MetaMorphe: Designing Expressive 3D Models for Digital Fabrication

Cesar Torres **Eric Paulos**
Electrical Engineering and Computer Sciences
University of California, Berkeley
{cearto, paulos}@berkeley.edu

Figure 1. a) the UC Berkeley Campanile embedded with audio data from Marco Savio's "Bodies upon the gears" speech, b) *EarthquakeWare*, utensils and tableware expressing seismograph data near you, c) small multiples of airplane and bird forms reformed and retargeted using a MetaMorphe style.

ABSTRACT

The creative promise of 3D digital fabrication tools is tremendous. However due to the wide range of tools and interfaces, a common static file format called STL is used for sharing designs. While customization tools add creative handles to these digital models, they are often constrained to pre-configured parameters limiting the creative potential of shared digital models. We introduce MetaMorphe, a novel digital fabrication framework that uses a common web-programming metaphor to enable users to easily transform static 3D models into re-formed, re-made, and re-imagined customized personal artifacts. We demonstrate the compatibility of MetaMorphe with three well-established design interfaces, direction manipulation, scripted-CAD, and generative design. Through a user study with design experts, MetaMorphe reveals that decisions that physically produce bespoke artifacts or encode unique metadata actively affect perceptions of authorship, agency, and authenticity. We discuss how expressive model-building tools such as MetaMorphe enable a cultural shift in 3D design in terms of participation, personalization, and creativity.

Author Keywords

digital fabrication; creativity support tools; design; DIY;

ACM Classification Keywords

D.2.2 Design Tools and Techniques: User interfaces

C&C 2015, June 22–25, 2015, Glasgow, United Kingdom.

DOI: http://dx.doi.org/10.1145/2757226.2757235

INTRODUCTION

"The authenticity of a thing is the essence of all that is transmissible from its beginning, ranging from its substantive duration to its testimony to the history which it has experienced"
- Walter Benjamin (1936)

Just as Benjamin noted how objects are fundamentally transformed through *mechanical* reproduction, the way we see and make objects is being challenged by today's *digital* fabrication (DF) technologies. Even now grassroots innovation in the Maker Movement is addressing critical themes in education, manufacturing, and health care [4, 17]; more opportunities exist for design technologies to be even more accessible.

Currently, artists and hobbyists freely share their designs on online repositories like Thingiverse [20]. However in order to span modeling tools and for convenience, files are shared in a lowest-common-denominator format called the STereoLithography (STL) which unfortunately only encodes a static mesh. Emerging tools such as Thingiverse Customizer[1] expand the variety of forms from a single design; however such parametric designers have been found to be less than engaging. In a quantitative analysis of digital models on Thingiverse, Oehlberg, et al., observed that Customizer objects make up 74% of remixed objects, yet rarely elicit subsequent user activity or contribute additional content to Thingiverse [23]. Other sites such as GrabCAD[2] provide original CAD files, however altering these models depends on a user's ability to use more sophisticated CAD tools. Thus for a novice user, designing a model quickly breaks down to finding the "correct" pre-made model, downloading it, and printing it. While initially satisfying, this static design practice generally prevent users from critically engaging with either the form or function of a printed object.

---

[1] `http://www.thingiverse.com/apps/customizer/`

[2] `https://grabcad.com`

The current state of 3D modeling tools and sharing practices encourage *designs to favor replication and sameness*. While this "copy exact" is indeed a strength of 3D modeling, we argue that design tools are needed that actively engage users and invite new creative opportunities for variation, personalization, derivation, and versioning (Figure 1). We imagine models that are encoded with multiple forms, easy to derive and extend, and incorporate new digital practices. We term this metamorphic design, and introduce MetaMorphe — a DF design framework for modifying static, digital meshes and creating customized, personal artifacts. This paper makes two contributions to DF design tool research.

First, we introduce a JavaScript framework which takes a new look at CAD, and separates modeling into structural, style, and interactivity concerns. This allows developers to flexibly interact with a mesh in a form similar to web programming and quickly create engaging modeling interfaces that are tailor-fit to different creative practices.

Second, MetaMorphe presents a "meta-design" space through a parallel interface (Figure 4) consisting of multiple interaction styles. This interface allows users to engage with *form* through direct manipulation, inspect and modify *functionality* through scripted CAD, and produce multiple variant *styles* of a single design through a generative interface. By using a common underlying framework, MetaMorphe provides a fluid transition between each interface style, allowing users to move between and reflect on different design priorities.

This paper first motivates the design of MetaMorphe through related work. We then outline the rationale behind our web-inspired framework, provide a grounding example of interaction with the tool, and present supported design practices. Lastly, we evaluate the MetaMorphe interface through a workshop study with creative experts, and conclude by discussing implications of our findings for digital fabrication.

RELATED WORK

MetaMorphe is inspired by emerging DF design tools and concerns arising from communities of artists, makers, and industry. We also examine creative design practices that are redefining digital interactions with physical media.

Sharing models

As digital fabrication matures, a revised STL format is proposed to support new techniques such as multimaterial printing [9]. Similarly, Reprap hosts a forum[3] for discussing improvements to the STL format. Such proposals include adding functional metadata, altering data structures, or encoding a voxel representation. Eschewing a static file altogether, OpenFab proposes a GPU-like pipeline for supporting multi-material prints; geometries are procedurally evaluated thereby reducing the memory footprint and startup time [34]. Autodesk has incorporated iPart and iAssemblies, a parametric design widget that allows designers to specify dynamic components and generate derivative parts. MetaMorphe suggests the need to incorporate multiple design instances in model representations in order to encourage derivative work.

Crafting interfaces

Several studies examine how modeling tools can be made more accessible to users outside mechanical design. Most relevant to our work, Jacobs, et al., *Codeable Objects* is a tool that enables novice users to produce personal and functional objects through parametric models and generative patterns. In her study, Jacobs confronted a tension between the inherent dissimilarity between traditional fabrication techniques and computational design tools [11]. Such dissimilarities have traditionally been absolved by incorporating more traditional craft processes in the design of digital tools [19]; this continues to be a current trend:

In software, MeshMixer engages with physical forms through mesh mixing, or the collage and hybridization of multiple models. Sculpting metaphors (e.g. pinch, tug) are widespread in tools like Autodesk Sculpt. Hybrid techniques such as Nervous System[4] sculpt a mesh topology by subdividing geometries using a brush interaction. A proprietary online community has developed from these tools centered around model sharing, demonstrating that this type of customization is highly desired by users. MetaMorphe further extends these sharing practices to include models that are encoded with multiple designs, and provides a mechanism for users to "source" a design's history that is often lost.

Tangible interfaces are more akin to the physical making process. *FreeD* [39], a reductive sculpting tool, provides tactile feedback to novices and interactively guides them as they sculpt a digital model freehand. Alternatively, real world objects are collaged and digitized into new forms and shapes [7, 8]. MixFab, a mixed-reality CAD interface, allowed users to carry out CAD operations using gestural manipulation [35]. Similarly, sketch-based editing in *ModelCraft* is enabled by capturing physical annotations and converting them to operations on digital models [31]. While these types of tools enable more natural interactions for constructing and manipulating objects, the space of operations is more restricted to those already present in the equivalent physical media [10]. In work on digital tool building for maker communities, Jacobs, et al., suggests CAD tools should enable designers to "reconfigure virtual and physical modular parts through a small number of operations that are derived from the topology of the parts themselves" [12] as an alternative to a full-featured *tabula rasa* CAD interface. Through parallel interfaces, MetaMorphe supports both *crafting* interfaces with direct manipulation and *computational* interfaces with generative design and scripted-CAD interactions on *existing* designs.

Scripted CAD

Open initiatives like openSCAD [16] and openJSCAD [21] are gaining footholds within the scripted-CAD community. In particular, these tools allow programmers to produce end-user interfaces for customizing models which allow users to alter features such as width, curves, or text. These customization tools engage novice users by providing creative handles to otherwise static designs. However the programming scheme for developing these interfaces still relies on domain knowledge of computational geometry. To the end-user, their choice

---

[3] `http://reprap.org/wiki/A_community_specification_for_an_improvement_to_STL_files`

[4] `http://n-e-r-v-o-u-s.com/`

of parameters is already fixed by the interface designer. A large number of authors (42%) on Thingiverse only produce generated designs and never contribute other content, suggesting that many authors lack the technical expertise necessary to modify designs using CAD tools [23]. While these tools provide some customization, they do not go further to explore more expressive and creative design a user might envision. Autodesk's Project Dreamcatcher [1], an experimental generative modeling tool, proposed a higher-level shape descriptor for specifying function and form. While not as abstracted, MetaMorphe exposes customization code through human-readable style sheets, providing a scaffold for users alter and build upon existing designs.

Bespoke Fabrication: Outside the"copy-exact" paradigm

Though "copy-exact" is a major benefit of mechanical reproduction, DF design tools have opened new opportunities for customization. However in the rush to "copy exact", there has been little attention to modeling tools that value personalized or unique artifacts. Below we detail the emerging role of bespoke fabrication in practice and production.

One-off designs

The introduction of chance and meaning in the design process has been used to digitally fabricate unique artifacts. For instance, Zoran et. al. incorporated history by adding breaks and repairs into the form of digitally fabricated artifacts [38]. Different forms of data have been used to add personal meaning: stories and memories are used to shape knitted crafts [27], environmental data is used to shape artistic sculpture [37], or even a life-logged relationship is embedded in the design of matrimonial rings[5]. Other approaches manipulate the fabrication process. *FreeD* allowed users physical control of the DF process leaving impressions of "the hand of the artist"[39]. Willis et. al. used interactive audio and gestural input to give form to artifacts in a series of *Interactive Fabrication* tools [36]. Our paper aims in foregrounding these unique and bespoke interactions in the design of metamorphic models.

Digital Fabrication as an Artistic Medium

Artists have shown that digital fabrication is a critical disruptive medium that can promote a new class of designs. For instance, Tarik Sadouma's *nike town 2* utilizes the rubber treads of a shoe to create dynamic urban landscapes [28], while *DERRICK* [33] features *digital* IKEA furniture models which have been hacked and "infected" to produce misshapen forms that materialize a *biological* phenomena. Engaging directly with 3D printing aesthetics, Artist LIA subverts the layer-by-layer DF process with *Filament Sculptures* [18] by producing custom g-code, or printhead instructions, to produce continuous free-form lines to form a type of 3D thread art. Plummer-Fernandez explores a different sharing practice in *Disarming Corrupter*, a protocol which encrypts and corrupts digital models [26]. MetaMorphe provides an open method for artists and designers alike to explore the physical language of artifacts in order to critically engage with everyday forms.

SYSTEM DESIGN

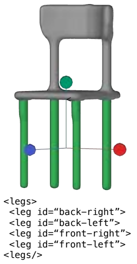
As we developed methods for designing dynamic digital models, we quickly realized that we needed a flexible way to create and iterate on different design interfaces. In this section, we cover the design rationale behind the MetaMorphe framework and how this enables developers to quickly create interfaces for manipulating a mesh.

MetaMorphe is a web-based framework written in JavaScript and extends the THREE.js[6] WebGL library. As the *lingua franca* of the web, building the framework using JavaScript provides easy access to several external APIs (e.g., weather data, webcam, microphone, and GPS) that can enable novel interaction techniques. The framework uses Separation of Concerns (SoC) — a common software pattern used in computer science to separate a program into distinct sections. This pattern appears prevalently in web architectures as HTML/CSS/JS and in HCI at the Model-View-Controller. Inspired by the Web SoC, we incorporate "view-source", a functionality known to encourage open cross-disciplinary design practices driven by learning from the work of others [24]. As an added benefit, SoC implicitly creates disciplines, allowing users to develop specialized skills using a subset of concerns (e.g., visual design, interaction design), and enables a scaffold for specialized users to adopt complementary skills gradually. The MetaMorphe framework is divided into three concerns: *structure*, *dynamic style*, and *interactivity*.

Structure concern - Mesh selection

Moving individual points on a mesh is a tedious process; selecting elements in a design based on a higher-order arrangement or relationship aligns with many traditional design tasks. For instance, similar elements in a design can be grouped together, breaking down complex design problems into modular pieces. This enables interactions like "select all limbs" or "select sibling finger". In particular, hierarchical structures have proven as effective selection schemes, allowing users to focus on smaller conceptual components.

Using STL models from the Princeton Benchmark [5], and segmentations and labels provided by [13], we decomposed a mesh into an XHTML tree (Figure 2). The mesh is separated into *regions* based on form semantics (e.g. **chair** → ⟨**support**, **backsupport**, **leg**⟩). This allows designers to custom-annotate and group components to suit their specific needs. Similar to HTML, attributes such as **id** and **class** can be used to group and identify regions.



```
<legs>
 <leg id="back-right">
 <leg id="back-left">
 <leg id="front-right">
 <leg id="front-left">
<legs/>
```

Style concern: 3D Modeling Style Sheets

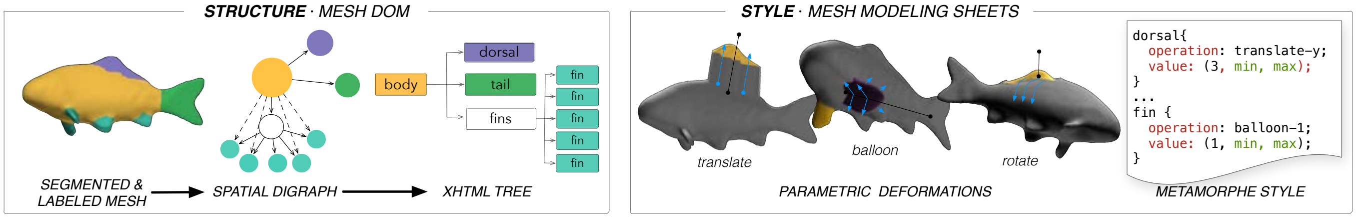
An iterative design process requires the ability to prototype quickly and interchangeably. Using a stylesheet paradigm similar to CSS, we specify a concise language for controlling how an artifact's form changes.

For each structural element, or region, in a mesh, we impose a simple parametric model described by Kho et.al. [14]. This technique binds vertices on a mesh, or skin, to a reference curve, or bone. This allows simpler deformations to the bone

---

[5] `http://www.diarings.com/`

[6] `http://threejs.org/`



```
dorsal{
  operation: translate-y;
  value: (3, min, max);
}
...
fin {
  operation: balloon-1;
  value: (1, min, max);
}
```

Figure 2. MetaMorphe Framework. Extracting structure (left) begins with segmenting the mesh, and then using the regions (colored circles) to construct a directed graph. Dummy nodes (white circles) can be used to group regions with similar volumes (e.g., fins); the minimum spanning tree (solid arrows) provides a hierarchy served to the user as XHTML. Applying styles (right) consists of constructing a MetaMorphe style to specify the parameters of one of three deformations and the mesh region it will affect.

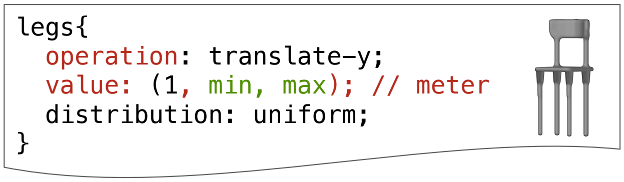
to deform the skin while preserving local geometry. For out-of-the-box interaction, bones are specified as a region's principal component. We implement the following common deformations on each region:

1. *translate* moves a region along a selected axis,
2. *balloon* scales a region relative to a selected axis,
3. *rotate* moves a region about a selected axis.

A region can be deformed by specifying the following parameters through a *style* (Figure 2) described by the following properties:

- **operation** - [translate, rotate, balloon]
- **axis** - direction of deformation
- **value** - strength of the deformation
- **value-type** - unit of measurement (length, angle, volume)
- **min**, **max**, **distribution** - constraints on value
- **smooth** - controls deformation propagation
- **texture** - displaces vertices based on a heightmap

These properties are chosen to provide the basic functionality of direct manipulation, such as the conversion of a chair to a bar stool (Figure 3). In shorthand notation, the style translates the legs of a chair by one meter relative to the y-axis. More dynamic design properties enable users to specify ranges with the **min** and **max** property, as well as how values are sampled from the range using the **distribution** property. Lastly, not depicted, **texture** specifies a displacement map for encoding values on the surface of the mesh, described in §§ Designing with Data. We leave out traditional graphics properties such as specifying: **material, reflectance**, etc.., to focus on styles that alter form.



```
legs{
  operation: translate-y;
  value: (1, min, max); // meter
  distribution: uniform;
}
```

Figure 3. A bar stool style for Princeton Benchmark #120 - Chair

Interactivity concern: Feature representation

Designing how a digital model responds to a condition or input and produces a relevant output is difficult due to the cost of loading and rendering new geometries. To allow for a flexible way to alter designs quickly, we convert each set of applied styles into a feature vector. This feature representation captures mesh properties at a specific *style-instance* akin to a *key frame* or *morph target* in traditional animation. The feature vector can be decomposed into two-parts:

- *Metadata* - this represents non-real values, such as the type of operation (e.g. translate) used. This allows a given feature vector to be backwards-compatible, i.e. can be parsed back into style treatments.
- *DNA* - this represents the appearance of the mesh and contains the real-valued properties of each style. This DNA has both a phenotype (i.e. the actual external physical representation) and a genotype (i.e. a range of potential but unexpressed physical traits). When phenotype subvectors of different designs are averaged, it generates a new subvector which represents the "mixture" of those two *style-instances*. In comparison, a genotype subvector needs only itself to generate new vectors based on values encoded by **min, max** and **distribution** style properties.

We describe example interactions with feature vectors in §§ Generative Interface and §§ DNA as a Metaphor.

METAMORPHE INTERFACES

Using MetaMorphe we created three parallel interfaces (direct manipulation, scripted, generative) to showcase the compatibility of this SoC framework with interaction techniques used by different creative practices.

Direct Manipulation Interface

Due to the natural translation from physical design and the need for realtime feedback, direct manipulation is the current *de facto* standard for 3D modeling. Especially for users without a programming background, direct manipulation offers the shortest learning gap. In order to utilize the natural exploratory property of this interface, we utilize the *generator* design pattern — whereby specific design tasks (e.g., choosing a color) are streamlined and produce only the relevant information needed to achieve the task (e.g. a RGB value).

The interface is decomposed as follows: an STL model is loaded onto a central screen (Figure 4A). Following conventions of industrial CAD interfaces, three operations are exposed to the user as noun-verb actions: balloon, rotate, translate. Axis handles used to control the strength of an operation. A form on a separate pane holds all possible values for a *style*, and is brush-and-linked such that any direct manipulation to the mesh updates the relevant values in the form. Lastly, an "export" button generates MetaMorphe style text.

Scripting Interface

A programmatic representation provides expert control of the mesh and foregrounds interactive design, however it also introduces the largest semantic gap for novice users. To provide

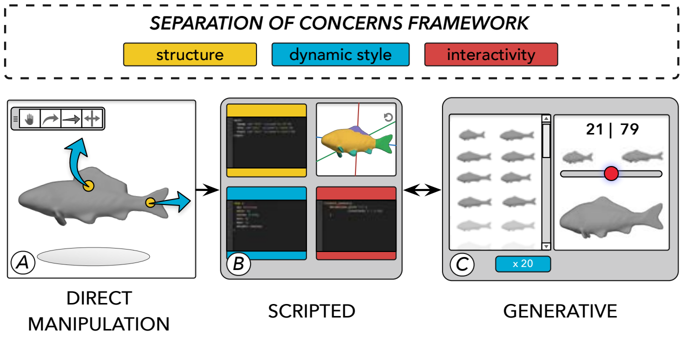


Figure 4. Makers explore form with direct manipulation (left), express dynamic styles and interactions with scripted CAD (center), and evaluate and refine dynamic models in a generative interface (right). These interfaces use a common SoC framework.

a quick iterative coding environment, the MetaMorphe scripting interface is inspired by the rapid prototyping online IDE jsFiddle[7], whereby the screen is divided into four partitions: style, structure, script, and an output mesh (Figure 4B).

We extend this pattern by adding data access widgets to use data from existing datasets or live data feeds. For our user study, we incorporated an interface to the Aeris Weather API[8], and an interface to a hardware microphone to record 2 seconds of audio. A compile button executes the appropriate script, links styles to their appropriate regions, and generates a set of parametric sliders for each style. These scripts are logged per user in a SQL database and are used to a) supplement other interfaces, and b) share code with other users.

Generative Interface

MetaMorphe also extends a user's ability to explore a design space more explicitly through a multiples view. This view divides the screen into two areas: a live view of the mesh, and a gallery of small multiples (Figure 4C). For a set of styles applied to a mesh, the framework allows us to extract a feature vector which can be used as follows:

- Applied to the mesh. This is done to generate an image and populate the gallery.
- Used to generate new feature vectors. A new set of styles is generated via the **min**, **max**, and **distribution** constraints in each respective parent style. More sophisticated design space search algorithms exist [25]; we show its feasibility under a web programming paradigm.
- Mixed with multiple feature vectors. In this interaction, a user specifies two or more parent styles and "mixes" these two models by applying a weighted sum to the parent feature vectors. In the MetaMorphe interface, we expose a slider as a method of interpolating between two models.

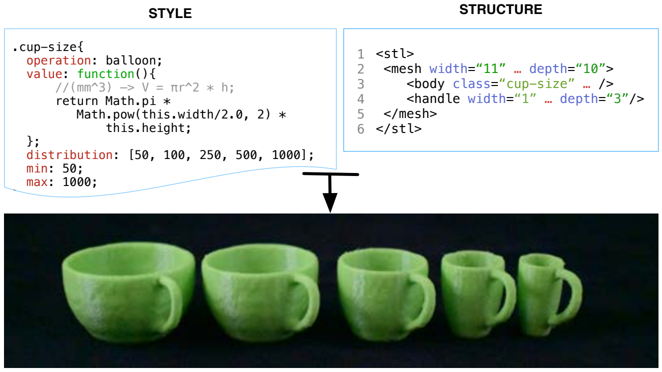
METAMORPHE DESIGN SCENARIOS

MetaMorphe leverages well established modeling principles like parametric design, but also supports a range of new interaction metaphors detailed in this section. To situate our framework we outline an evolution of several envisioned usage scenarios. These illustrate the power and novelty of how MetaMorphe facilitates creativity, sharing, and co-design across the landscape of 3D digital fabrication.

First, Patricia, a hobbyist baker, has perfected her grandmother's cookie recipe and needs to produce several batches for her new small business. Since she uses non-standard ingredient quantities, measuring has become the bottleneck. Using MetaMorphe, she decides to develop a custom measuring cup in order to streamline her process. As a starting point, she finds and loads a cup model from an online repository.

Using the *manipulation view*, she explores different deformations on the cup's body (Figure 5). She quickly realizes that applying a balloon deformation effectively changes the cup size. She switches over to the *scripted view* to add a *style*, She names this **cup-size**, selects the **value-type** of interest as the volume (milliliters), and sets the volume to a fixed **value**.

She further sees the option for specifying the **distribution** property that allows her to specify a set of allowable values apropos to her recipe. In the *generative view*, she sees the **cup-size** style expressed as five cups with differing capacity and prints the set to start her baking empire.



STYLE

```
.cup-size{
  operation: balloon;
  value: function(){
      //(mm^3) -> V = πr^2 * h;
      return Math.pi *
         Math.pow(this.width/2.0, 2) *
             this.height;
  };
  distribution: [50, 100, 250, 500, 1000];
  min: 50;
  max: 1000;
```

STRUCTURE

```
<stl>
 <mesh width="11" … depth="10">
    <body class="cup-size" … />
    <handle width="1" … depth="3"/>
 </mesh>
</stl>
```

Figure 5. In this example, a cup model is applied a ballooning deformation and parametrized to produce a set of measuring cups.

Weeks later after growing her business, she decides to revisit her design to incorporate her newly developed brand. She is inspired to use a 3D scan of her two cupped hands. She loads it into the framework and using direct manipulation selects the parts of the hand that form the "cup". Looking back at her previous design, she copies the **cup-size** style and applies it to the hand model. She then switches over to the *generative view* to see the different small multiples of the hand cup models expressed by the **cup-size** style. She then compares each model and narrows down to the one that looks the most aesthetically pleasing. She decides to submit her creation to the MetaMorphe repository for others to view and extend.

SCRIPT

```
// obj is user-submitted stl file
// returns screenshots of five generated models
var canvas = document.getElementsByTagName("canvas");
MM(obj).load(function(stl, canvas){
   stl.addStyle("cup-size")
      .generate(5);
}
```

Figure 6. A simple measuring-cup script for a ad-hoc model using the MetaMorphe framework. The script is implemented in Javascript cascade-syntax.

---

[7] `http://jsfiddle.net/`

[8] `http://www.hamweather.com/products/aeris-api/`

Later, Alex, a web developer, comes across Patricia's clever measuring cup design and views its source. She decides to make it an app, and using the MetaMorphe interactivity script writes a short "Measuring Cups" widget (Figure 6) that takes a user-submitted model, applies the **cup-size** style, and renders images of the generated measuring cups. Lastly, she adds a download mechanism for a user to print their creation.

While an end-user might accomplish the same operation with a traditional CAD modeling tool, either by scaling a cup model or making one from scratch, we claim that the MetaMorphe's framework provides a richer and more creative user experience for rapid and flexible design exploration and collaboration. The **cup-size** style can be transferred and extended to other structures while needing to minimally alter code. As designs become even more complex, we expect increased benefit from providing semantic information to create templates for more complex or custom uses.

Designing-with-data

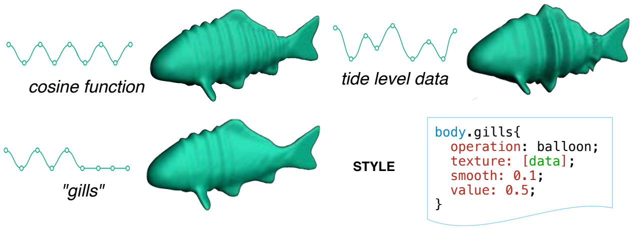
The increased control from scripting enables a novel design capability: *the ability to incorporate data as a primary design element in the creative process*. Incorporating data into artifacts has been explored as physical data visualizations [32], as physical activity artifacts [15], and as a means of creating reflective, meaningful objects [22]. However, designing with data is not only useful as a visualization, but as an exercise in understanding how structures and forms can change. The MetaMorphe framework provides methods for not only encoding data as a surface texture but also for using data to *conditionally* alter an artifact's form. The former utilizes the **texture** property in the style treatment as a heightmap and alters the surface of the mesh. The latter uses conditional Javascript programs to selectively toggle styles. We detail a few powerful examples of *designing-with-data* in practice.

Data-driven design

Medical information from sensor data is becoming increasingly accessible through wearable devices. Linking and adapting to a patient's changing medical statistics, symptoms, and anatomy can be used to influence the design of a digital model. For instance, as someone heals they may want or be allowed or encouraged to increase movement of a joint or limb as it heals. MetaMorphe could be used to easily specify a design that allow more and more flexibility in each iteration of a cast. Since each design has an associated feature representation, at scale, failed designs can be leveraged to revise existing designs based on new understanding of use cases.

Digital editioning

In another scenario, an digital artist creates a model of an object and publishes the STL file for others to use. Countless users download and 3D print this design, yet they are all identical. Limited supply is already a common practice amongst digital practitioners; a designer can choose to allow access to certain types of designs at a certain time and under certain conditions. The artist can "design in" uniqueness such that for each person, and each print, the artifact can exhibit some guaranteed uniqueness. These variations can act as a patina based on the time it was printed, the location, political conditions, or a host of other factors. These objects could serve as cryptographic encodings, or act as markers for large scale manufacturing inventory control. Figure 1b displays everyday objects (tableware) with conditional forms subject to seismograph readings tied to a specific region (2014 Napa Earthquake). John Simon's 32 by 32 pixel permutation grid *Every Icon* [29] is an example of such an object that exists in multiple instances, yet each instance embodies a unique variation with added value.



```
body.gills{
  operation: balloon;
  texture: [data];
  smooth: 0.1;
  value: 0.5;
}
```

Figure 7. Designing-with-data. The body of a fish model is applied a non-uniform balloon deformation. The weight of that deformation is controlled by a cosine function to create a texture along the body, a tapered cosine function to add the appearance of "gills", and a NOAA tide levels feed is encoded into a "live" data sculpture.

Designing a public art installation

In this example, an environmental artist has been commissioned to create an art and science public installation on tides (Figure 7). While an interactive visualization could be used to "show the numbers", the artist decides to instead create a data sculpture that can appeal to a wider audience, does not require a electronic display, and still maintains fidelity to the data. She selects a fish model and perturbs its surface structure as a function of location and tide levels retrieved from a national database. Placing these along beaches can attract visitors curiosity and provide a link for them to provide their own GPS coordinates (supplied from a browser) and view a constantly changing, unique model relative to their location. Similar to photographs, printed artifacts can serve as snapshots *linked* to certain times and places.

DNA as metaphor

Evolution as algorithm has historically been a powerful design strategy for incorporating chance and uniqueness [30], and most recently 3D objects have been evolved using biological morphologies to create unique forms [6]. Akin to keyframing in animation, we use the *DNA subvector* to mix, reproduce, and search through a design space. This allows users to essentially manipulate more than one parameter at a time; users can selectively assign different styles, and weight how much each style influences the final offspring design.

Getting the right wearable design

A hardware designer is making a data glove with wearable snap-on sensors worn on the finger tips. Their end-goal is to produce a design that feels un-obstructive, but also holds the hardware in place. The designer specifies places of the model that should not change (the hardware footprint), and specifies a few styles that change the center of mass, angle of the clasping mechanism, and the tip width. Using the *generative interface* to produce variant designs, she chooses a few possible candidates, and prints them in an array (Figure 8).

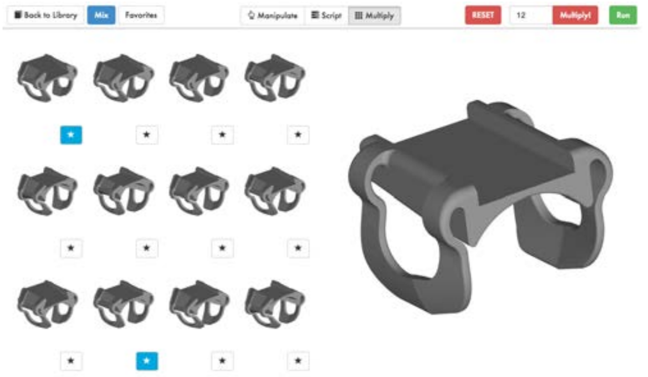


Figure 8. A high-poly finger-worn IMU distal ring design (149K faces, 7.6MB). Twelve multiples were generated and rendered in 7.2 seconds. Four styles were applied to the model: swivel and expand-out to the IMU footprint, and clasp-sides to both claspers.

Especially in the case of fitted items such as wearables, optimal designs require physical testing. By generating these multiples, the designer can use designs at the fringe, or at the extremes of certain parameters to gain critique and feedback. After testing each design, she finds that two designs which fit the criteria. She then references each design's object-DNA and reproduces an additional four "in-betweens" to refine the final model. This example highlight how generating design alternatives with current CAD tools is laborious, requiring manually altering each derivative model, and keeping separate files for each prototype. MetaMorphe enables iterating and refining designs as a part of the tool workflow.

EVALUATING METAMORPHE

We conducted a user study with creative experts from digital and traditional practices to provoke discussion on how themes in the MetaMorphe interfaces affect the design of digital models. We focus on *new design techniques and approaches* in the context of digital fabrication, rather than strictly modeling. The focus in expertise within our user study was designed to expose many of the issues faced by designers, such as learning new design tools, creatively exploring a design space, and selecting a final design. As such, our different user groups were chosen to diversify the feedback, and provide insight for how future interfaces can be designed to support multiple creative practices. Creative practices across our selected users were grouped into three categories: 3D traditional (e.g., sculpture, assemblage), 2D digital (e.g. graphic design), and 3D digital (e.g. mechanical design, 3D animation, architectural design). For the purposes of this paper, we will refer to participants in the above groups as **Sculptors (S)**, **Designers (D)**, and **CADers (C)**, respectively.

Recruitment and Selection

We submitted messages through local listservs within the Art, Architecture, and Engineering departments at our local university and advertised through *craigslist* to the surrounding community. Participants were selected on a 3-quota sampling, where each participant was asked to complete a survey. The survey consisted of three 10-point Likert scales where participants self-reported proficiency within each creative practice. Only participants with self-reported ratings above intermediate expertise (≥ 7) in at least one of the creative practices were asked to participate. Our final study group consisted of nine creative experts (3 female, 6 male) with the following years of experience: **Sculptors** (4, 16, 17), **Designers** (4, 6, 25), and **CADers** (3, 5, 29). Only two participants report 3D printing experience (**C1**, **C2**).

Study Design

For each session, participants were asked to individually meet with us in our digital fabrication studio. Participants were paid $20/hr; each session lasted two hours and consisted of a warm-up tutorial, three design tasks, and lastly a card sort. We also conducted interviews before and after each session. Participants were also asked to reflect out-loud their reflections on tools and design process as they went through the workshop. The experimenter aided solely on interface issues.

Each participant engaged in three separate modeling tasks using any of the three MetaMorphe interfaces. The modeling tasks required participants to search for an existing STL model from the Princeton library and engage with the model as follows: a) a *technical* task: add styles to a model to describe multiple forms, d) a *transfer* task: design a style that can be applied over a class of models, and lastly c) a *context* task: design a model intended for someone else. We instructed participants that one of the digital models made would be 3D printed for them to keep. Due to long 3D printing times, objects created during this study were later fabricated and mailed to users; participants were interviewed several days after receipt of their artifacts.

Findings

Each participant completed the three design tasks and engaged with the "designing with data" paradigm through either of the two provided feeds of weather or voice. Five participants chose to have their context task model printed, while four chose to have the transfer task model printed (Figure 9).

MetaMorphe as Creative Tool

Several novel design trends emerged from using MetaMorphe: *Exaggerated Anthropomorphism* where we observed several individuals adding emotion to animal and human models by exaggerating or anthropomorphizing appendages such as the case of a *muscle* class for an action figure (**C2**) or a pig caricature designed to the likeliness of **C3**'s landlord. *Remixing* where metaphors were remixed within a single object such as **D1**'s hip sunglasses applied a rippled effect from Los Angeles weather patterns on the lens (Figure 9A) or **D3**'s beer glass encoding of Tokyo's weather data, designed for her sister. Her inspiration, she reflected, was that the hot temperatures in Tokyo called for a shared beer (Figure 9D).

We also observed direct *Nostalgic and Memento Mappings* using voice or sound to enhance visual form and personalize models such as **S3**'s "BeatTable" that encoded one of his experimental music compositions or **S2**'s homage and remembrance of Masaru Emoto's writings and water experiments through a conceptual diptych of two simple forms perturbed by "positive" and "negative" sounds (Figure 9C). When searching for a context task model, **C1** encountered a donkey which reminded her of a humorous shared memory

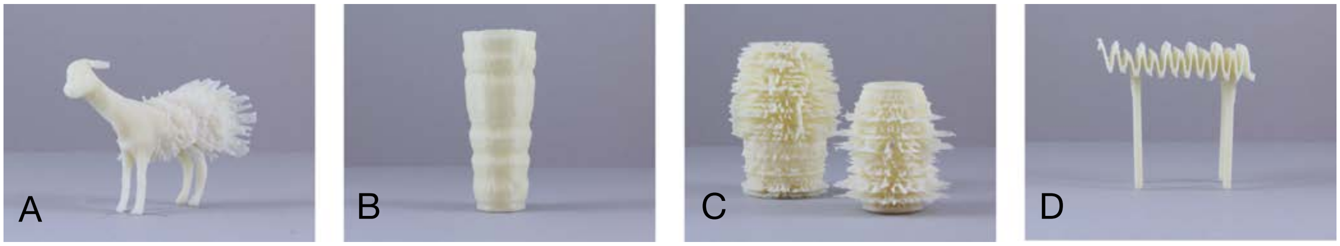


Figure 9. Models 3D printed from the user workshop. A) *Iron burro* - a nostalgic shared memory is encoded on the body of a donkey as sound (C1), B) Tokyo weather data is mapped onto the side of a glass (D3), C) Masaru Emoto-inspired positive and negative forms (S2), D) flexible table structure made from convoluted weather and tide data (S3).

between her and her father; in her piece christened *Iron Burro*, she encoded a soundbite of that personal story onto the torso of the donkey (Figure 9A). Finally, many of the designs explored *Improved Functionality* over form such as a *grip* class style designed as a non-slip texture for tool handles. However, the designer also conceptualized the texture as a means of artistically encoding an artist signature (**S3**).

Many of these designs would be extremely difficult if not impossible to achieve using traditional 3D modeling tools. However, many of our participants readily engaged with these creative and expressive new forms using MetaMorphe.

MetaMorphe as Computational Literacy Tool

Making system mechanisms transparent through a visual interface promoted learning and computational literacy. Amongst study participants, three were regularly coders (**C1**, **C3**, **D1**) while several described failed or continuing attempts at picking up programming (**D2**, **S1**, **S3**). Although not a requirement for participation (and not elicited) all participants revealed familiarity with the HTML/CSS markup style. While initially apprehensive about the scripting environment, it became a central point of interaction for these designers. During the transfer style task, several participants stated that they experienced a learning moment. Furthermore, participants expressed having a multiplicity of views and realtime visual feedback enhanced their experience and helped them understand the underlying mechanism of the system and adjust their conceptual model:

> **D2:** The image and the code help people see something physical and see how [the code] changes these values... to see that this [code] is equated with this [model] when people play with it. I see it as a way to understand computer science, coding, and language by breaking it down based on visual exploration.

This corroborates the concept of a Web framework as a platform for learning, as users are able to establish a programming metaphor through practice as they iterate on their work and re-enforces the value of the view source function. This is an early indication that MetaMorphe affords scaffolding and learning within 3D design and perhaps DF design literacy.

MetaMorphe as Facilitator of the Design Process

Sites of exploration evolved as users became familiar with the MetaMorphe framework. In general, initial frustration of specifying a style in the *scripting view* led participants to concentrate most of the exploratory design process in the *manipulation view*. However, once interface mechanisms were understood, participants found that they enjoyed the control from the *scripting view* and shifted the majority of their design time to this view. In fact, designers desired more control to lock parts of a design as they drilled into more fine grain manipulations (**D1**, **D2**, **C3**).

We noted that more experienced participants incorporated a bidirectional flow (**C3**), returning to the *manipulation view* and iterating regularly with the small multiples in the *generative view*. Several **Designers** recognized the role this software process had in defining their workflow, such as *layering* in Adobe Photoshop (**D2**). While not originally part of their design process, the software-defined process gradually began to mirror their analog design processes:

> **S2:** [MetaMorphe's interfaces] mirrored my creative process enough that I could easily start following into its parameters. It's slightly different from menu selects, and once you start getting used to them then you understand the process it has to go through, you start adapting your own style to it also.

Due to the range of creative design processes that each user follows, creating a single pipeline interface that adapts to a user's design process requires a much more custom solution. These findings suggest that introducing bi-directional sites of exploration allows for a more iterative design process.

MetaMorphe as Co-Designer

General perceptions of the generative interface were often compared to perceptions of coding. Ideas of control similarly manifested as a user's affinity to the scripting interface. For those with more programming experience, the introducing non-deterministic actions was linked to a lack of agency:

> **D1:** I am not someone who lets random handle it. I want to figure it out on my own, I do not know if [the generative interface] is for me . . . in general I go for one thing.

In contrast, those with less affinity to the scripting interface embraced randomness as a resource, like:

> **D3:** It's always a challenge when you are working on a piece and get stuck. I get frustrated. This [generative interface] opens up a doorway to continue pushing you forward.

Straight-forward manipulation of parameters provided little agency to users. In general, the act of exploring or messing around with a design was lackluster, whereas adding in sounds or weather data made it easier to find an intention that formed a design (**C1**). For others, the expressibility of these interactions were surprising, random, and interesting,

> **S2:** Where as this is just modifying an alteration, for those which used my sound input, I created new dynamic forms and shapes

> that don't resemble the original object. Through voice, through sound, tone, modulation, and volume, I had control of the final form. Those designs belong to me.

Perceptions of authorship encountered during the study can be divided into three general areas: 1) perception that the underlying model supersedes any artificial or superficial alteration, 2) that manipulating existing models was really a partnership and that work was a factor of a collaborative effort between the original creator of the current user, 3) and lastly that the end work was sufficiently different in context of concept, form, and as a factor of the type of data which is embedded. Notably, seven of the nine final produced models contained personal audio data or personal geographical data. Participants reported that these works were the most memorable of the process and those they felt they had contributed more to the design.

Printed MetaMorphe Artifacts as Nacent

Each participant was interviewed several days after receiving their digital model. When asked on the current location and whereabouts, two participant reported using them as functional objects (e.g. as a flower pot [**C2**], small pedestal [**S3**]). The other participants reported a more memento-type placement on a desk or workspace [**S3**, **S3**], and as a gift [**C1**]. Notably, the act of physically printing the object altered previous perceptions of authorship. Designs were viewed as being entirely the participant's (**D2**, **D3**). Future explorations of 3D printing were all centered on creating bespoke objects:

> **D3**: Symmetry is easy to make, and assymetry is accidental and more interesting. I want to make it more accidental. The Tokyo weather added some of this accidental quality.

The plastic material quality of the objects still caused many to classify them as "knick-knacks"; however perceptions of objects with subjective data is captured by **C1**: It is a memento but its much more than a souvenir, its real life!.

DISCUSSION

As digital fabrication continues to develop, dynamic models will become a central tenant of design. Our findings suggest that MetaMorphe enables designers to reflect on how objects can embed different histories and futures, and engage designers with renewed agency.

In his seminal 1936 text, *The Work of Art in the Age of Mechanical Reproduction*, Walter Benjamin describes how perceptions of objects changed under mechanical reproduction [2]. Among these, Benjamin identifies a ritualistic value around unique artifacts that he terms "aura" and suggests that mass manufacturing diminished this uniqueness. MetaMorphe attempts to introduce new notions of "aura" into digitally fabricated objects. Figure 9 illustrates the type of crafting around personalized artifacts that resulted from designing-with-data. Tools that integrate subjective data into digitally fabricated artifacts add new personality and meaning to objects. This has implications for sustainable design since such legacy-containing artifacts can persist as heirlooms.

In addition, increased meaning and agency in design can add an important element to STEM education. While creating highly-personal objects such as named lasercut keychains has been shown to encourage making-enthusiasm amongst young learners, these fabrication technologies become afflicted with "keychain syndrome", or the tendency to use these technologies to mass-produce trinkets instead of engendering invention [3]. More dynamic designs can allow more visibility into the process of designing and fabricating 3-D models. For instance, MetaMorphe presents both a parametric slider and its associated dynamic style, exposing the design decision and process of the original creator. While MetaMorphe specifically manipulates existing designs as opposed to starting from scratch, our user study demonstrates that many participants felt a strong connection with artifacts that embody relevant and subjective data. This increased link to authorship can be an important driving force in motivating invention.

LIMITATIONS AND FUTURE DIRECTIONS

Our objective with MetaMorphe was to find alternative ways of programming digital models that encourage metamorphic design. As such MetaMorphe is not a full-featured CAD tool; however MetaMorphe can incorporate more general purpose parametric CAD operations (e.g. tessellation, shelling, chamfering), yet non-parametric operations (e.g., brush interactions) do not have as simple an interface. Incorporating specialized markups side-by-side is a potential method for providing a heterogeneous modeling environment.

Conveying the mechanical and functional properties of a digital model was a limiting factor to how participants conceptualized which designs were modifiable. Many participants expressed a desire to specify real-world values (e.g., mm, inches) [**C1**, **C3**], or place the design in context (e.g., with a backdrop) [**C2**, **D3**]. Furthermore, the plastic material caused many to perceive printed artifacts as kitsch. Currently, MetaMorphe supports surface interactions, however designing dynamic models is more than just changing form. Under the mesh skin, structural properties like softness, flexibility, and the ability to leave impressions (e.g., leather can enhance the materiality of an artifact. As DF technologies mature integrating knowledge of material behaviors and properties in design tools can increase the diversity of designs.

CONCLUSION

Digital fabrication can promote a new class of designs that are more personal, sustainable, and dynamic. MetaMorphe provides a mechanism for enabling these behaviors in a seamless way when engaging with repositories of existing digital models. We leverage web programming metaphors to facilitate a scripted CAD and open-sharing design practice. We show that this approach supports generative, programmatic, and direct manipulation interaction styles. Participants who used MetaMorphe were able to easily explore a broad design space and create individual artifacts that embodied personal reflection, material engagement, and expressive forms.

ACKNOWLEDGMENTS

We thank Jasper O'Leary and Akhila Raju for their aid with user studies; Tim Campbell, Valkyrie Savage, Björn Hartmann, and the anonymous reviewers for their insightful comments. This research was supported by NSF Grant No. IIS-1451465, Adobe GEM, and a NSF GR Fellowship.

REFERENCES

1. Autodesk. Project Dreamcatcher, 2015.
2. Benjamin, W. The Work of Art in the Age of Mechanical Reproduction. *Visual Culture: The Reader* (1936).
3. Blikstein, P. Digital fabrication and making in education: The democratization of invention. *FabLabs: Of Machines, Makers and Inventors* (2013), 1–21.
4. Buechley, L., Rosner, D. K., Paulos, E., and Williams, A. DIY for CHI: methods, communities, and values of reuse and customization. In *CHI EA*, ACM (2009), 4823–4826.
5. Chen, X., Golovinskiy, A., and Funkhouser, T. A benchmark for 3D mesh segmentation. In *Trans. on Graphics*, vol. 28, ACM (2009).
6. Clune, J., and Lipson, H. Evolving 3d objects with a generative encoding inspired by developmental biology. *ACM SIGEVOlution 5*, 4 (2011), 2–12.
7. Follmer, S., Carr, D., Lovell, E., and Ishii, H. CopyCAD: remixing physical objects with copy and paste from the real world. In *Proc. UIST*, ACM (2010).
8. Follmer, S., and Ishii, H. Kidcad: Digitally remixing toys through tangible tools. In *Proc. CHI*, ACM (2012), 2401–2410.
9. Hiller, J. D., and Lipson, H. STL 2.0: a proposal for a universal multi-material Additive Manufacturing File format. In *Proc. Solid Freeform Fabrication Symposium'09* (2009), 266–278.
10. Hutchins, E., Hollan, J., and Norman, D. Direct manipulation interfaces. *Human-Computer Interaction 1*, 4 (Dec. 1985), 311–338.
11. Jacobs, J., and Buechley, L. Codeable objects: computational design and digital fabrication for novice programmers. In *Proc. CHI* (2013), 1589–1598.
12. Jacobs, J., and Zoran, A. Hybrid practice in the kalahari: Design collaboration through digital tools and hunter gatherer craft. In *Proc. CHI '15*, ACM Press (2015).
13. Kalogerakis, E., Hertzmann, A., and Singh, K. Learning 3D mesh segmentation and labeling. *Trans. on Graphics 29*, 4 (2010).
14. Kho, Y., and Garland, M. Sketching mesh deformations. In *ACM SIGGRAPH 2007 courses* (2007).
15. Khot, R. A., Hjorth, L., and Mueller, F. F. Understanding physical activity through 3d printed material artifacts. In *Proc. CHI '14*, ACM Press (2014), 3835–3844.
16. Kintel, M., and Wolf, C. OpenSCAD, 2011.
17. Kuznetsov, S., and Paulos, E. *Rise of the Expert Amateur: DIY Projects, Communities, and Cultures*. ACM, 2010.
18. LIA. Filament sculptures (2014).
19. McCullough, M. *Abstracting Craft : The Practice Digital Hand*. The MIT Press, July 1998.
20. Mota, C. The rise of personal fabrication. In *Proc. C&C*, ACM (2011), 279–288.
21. Mueller, R., Nieuwnhuijse, J., Bespalov, E., and Hogdson, G. OpenJSCAD, 2013.
22. Nissen, B., and Bowers, J. Data-Things: Digital Fabrication Situated within Participatory Data Translation Activities. In *Proc. of CHI '15*, ACM Press (2015), 2467–2476.
23. Oehlberg, L., Willett, W., and Mackay, W. E. Patterns of Physical Design Remixing in Online Maker Communities. In *Proc. of CHI '15*, ACM Press (2015), 639–648.
24. O'Reilly, T., DiBona, C., Stone, M., and Cooper, D. Open source paradigm shift, 2004.
25. Ovsjanikov, M., Li, W., Guibas, L., and Mitra, N. J. Exploration of continuous variability in collections of 3d shapes. In *Trans. on Graphics*, vol. 30, ACM (2011), 33.
26. Plummer-Fernandez, M. Disarming Corruptor, 2013.
27. Rosner, D. K., and Ryokai, K. Reflections on craft: probing the creative process of everyday knitters. In *Proc. C&C*, ACM (2009), 195–204.
28. Sadouma, T. Nike town 2, Jan. 2015.
29. Simon, Jr., J. Every Icon, 1996.
30. Sims, K. Evolving virtual creatures. In *Proc. Computer Graphics and Interactive Techniques* (1994).
31. Song, H., Guimbretire, F., Hu, C., and Lipson, H. ModelCraft: capturing freehand annotations and edits on physical 3d models. In *Proc. UIST*, ACM (2006), 13–22.
32. Swaminathan, S., Shi, C., Jansen, Y., Dragicevic, P., Oehlberg, L. A., and Fekete, J.-D. Supporting the design and fabrication of physical visualizations. In *Proc. CHI '14*, ACM Press (2014), 3845–3854.
33. van den Berg, D. DERRICK, Oct. 2014.
34. Vidime, K., Wang, S.-P., Ragan-Kelley, J., and Matusik, W. OpenFab: a programmable pipeline for multi-material fabrication. *Trans. on Graphics 32*, 4 (July 2013), 1.
35. Weichel, C., Lau, M., Kim, D., Villar, N., and Gellersen, H. W. MixFab: a mixed-reality environment for personal fabrication. In *Proc. of CHI*, ACM Press (2014), 3855–3864.
36. Willis, K. D., Xu, C., Wu, K.-J., Levin, G., and Gross, M. D. Interactive fabrication: new interfaces for digital fabrication. In *Proc. TEI* (2011).
37. Zhao, J., and Moere, A. V. Embodiment in data sculpture: a model of the physical visualization of information. In *Proc. DIMEA*, ACM (2008), 343–350.
38. Zoran, A., and Buechley, L. Hybrid reassemblage: An exploration of craft, digital fabrication and artifact uniqueness. *Leonardo 46*, 1 (Feb. 2013), 4–10.
39. Zoran, A., and Paradiso, J. A. FreeD: A Freehand Digital Sculpting Tool. In *Proc. CHI*, ACM (2013), 2613–2616.