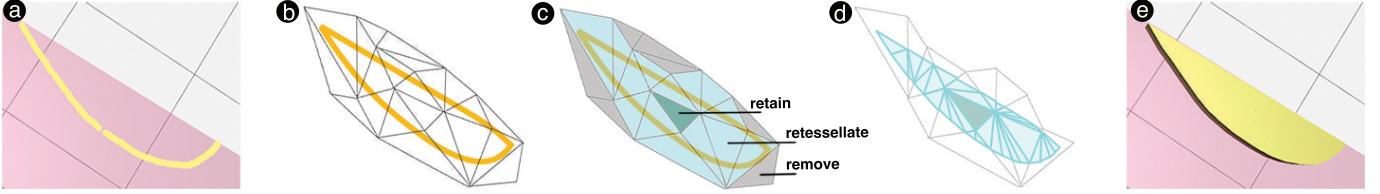-print and post-print insertions, and provides a solution with minimized path length. The first column of Figure 18 illustrates the follow insertion path finding scenarios.

In-print insertion assumes the entrance of insertion is at the layer where the print job will be paused and the embeddable inserted. First we orient the design object to a user-specified printing direction. Next, along that direction we compute the pausing point, which is a point on L with the highest value when projected on the printing direction. Now, we can extend either one of L's end points to the layer defined by the pausing point. As shown in Figure 19, we compute the tangent vector of an end point, and generate a circular extension path whose radius equals the embeddable's bend radius. Amongst all such possible paths in 3D space, we discard those that cannot reach the pausing layer, and for those that can, we search for the one with the shortest length. We perform this process twice, once for each end point, and obtain the shortest path to insert the embeddable at the pausing layer.

Post-print insertion We perform a similar shortest extension path searching process, except the destination is on the surface of the design object, rather than a horizontal sliced plane.

Non-deformational Insertion of DOW=0, 2, 3 Embeddables

For non-deformational insertion, the insertion path needs to accommodate a sliding volume of the embeddable's geometry (e.g., without undergoing bending). Given a user-defined printing direction, Medley finds a insertion path that minimizes such volume.

In-print insertion. Similar to the deformational case, we first compute the pausing layer. Next, we perform a step-wise (step=5°) search across possible insertion directions at the pausing layer. For a given insertion direction θ, we compute
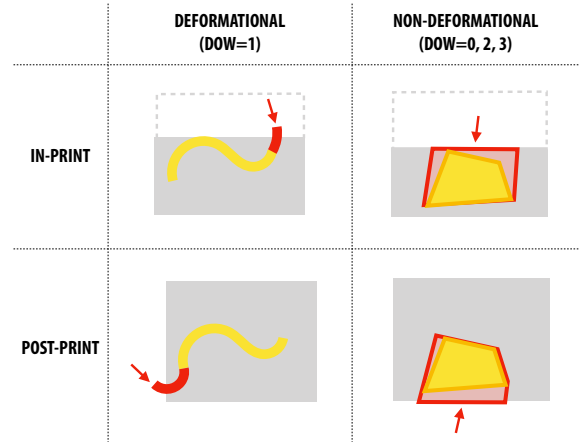
Figure 18. A 2x2 space of inserting embeddables that considers (i) whether insertion occurs during or after printing the design object; and (ii) whether the embeddables undergo deformation during insertion.

the minimum volume of the design object that needs to be removed in order for the embeddable to be inserted. To compute this volume, we first discretize the embeddable based on the printing layer height Δh: given θ, the to-remove volume can be discretized into N layers, each perpendicular to θ and has a height of $\Delta h / \cos(\theta)$ as shown in Figure 20.

The minimum volume of the entire N-layered embeddable can be computed as $V_{min} = \cup_{i=1}^{N} v_{min}^{(i)}$, where $v_{min}^{(i)}$ is the minimum volume for inserting the ith layer, which can be obtained by
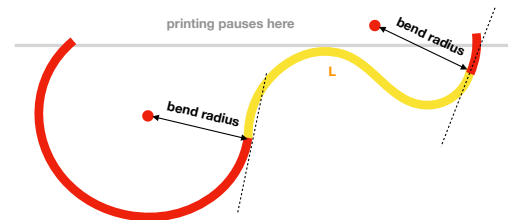
Figure 19. Finding an optimal insertion path for DOW=1 by extending each of the end points along its tangent vector, and along a circular path of radius equal to the bend radius of the material. We compute and search for the shortest amongst all possible paths.
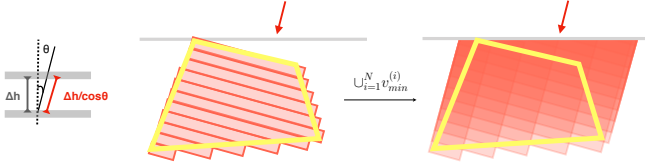
Figure 20. Finding an optimal insertion path for DOW=2, 3: for a given insertion direction θ, we compute the minimum volume that needs to be removed in order to insert an embeddable; we then search step-wise across possible insertion directions to find one with the minimum to-remove volume.

sliding (or extruding) that layer from its position along θ until its entirety is outside of the (partially printed) design object; $\cup$ is a union operator. As shown in Figure 20, the minimum volume for inserting an embeddable is the minimum volume for inserting each of its layers at their corresponding position. Our search process finds θ_{min} with the minimum V_{min} amongst all possible insertion directions.

Post-print insertion. We perform a similar search process, except for each θ the volume for insertion needs to start from the surface of the design object, rather than a pausing layer.

GENERATING GUIDES FOR CRAFTING EMBEDDABLES

As a final step, we perform a Boolean operation, subtracting the geometry of the insertion path from the design object. As shown in Figure 21, one problem is that this operation will generally leave an 'hole' on the object's surface (for post-print insertion), or in its inner structure (for in-print insertion). To mitigate this problem, we generate a 'cap' that can be 3D printed to fill in the hole once an embeddable is inserted. Figure 23 shows a example of embedding a nut and a cap (printed in advance) during the printing process.

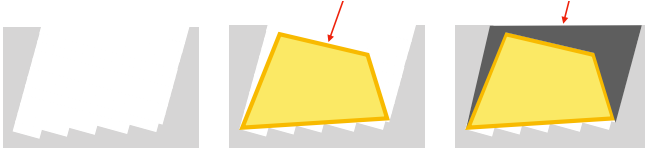Figure 21. Generating a cap to fill up empty space created by removing volume for the insertion path

For non-deformational insertion ($DOW = 0, 2, 3$), we first compute the convex hull of the embeddable, and then subtract it from the geometry of the insertion path to obtain the cap. This approach, however, does not work for deformational insertion, as the printing material might not have the bend radius to allow it to be inserted into the hole. Second, to insert a thin wire/rod, one often needs to cut it longer than the insertion path in order to hold and push it in, which will already fill the hole once inserted. Given these considerations we currently do not to add a cap to deformational insertions.

Embeddables with non-zero DOW also needs to be cut or carved to match the geometry specified in the tool. We generate guides to aid this crafting process. For $DOW = 1$, we output the length l of the embeddable by summing up the distances of the segments on the polyline drawn by the user. For $DOW = 2, 3$, there are two types of geometry. One is extruded from a polyline, and thus can be obtained by cutting or carving a $l \times w \times t$ piece, where l, w and t correspond to the drawn length, the adjustable width, and the embeddable's material

thickness (fixed for $DOW = 2$, adjustable for $DOW = 3$), respectively. The other type of geometry is extruded from a polygon. We generate a thin piece of the polygon that can be printed and then used as templates for cutting/carving a piece of embeddable with t thickness. For example, to craft the embeddables for the wrench (Figure 6), we printed two templates alongside the wrench (Figure 22a). As shown in Figure 22b, we then used the templates to trace the designed profile on a sponge and then carved out the embeddable with the thickness informed by Medley's output (Figure 12e).

We apply two strategies to attach embeddables to printed objects. First, having calibrated our 3D printers, we slightly scale the generated insertion path (between 0.5 and 1 mm smaller) to press fit the embeddables, such as fitting a wire into a 'noodle', and fitting a piece of sand paper into the slot on a printed sander (Figure 1c). In other cases where the objects and the embeddables cannot press into one another, we use a versatile and temperature resistant adhesive[4].

FABRICATION AND SOFTWARE IMPLEMENTATION

All the results were printed using an Ultimaker 3[5] and a 'homemade' FDM printer built in our lab. We printed the 'noodle', the door stop, the lamp shade, and the glasses' hinges using TPU and the rest using PLA. All the embeddables were either found in our lab or purchased from Amazon. We used wire cutter and utility knife for cutting embeddables into shape. Medley was implemented using JavaScript and three.js[6], and runs in a Google Chrome Web browser.

DISCUSSION

In this section we discuss outstanding issues, existing limitations and future work for embeddables.

Generalizing to other fabrication processes/materials Although all of our examples were made using 3D printing, we expect the design space to be generalizable to a broader set of fabrication methods. For example, users can use a CNC machine to create the same sander example in Figure 1c, or use casting to create the utensil holder in Figure 1d. However, different fabrication methods have limitations. For example, casting will have trouble with the example in Figure 23 as it will require creating a cavity *fully* inside a 3D object; laser cutting is somewhat limited to embeddables extruded from 2D patterns and with limited thickness.

---

[4]Currently we use Gorilla Original Glue: http://www.gorillatough.com/gorilla-glue
[5]Ultimaker 3: https://ultimaker.com/en/products/ultimaker-3
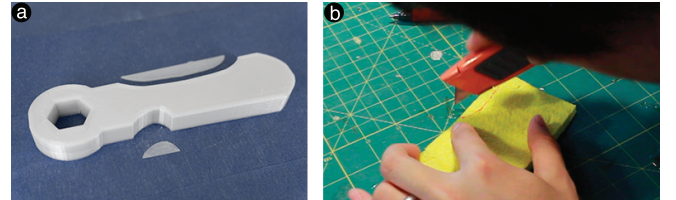[6]three.js: https://threejs.org

Figure 22. For freeform custom embeddable geometry, we generate printable templates to assist the crafting process: in the case of making the soft wrench handle, we can use the printed templates to trace the embeddable's shapes on a sponge before carving them out.
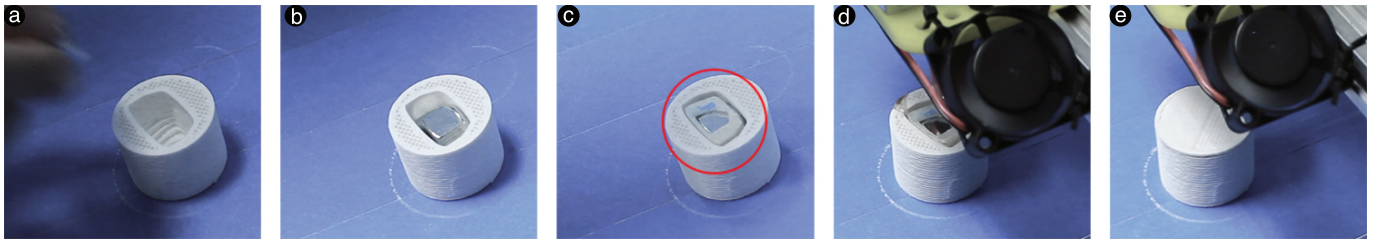
Figure 23. Demonstration of inserting a cap during the printing process: an embeddable (a nut) is inserted (ab), and then a cap (c), after which the printing continues (d) and finishes (e).

In terms of material, we focus on cutting finished material from found objects; future work can extend the design space to address other types of operation, other types of material, such as exploring the distinction of raw vs. finished material.

Adding to the current library of embeddables Our embeddable library is extensible. As shown in Figure 12, clicking 'Edit' on each embeddable card brings up a dialog for editing the embeddable. We can change the basic information, such as clicking the name to edit it, or dragging a new image to the dialog to update the thumbnail. More importantly, one can specify information related to material properties. The first step is to select the DOW, based on which the dialog further request additional information, such as a 3D model for $DOW = 0$ object, radius for $DOW = 1$ (e.g., radius of a wire), thickness for $DOW = 2$ (e.g., thickness of a sheet), and minimum bend radius for $DOW \in \{1, 2\}$ (e.g., how much a wire or a sheet can be bent). Using the a similar dialog, we can also add a new embeddable to the library.

Practical issues in obtaining embeddables The current version of Medley allows users to reuse found objects they already have and create individual embeddables for them. However, a practical issue is such embeddables might not always be available. Future work could solve this problem by suggesting similar embeddables if users do not already have them at hand.

Exploring other types of embedding techniques Our examples were made with the very basic and effective embedding technique, namely creating a snug-fit cavity/negative space within a design object. Future work could explore different methods to achieve similar embedding results, e.g., molding, or printing-over directly from an embeddable [2].

Guidance for quantifying and comparing embeddables Currently our library employs a simple rating approach (scale 1 to 10) to quantify the material properties of embeddables, which enables searching for specific embeddables with a target property. As the library grows larger, it would be useful to compare material properties across embeddables, and sort the search results accordingly. One possible direction for future work is to employ [28]'s method, using the crowd to perform pair-wise comparison for a given material property, based on which a ranking can be computed across all embeddables. Further, we also would like to explore ways to inform users of different material properties and provide more support for them to understand and explore the usage of different embeddables.

Limitations of workability and insertion While our library and fabricated examples demonstrate a wide range of embeddables, there are objects that cannot be used as embeddables due to a lack of workability or difficulty with insertion. For example, a bike spoke has a nice combination of tensile and compression strengths, and has a small dimension to fit in thin structures; however, it is difficult to cut without specialized tools. Our future work plans to provide a list of tools, which can be incorporated into the library, as a way to inform users how to cut or carve a particular embeddable. Some objects are easy to cut, but much more difficult to insert into a printed design. For example, carbon fibre has great structural properties and has been used in industry for strengthening components such as bike frames or airplane turbines. However, a single thread of carbon fibre is fairly soft and very difficult to insert into a 'tunnel' in ways similar to inserting a metal wire. In the future we plan to develop ways to insert such soft embeddables (e.g., perhaps by extending [13]'s approach with fabric).

Simulation to inform embeddable choice and design While our library and tool allow people to explore and design with different embeddables, as future focus we should also provide more guidance on: *(i)* what are the existing problems of the design, *(ii)* what embeddables can be used to solve or mitigate such problems, and *(iii)* once inserted whether an embeddable has improved the design.

Automatically suggesting and generating designs Even with simulation, it still requires manually choosing and applying embeddables that are selected from a potentially large library. In the future we will experiment with ways to automatically generate design suggestions based on the input object and problem description (e.g., "increasing the strength around this area"), which is similar to approaches demonstrated in prior work [21]. Further, it is also interesting to search for existing designs from other users that match the current design problem, and also use them to generate more design suggestions.

CONCLUSION

To approximate a future of multi-material printing, we have constructed a library (and the tool support) of embeddables– everyday objects that can be cut, worked and embedded into 3D printed designs to augment their material properties. We hope this work can promote future research on end-user material literacy, as well as inspiring designers and makers to explore the material aspects of personal fabrication.

ACKNOWLEDGMENTS

REFERENCES

1. Bernd Bickel, Moritz Bächer, Miguel A Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and fabrication of materials with desired deformation behavior. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 63.

2. Xiang 'Anthony' Chen, Stelian Coros, Jennifer Mankoff, and Scott E Hudson. 2015. Encore: 3D printed augmentation of everyday objects with printed-over, affixed and interlocked attachments. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. ACM, 73–82.

3. Xiang 'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E Hudson. 2016. Reprise: A Design Tool for Specifying, Generating, and Customizing 3D Printable Adaptations on Everyday Objects. In *the 29th Annual ACM Symposium on User Interface Software and Technology*. ACM.

4. Daniel Groeger, Elena Chong Loo, and Jürgen Steimle. 2016. Hotflex: Post-print customization of 3d prints using embedded state change. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 420–432.

5. Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. 2016. Metamaterial mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 529–539.

6. David RH Jones and Michael F Ashby. 2011. *Engineering materials 1: An introduction to properties, applications and design*. Elsevier.

7. Gierad Laput, Eric Brockmeyer, Scott E Hudson, and Chris Harrison. 2015. Acoustruments: Passive, acoustically-driven, interactive controls for handheld devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2161–2170.

8. Dingzeyu Li, Avinash S Nair, Shree K Nayar, and Changxi Zheng. 2017. AirCode: Unobtrusive Physical Tags for Digital Fabrication. *arXiv preprint arXiv:1707.05754* (2017).

9. Mosaic Manufacturing. 2017. Palette+ printer. `https://www.mosaicmanufacturing.com/pages/technology`. (2017). (Accessed on 08/21/2017).

10. Markforged. 2017. High Strength 3D Printing. `https://markforged.com/`. (2017). (Accessed on 08/22/2017).

11. Gianluca Memoli, Mihai Caleap, Michihiro Asakawa, Deepak R Sahoo, Bruce W Drinkwater, and Sriram Subramanian. 2017. Metamaterial bricks and quantization of meta-surfaces. *Nature Communications* 8 (2017), 14608.

12. Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic textures for additive fabrication. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 135.

13. Michael L Rivera, Melissa Moukperian, Daniel Ashbrook, Jennifer Mankoff, and Scott E Hudson. 2017. Stretching the Bounds of 3D Printing with Embedded Textiles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 497–508.

14. Valkyrie Savage, Ryan Schmidt, Tovi Grossman, George Fitzmaurice, and Björn Hartmann. 2014. A series of tubes: adding interactivity to 3D prints using internal pipes. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 3–12.

15. Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. Capricate: A fabrication pipeline to design and 3D print capacitive touch sensors for interactive objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 253–258.

16. Martin Schmitz, Andreas Leister, Niloofar Dezfuli, Jan Riemann, Florian Müller, and Max Mühlhäuser. 2016. Liquido: Embedding Liquids into 3D Printed Objects to Sense Tilting and Motion. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2688–2696.

17. Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to control elasticity in 3D printing. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 136.

18. Pitchaya Sitthi-Amorn, Javier E Ramos, Yuwang Wangy, Joyce Kwan, Justin Lan, Wenshou Wang, and Wojciech Matusik. 2015. MultiFab: a machine vision assisted platform for multi-material 3D printing. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 129.

19. Gilbert Strang. 2011. Introduction to Linear Algebra. (2011).

20. Stratasys. 2017. Objet260 Connex3. `http://www.stratasys.com/3d-printers/objet260-connex3`. (2017). (Accessed on 08/21/2017).

21. Michael Terry and Elizabeth D Mynatt. 2002. Side views: persistent, on-demand previews for open-ended tasks. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*. ACM, 71–80.

22. Kiril Vidimce, Alexandre Kaspar, Ye Wang, and Wojciech Matusik. 2016. Foundry: Hierarchical material design for multi-material fabrication. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 563–574.

23. Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik. 2013. OpenFab: a programmable pipeline for multi-material fabrication. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 136.

24. Guanyun Wang, Lining Yao, Wen Wang, Jifei Ou, Chin-Yi Cheng, and Hiroshi Ishii. 2016. xPrint: A Modularized Liquid Printer for Smart Materials Deposition. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5743–5752.

25. Daniel Wigdor and others. 2016. Foldem: Heterogeneous object fabrication via selective ablation of multi-material sheets. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5765–5775.

26. Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. 2012. Printed optics: 3D printing of embedded optical elements for interactive devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 589–598.

27. Karl DD Willis and Andrew D Wilson. 2013. InfraStructs: fabricating information inside physical objects for imaging in the terahertz region. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 138.

28. Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. 2015. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 86.