Compositional 3D Printing: Expanding & Supporting Workflows Towards Continuous Fabrication

Jeeeun Kim
University of Colorado Boulder

Clement Zheng
University of Colorado Boulder

Haruki Takahashi
Meiji University

Mark D Gross
University of Colorado Boulder

Daniel Ashbrook
University of Copenhagen

Tom Yeh
University of Colorado Boulder

ABSTRACT

We present Compositional 3D Printing, recasting the 3D printer as a tool for expression that responds to real-time design decisions, analogous to composing a piece of music using a mixer. Our paradigm supports a wide range of inputs and interactions for designers, to be used in the moment; not only before printing, but anytime during production. We propose the design space of this digital fabrication paradigm, and outline methods and technical details with which researchers and practitioners can expand this space.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**;

KEYWORDS

Personal Fabrication, 3D printing, Creativity Support

ACM Reference format:

Jeeeun Kim, Clement Zheng, Haruki Takahashi, Mark D Gross, Daniel Ashbrook, and Tom Yeh. 2018.: Expanding & Supporting Workflows Towards Continuous Fabrication. In *Proceedings of SCF'18: Symposium on Computational Fabrication, Cambridge, MA, USA, June 17-19, 2018 (SCF'18)*, 10 pages.
DOI: https://doi.org/10.1145/3213512.3213518

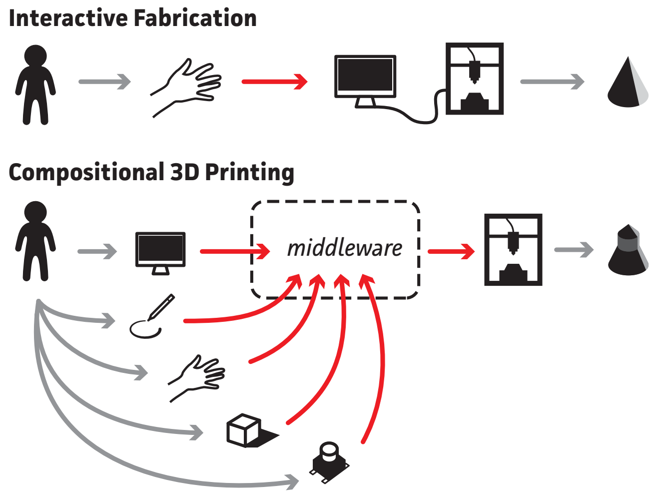

Figure 1: Compositional 3D printing (bottom) compared to current interactive fabrication concept (top).

1 INTRODUCTION

The emerging field of personal fabrication leverages computational power to enable human/machine collaboration for creative endeavors. Advances over the last decade have made this technology accessible to a wide audience, from hobbyist makers to professional designers. It provides a medium for a myriad of expressions, from engineered mechanical parts to artistic and playful sculptures.

Yet, digital fabrication today is reminiscent of batch processing in the early days of computing [Baudisch and Mueller 2017]. Code is first composed by a programmer using one type of interface. It is then submitted for execution, and a user loses control over the code until processing is complete. The current digital fabrication paradigm echoes this mainframe computing model—where computer-aided design (CAD) files are loaded into a fabricator. Here, *design* and *production* are strictly separate. Input is often a single file that is entered to the machine. Aside from starting, pausing and stopping jobs, machines do not permit any interaction, such as real-time edits that a user might want to make. Furthermore, these CAD inputs are often created in software packages using conventional graphical user interfaces, leaving no room for other embodied processes in creative work such as free-hand sketching and manipulating physical tools and objects.

We are inspired by existing creative processes that involve computational power—that of composing music with a mixer. Through a mixer, a composer interacts continuously with the device to layer and modify sounds. The composer has a rich set of bodily vocabularies to express musical ideas, using various instruments. The music mixer takes an initial input and constructs a base sound. A composer adds and edits the beats, pitch and rhythm on-the-fly.

We present Compositional 3D Printing, recasting a creative process to enable users to "compose" a model via real-time design decisions and expressions while interacting with a 3D printer directly (Figure 1). This shifts 3D printing away

SCF '18, June 17–19, 2018, Cambridge, MA, USA

ACM ISBN 978-1-4503-5854-5/18/06...$15.00
https://doi.org/10.1145/3213512.3213518

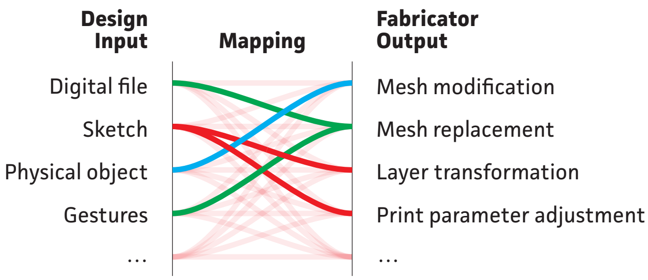

Figure 2: Mapping and Expanding the design space of Compositional 3D Printing. We support various mappings between current/future design inputs to the number of design outputs during fabrication process

from a print-and-forget paradigm. Compositional 3D printing shares two key principles with music composition— **continuous interaction** and **multiple vocabularies**.

Our contributions are threefold:

- the Compositional 3D Printing which supports continuous interaction and multiple design expression
- six workflows that show the design space, how different user groups compose a 3D model on-the-fly,
- technical implementation of an open-source middleware to support the Compositional 3D Printing and to help expand the design space.

2 COMPOSITIONAL 3D PRINTING

With the democratization of digital fabrication tools for *personal* fabrication, a diverse spectrum of people approach 3D printing with a wide range of design literacies depending on their domain knowledge and goals— from hobby projects to professional product design. Design professionals can employ the plethora of CAD programs and digital fabrication processes available to model and fabricate their designs; but their creative process may also include other modes such as sketching and model making by hand. On the other hand, design novices often struggle with the steep learning curve that professional CAD programs impose, and may better express their ideas through hand sketches and craft materials, which an intelligent system can translate into a 3D model [Mueller et al. 2012; Saul et al. 2011]. We also observe amateur makers create 3D models with different expressive needs. For the same project however, she may also explore the aesthetics of different surface patterns.

Researchers have proposed *interactive fabrication* to account for this diverse relationship between users of digital fabrication. In these approaches, users to directly manipulate an object in real time through embodied interactions during physical fabrication (Figure 1). Compositional 3D Printing does not propose to completely replace previous efforts towards bi-directional interactions, but to *expand* their respective design spaces, so the design can include various design literacies and expressions. Furthermore, we harness the machine's affordances for richer creative output.

Apart from supporting a wide array of design literacies, digital fabrication tools should also facilitate the iterative design alteration while the production is operating. Schön describes designing as a conversation between reflection and action; and reflection can take place as a response to a design action (***on*** action), or during the action itself (***in*** action) [Schön 1992]. Currently, a design iteration employing 3D printers is only as granular as the complete design to fabrication cycle. We want to support users to effect design changes during the 3D printing process as well, facilitating reflection and action at any time during the planning to fabrication process.

Compositional 3D Printing supports designers to engage with a FDM 3D printer through multiple input modalities, as well as implement design decisions and actions even as fabrication is happening. To demonstrate this, we developed and implemented a system that functions as middleware between design input and fabricator output. This middleware supports associating inputs with different fabrication outputs based on a user's intent (Figure 2), while managing synchronous communication with the 3D printer to effect changes (Figure 3). Also, it provides channels that future interaction designers and developers can use to create new input modes for digital fabrication systems. In subsequent sections, we will elaborate on the principles , as well as demonstrate it through a variety of design workflows and supporting techniques.

3 RELATED WORK

Compositional 3D Printing is inspired by tools to aid computational crafting that allow interactive information exchange between a user and a computer, and recent advances in interactive fabrication that work within these paradigms.

3.1 Computational Craft

In a traditional craft process, the designer's input appears directly as a physical output, allowing designers to think about the next step with immediate feedback. Recent computational tools have explored this interactive nature of this process, reinforcing an inter-relation with the materials and tools for crafting [Oh et al. 2017]. Parameters exposed to end users provide an explorative space, to test features in the digital system and observe emerging outcomes[Johnson et al. 2009] Craft is the process of making with existing objects and materials, as MixFab invites users to bring existing materials to the digital design space [Weichel et al. 2014]. From hobby creation to fine products, these tools lead the making process to benefit from the precision of computation when the design activities of the user are imperfectly precise.

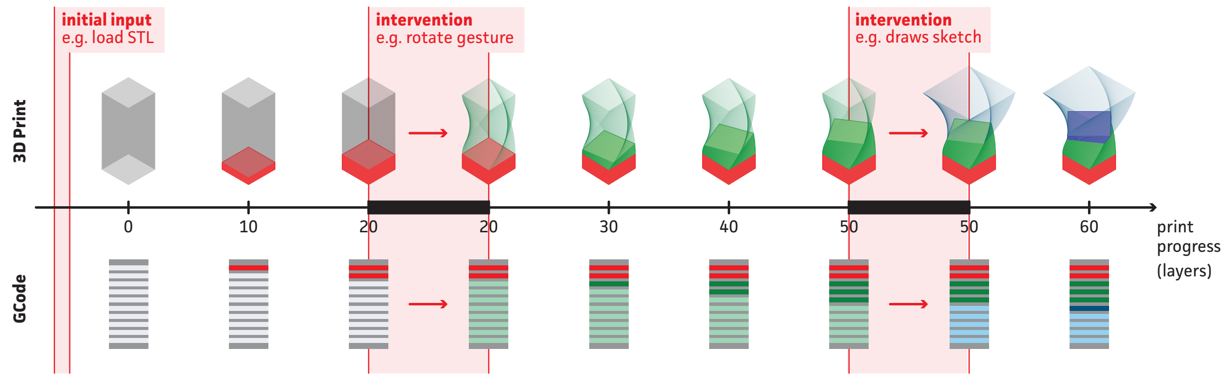

Figure 3: Compositional 3D Printing intakes multiple inputs during the process by segmenting Gcodes in two parts. It accepts just-in-time design inputs to apply changes on the model currently being created, returns physical feedback to a designer, providing an opportunity to test their design actions presented by various literacies

3.2 Interactive Computing

Interactive computing inspired the computational tools to support crafting and design, with the continuous interaction between user and computer [Igarashi 2010]. For example, Plushie and Beady construct a 3D simulation model in digital space, based on a user's 2D sketch input [Igarashi et al. 2011; Mori and Igarashi 2007]. A user draws a crude shape of a stuffed toy, then the computer generates a 3D model. Similar to a physical crafting process, the system enables a user to think about one feature at a time, seeing the instant outcome generated by the computer.

However, the design space of these practices is limited to the screen. The interaction happens via a 2D representation of a 3D model. It disconnects machine parameters from the process, to be used as design sources.

3.3 Interactive Fabrication

Today's interactive fabrication takes into account the relation between the human and the tools, inspired from traditional crafting tools. The goal is not to leave production solely to machines, but to facilitate the participation of users in the entire pipeline. There have been many attempts to make the personal fabrication design task interactive.

Constructables [Mueller et al. 2012] involves end users in directly manipulating the workpiece, observing the instant result created by the machine. This process synchronizes design input and output, by the user and the machine taking turns. As envisioned in Interactive Fabrication [Willis et al. 2011], improvised physical design input influences the work-in-progress. In addition, D-coil and FreeD add intelligence to a machine, which assists designers actions with haptic feedback during production according to the original design input [Peng et al. 2015; Zoran and Paradiso 2013]. In a similar bi-directional information flow, ReForm allows a user to update physical models, propagating changes to the digital model and vice versa [Weichel et al. 2015]. On-the-fly fabrication [Peng et al. 2016] helps a designer update the model with her intention during design production. A designer can focus on one feature at a time, first creating the rough body shape of a pot, then moving to the next step, handle and spout.

While existing frameworks establish the basics of continuous fabrication to update the model along with users' in-situ design decisions, design interactions are often limited to screen based CAD tools, embodied interactions are one or set to a few specific types per machine. It is difficult to extend a specific workflow suggested in one technique to a different type of machine, for example, applying wire printing on-the-fly to vinyl cutting.

4 DESIGN SPACE

4.1 Guiding Principles

Compositional 3D Printing is established upon two core principles: enabling continuous interactions and supporting multiple modeling design styles that raise design considerations for future design of personal fabrication.

4.1.1 Continuous interaction. A user of Compositional 3D Printing must be able to keep refining a model until production finishes, rather than simply waiting for her digital model turns into the physical form. A designer must have a better understanding of process and so may have more control. She should be able to express how to modify the model on-the-fly to explore effects, whenever she desires. In this process, the information should flow bi-directionally, not only from user to printer, but also from printer to user. It facilitates turn-taking between the machine and human; the user expressing inputs via interventions (as demonstrated in the red zones in Figure 3) and the machine accepting them that effect changes and modifying its behavior accordingly (illustrated as G-code transitions). The machine should listen to any real time event that might occur at any time, so the production task may be changed by the user's arbitrary interventions.

4.1.2 Multiple Design Vocabularies beyond Boundaries of Modeling Tools. Compositional 3D Printing encompasses a wide array of interaction styles to compose 3D objects, not limited to loading an STL file. It encourages the use of a variety of embodied "design languages" to express a rich set of design intent that can influence different stage of design/production. Model composers can use body language such as gestures, a pictorial language such as sketching, or a tactile language such as showing a physical object to create a digital model, directly change a machine's toolpath or printing parameters. These embodied interactions can be associated with many different meanings at different stages of making. Richer vocabularies allow users to more flexibly convey their creative intent using the best-suited actions to convey it, regardless of their different literacy skills that are understood by a specific modeling algorithm which may be far from human's design language.

4.2 Workflows

In this section, we describe design workflows incorporating interventions in the fabrication process as part of many creative endeavors.

4.2.1 Motivating Scenario. One autumn day, a user decides to make a gift for the guests of her birthday party. She wanted a gift that would commemorate the season. She brought a fallen leaf so that her 3D printer could extract its shape, and she used a pen to sketch a contour line with details of the leaf. The printer extruded the shape vertically for about 1 mm. As she wanted to build a small desk organizer, she needed to print thin borders around the shape. So she paused printing to renew her model, then drew an outline with a thick stroke on top of the model. While the borders were being printed, she changed her mind and decided instead to create a chopstick rest that guests could use for the dinner and then take home as a parting gift. She again paused the machine, drew a zigzag line in mid-air using her index finger inside the space to instruct the 3D printer to print loose scaffolding, to support the upper layer of the design. She then let it replicate the shape printed on the bottom and printed several more layers on top as a cover. She placed real chopsticks to visually validate the height where they rest. At the right time in the production process, she drew a concave line in mid-air to cut out the original shape. Finally she saved the result for reproduction.

4.2.2 Augmentation Using Real-world Objects. Augmenting designs to real-world objects is a common practice in digital fabrication [Chen et al. 2016; Guo et al. 2017]. Measuring the objects' physical dimensions and entering them into a digital model is non-trivial task in this practice. Although many online open-source repositories provide off-the-shelf models that can be customized by modifying parameters, measuring to get a 'right' model can be error-prone process for novices. [Kim et al. 2017]. Novice makers may be inexperienced with customization, often iterating several times to get an artifact to fit correctly. Our system eliminates this tedious process of measuring and digitizing values, prototyping, and iterating.

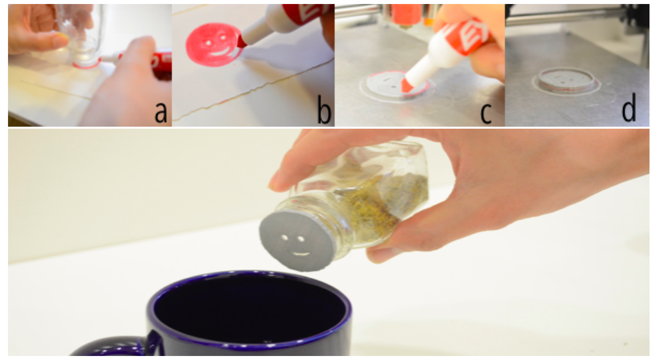

Figure 4: (top) Process of making a jar lid, by drawing the outline of a real-world object using surface filling to facilitate expressions. (bottom) The design results in a tightly fitting shaker lid to spread lemon pepper on the soup in a mug.

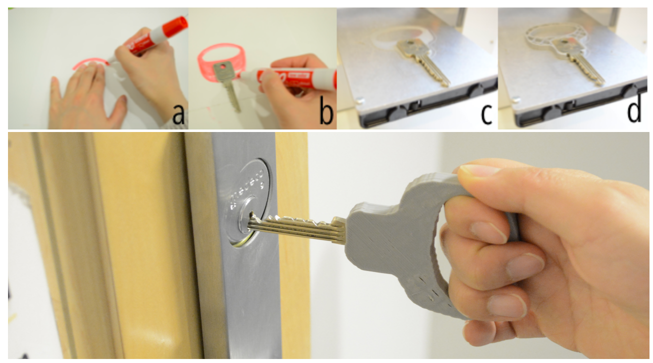

Figure 5: (top) Process of design an assistive key turner, without measurement of an existing object. (bottom) It results in an augmenting key to create a large adapter

To make a shaker lid with unique holes (See Figure 4), a user first follows the contour line of a jar. Without measuring, the user gets an exact physical dimension by following the physical outline (a). She fills the surface with color, leaving white spaces for holes (b) with her unique drawing. She then places the sketch on the printing base for the 3D printer to recognize the shape. When the printer completes several layers, the user realizes that the jar needs a lip for mounting, so she stops printing to draw an outline of the shape (c). The printer then prints walls around the lid, based on the new drawing, to form this new lip (d). The user then stops printing when she is satisfied with the shaker lid.

Similarly, to design an assistive key turner for children and people with fine motor impairments, the user starts by creating a sketch of an outline of three fingers (Figure 5a). Then she places a real-world object, a door key, to measure its size (b). The 3D printer recognizes the sketch placed on top of the build plate and prints the first few layers. While the key turner is being printed, the user pauses printing and places the key on top of the model to indicate that she wants to modify the model; creating an empty slot. The printer recognizes the shape in relationship with the holder (c). After printing additional layers up to the height of the object with slots (d), the user can embed the object into this space. Then the printer completes the rest, securing the inserted object in place.

4.2.3 Handcrafted Prototype to 3D Model. Cardboard prototyping is one of the most common methods for designing physical objects [Oh et al. 2017]. In this craft-oriented design process, a user quickly builds a prototype and allows improvisation and consideration of physical factors using flexible materials and tools. For example, a user can test force and gravity on a phone stand, using everyday materials such as masking tape and a knife within a short amount of time. In this workflow, a user wants to make a stand to hold her phone at a comfortable angle. She finds the right size and angle by experimenting, fixing the design using masking tape (Figure 6 a & b). She marks the shape with a red line and places it on the build plate (c). She then orients her physical prototype, so that the system can recognize the shape from the top view. The system then creates an STL model from the physical prototype (d).

4.2.4 Embedding Materiality in Off-the-shelf Models. Remixing and combining diverse materials are common approaches to designing a product [Denzin and Giardina 2016]. In many cases, users start making things with off-the-shelf models downloaded from online repositories. However, modification and customization of the model depend on the original designer's predefined parameters.

The Mechamagnet project [Zheng and Do 2018] is an approach to help industrial designers create tangible interactions by inserting magnets into objects at various points. The designer may want to test form factors with a 3D printer as a physical prototyping tool: for example, finding the distance between magnets that provides the most effective force feedback. With our system, the designer can load an initial STL file found from a repository for printing and start printing (a). When she finds the position to insert a magnet, she pauses and places real magnets to create slots. Strong magnets can attract each other if placed too close together, so a designer indicates positions of remaining the magnet placements by annotations (b). The designer then waits for the printer to print a few more layers before automatically pausing, indicating to the designer to physically insert the magnets in place. She then follows the required tasks, inserting the magnets (c), and resumes printing.

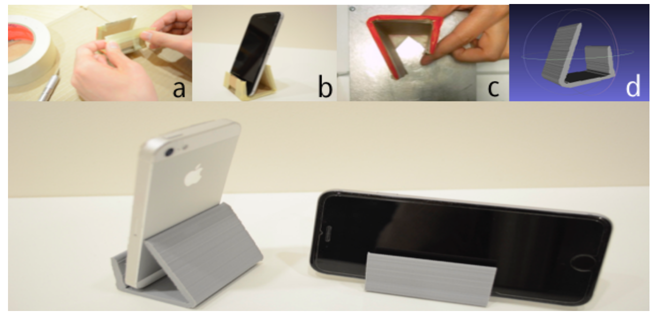

Figure 6: (top) Creating a 3D object from a handcrafted physical prototype. (bottom) The phone stand is first created using a cardboard, to test the physical attributes.

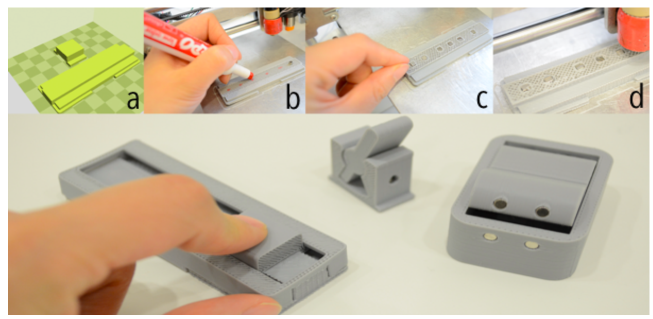

Figure 7: (top) Inserting a physical object. Starting with an off-the-shelf slider, a user places a foreign object to modify the original geometry. (bottom) One example could be the experimental creation of a Mechamagnet.

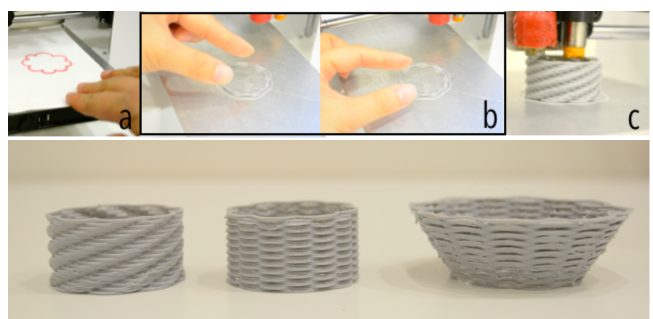

Figure 8: (top) 3D sculpting with interventions. (bottom) Drawing and hand gestures (rotation using two fingers) allow a designer to experiment with various texture effects of weaving.

4.2.5 Pattern Definition and Repeat. Experimenting with a 3D printer's printing parameters has been applied in many artistic practices [LIA 2014; Rael 2016]. By nature, the physical creation of an artifact relies on environmental factors, such as gravity or the speed at which material solidifies at room temperature. Designers may explore physicalities and artifacts created by these factors as input sources.

In this example, a designer draws a cylindrical outline with bumps as a primitive shape with which to initiate printing for testing variations that she might want as basketry textures

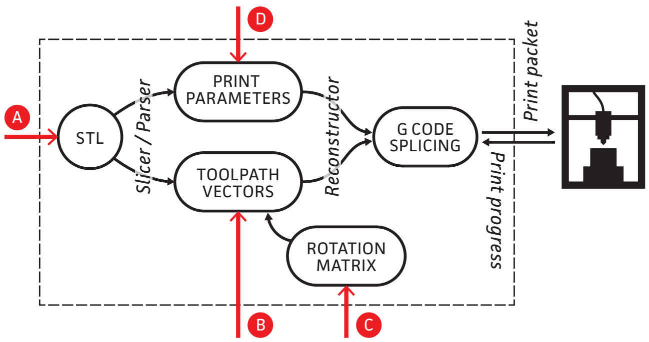

Figure 9: A multiple entries during the production of 3D model to harvest the in-situ design decisions that define changes on-the-fly.

(Figure 8). As a ceramic designer, she wants to explore how material moves and solidify by physical factors, and know which will fit her taste, among many potential variations. Thus, when the printer has completed the first few layers, she rotates the layer using finger gestures (b). The previewer shows what the rest of the prints will look like. With the user's confirmation, the printer prints the same layer patterns but with a slight modification; with the rotation in xy-plane. When the printer completes a few more layers, she claps to define the pattern from the history, (print three layers, and rotate at a certain angle) to repeat (c).

5 DESIGN INPUTS, OUTPUTS, AND MAPPING

Figure 9 demonstrates how the system caters multiple channels (Figure 9 A-D) to a user with which to change the model using various interaction vocabularies. Internally, the system provides communication channels with real-time design inputs and translates them to factors that influence the various granularities of a model (e.g., into meshes, 2D layers, or machine parameters).

5.1 System Architecture

The middleware consists of three parts: (1) a core that controls communication during execution, interpreting design inputs into machine language, (2) a 3D printer with a webcam and external hardware such as LeapMotion to recognize interactions, and (3) a previewer for an end user to visually validate design actions applied to the physical model.

The system establishes a serial communication channel with a 3D printer to send G-code line by line. It keeps track of the 3D printer's status, while checking for any event interruptions triggered by users' intervention. A user's real-time design expressions are recognized by an RGB camera and an external sensor mounted on the 3D printer. As these external devices sense design expressions, such as sketches with patterns and gestures to transform layers, the information is fed back into the system to create changes on a model.

In the system core, CuraEngine[1] CLI runs to generate a G-code file, from the off-the-shelf STL, or generated by an algorithm modeling tool (currently we support OpenSCAD and OpenJSCAD). CuraEngine enables the system to take care of printer parameters that are hard to manage using a traditional CAD system (e.g. changes infill). Once a G-code file is created, the system executes a separate G-code parser, to keep track of the 3D printer's movements in 3D vector format. The system syncs to the 3D printer to monitor its status, by listening for an 'ok' message. Through a previewer that shows the printer's movement paths, a user can visually validate what will be executed as G-code lines. This provides immediate feedback about the user's design expression. Our ultimate goal is to enable an onscreen-CAD free design process, but a user still may want to see what would be the result of her actions before the execution.

The system is developed in Javascript using Three.js and OpenCV.js, aiming at wider accessibility for designers/developers to expand the design space, using contemporary libraries of node.js, NPM[2] open-source packages.

5.2 Synchronization Between System & Printer

To enable fluid communication and bi-directional information exchange between a human and a 3D printer, the system sends G-code commands using a synchronous channel. This process prevents the system from sending the entire G-code at once, granting a user to intervene to alter the model. The printer sends a batch of commands in the size of its buffer at a time, monitor the status of the event (and holds the event until the printer completes commands in the buffer). The printer's state machine accepts multiple input conditions, waiting for an event interruption in the background. It triggers the transition between states of the machine from pause to resume, to modify the queue that stores G-codes. When an event interrupts, printer goes into the 'resume' state.

5.3 Design Inputs

In this section, we describe techniques that we adapt into our Compositional 3D Printing, and how these techniques are used as channels for expressing design intention. Any design input can replace any red zones presented in Figure 3, to initiate the printing task or to intervene in the process.

5.3.1 Sketch Input. Sketching is an activity that uses humans' muscle memories in making process, and thus helps a beginning designer engage their own cognitive reasoning in spatial modeling. In addition to the sketch-based on screen CAD systems, our technique provides special cases that are suited for realtime design expression. Using sketches a user can indicate a drawing to linearly extrude to make a volume,

[1]https://github.com/Ultimaker/CuraEngine
[2]https://www.npmjs.com/package/npm

create vertical walls, revolve the line to generate cylindrical object, or extrude with holes (Figure 10).

To capture sketches shown in the print bed, the system first extracts a foreground image to ignore background noise using an iterated graph cut algorithm [Rother et al. 2004]. In the case that a user draws sketches on top of the model partially printed, the system removes the dominant color of the foreground image. This helps get rid of filament color, to extract the user's new drawing (lips of a shaker) except the top layer of a model being printed (lid cover). The system then obtains the binary image to get pixel density and categorize them in two groups by thresholding. By cropping the image by the size of the printer's substrate [3], the system finds all contour lines in a closed loop, constructing 2D polygon vectors $P_1, P_2, \ldots, P_n$. To control the fidelity of the model, we apply polygon approximation using the Ramer-Douglas-Peucker algorithm [Prasad et al. 2012]. Among the obtained area size S_n of 2D polygons P_n, the system discards polygons using filtering, by areas that are too small, to remove noise. Empirically we set $S_{min} = 30$ and $S_{max} = 10,000$, values can later be recalibrated. The system counts valid contour lines with reasonable area sizes. These processes can recognize three unique drawing types that result in different numbers of closed 2D polygon lines as shown in Figure 10: line drawing ($n = 1$), drawing with filled surface ($n = 2$), and drawing with holes ($n > 2$).

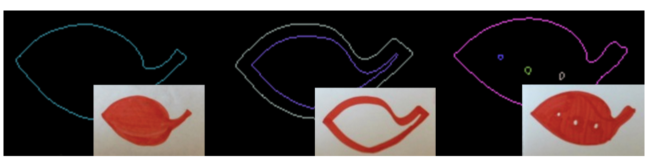

Figure 10: Three different contexts of a drawing: (from left to right) drawing in filled surface, drawing with thick stroke, and drawing with holes. Inset shows original sketches

For line drawing, capturing two different polygons (See Figure 10 middle) rather than extracting the center line of the stroke, enables the system to recognize thickness of the stroke as one of design expression. The scaling factor for each polygon is calibrated by the depth (from bottom h_0 to the height of camera h_{max}) where the sketch was placed, and in a relation with the printing bed size. The following process to create 2D polygons and 3D meshes based on these sketch contexts is described in a later section (5.4.1)

5.3.2 Physical Objects. Physcial objects can be inserted into a model to employ unique materiality (e.g. magnets), or be used to create a space inside a model (e.g. key holder). To localize an inserted object, the system remembers the center of each contour line of the sketch. Models are translated by the distance between center points given by $d = ||C_{x,y_{buildplate}}, C_{x,y_{sketch}}||$, where C refers to the 2D center point, relative to the printing base. A user locates the insertion by directly placing the object on top of the printed portion. The system then instantly captures the current scene of printing base from the top view when a user pauses, and captures the second scene when a user localizes an object and resumes printing. The system extracts the foreground image of both, so it can extract contour lines of the printed portion of a model (σ_1) and the inserted object on top (σ_2), to determine the positions of inserted objects, could be one or multiple. The system then subtracts the dominant color of σ_2 from σ_1, and thus keeps the inserted object's outlines only. Now both images follow the same pipeline described in earlier section (5.3.1), to get contour lines in 2D polygons and center positions.

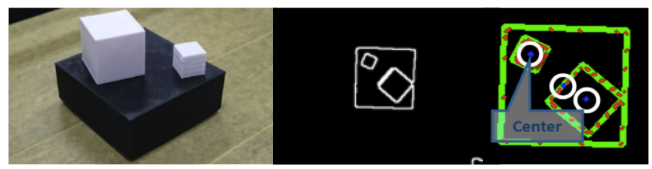

Figure 11: Inserted objects (left) are detected to retrieve relative position by comparing the position of their centers. Arbitrarily shaped objects are localized by bounding boxes.

5.3.3 Gestures. To support embodied input detection, we implemented a sensor value detection framework using Cylon.js [Cylon.js 2016]. A LeapMotion gesture sensor is used to capture three common gestures, (1) swipe to pause/resume, (2) rotation, and (3) scale with pinch, using thumb and index finger position detection. We defined (4) clapping to indicate 'repeat the history'. Whenever the predefined gestures are detected, the system pauses the current task, interprets the gestures and create an appropriate rotation matrix to apply to the layer, encoded into vectors.

5.3.4 Physical Sensor Values. Cylon.js robotics framework is also used to detect generic sensor values. Once an Arduino micro-controller with desired sensors is connected, the system reads a pin number to get sensor values that presents non-visual design expressions. Based on these values, designers can control printer specific parameters, for example, increasing extrusion amount as a physical knob turns, adding bumps according to background sound. In addition to the sketch identification by counting the number of 2D polygons, we also offer the user the ability to indicate their drawing context using a pen type, identified by RFID tags. In this manner, designers can specify which 3D effects they want to generate using different pen that similarly replicate crafting practice. For example, using a pen with a sharp tip, the drawing could indicate creating a dent.

[3] 150 by 150 pixels, starting at position x:50, y:10 of the camera frame. These numbers can be adjusted by the relative position of the camera mounted to the printer gantry and the 3D printer's build plate size.

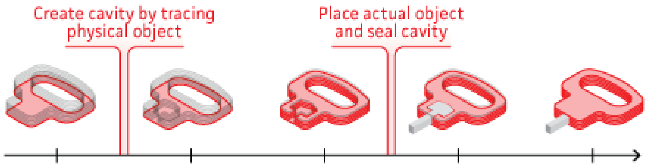

Figure 12: A physical object insertion results in a new model with the space inside that changes user's interaction with the machine

5.4 Design Output

Our middleware maps the captured design input into machines output behaviors coded in G-code. When a new G-code is created reflecting changes, the middleware replaces it with the old one in the queue.

5.4.1 STL model Creation. Sketch context known from the sketch input, (See section 5.3.1) is used to create step by step constructive geometry operations using known 2D polygons in Open(J)Scad scripts. If the number of valid 2D vector polygons found equals to 1, the system generates simple script to linearly extrude the polygon in z-axis. If more than one closed vector polygons are found, the scripts collects all polygons $PG_1, PG_2, ...PG_{n-1}$, except PG_n with the largest area, into one array A_{pg}. Then the union of linearly extruded polygons set A_{pg} is subtracted from E_{PG_n}, the linear extrusion of the largest surface. We set the extrusion height of polygons in A_{pg} always higher than that of PG_n by 1 mm, then center all in z-axis, to avoid manifold geometry. After all designated CSG operations (linear extruding, subtracting, translating, etc.) completed, the system generates an STL from the resulting polygon and calls CuraEngine to create instant output as machine language.

5.4.2 Geometry Modification by CSG operations. As shown in the assistive key turner and MechaMagnet creation workflows, inserting physical objects produces a space inside the 3D model. First, the system extrudes the contour line detected from the captured image of inserted object σ_2 vertically. As a user enters the object height by value using the GUI, the system creates a vertical volume of this object δ, localizes it in x,y-axis, then translates from the build plate's center by its own center point. This polygon is localized in xy-plane inside the original model by the height of current extruder. Taking the original models' geometry (Φ, as an STL), Open(J)Scad scripts are created to perform the CSG subtraction ($\Phi - \delta$) to generate the space for insertion. It also creates additional G-code commands to sync the printer's behavior and the user's action, pausing printing at the height where a designer's action is required, waiting for the designer to complete this action, and resuming to complete the rest of the layers as shown in Figure 12.

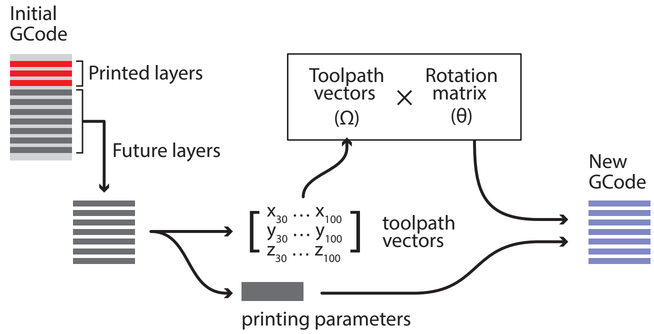

Figure 13: G-code is parsed and stored in vector(Ω) and meta data (ϵ). These information are used to update the model New vectors are spliced into a new G-code.

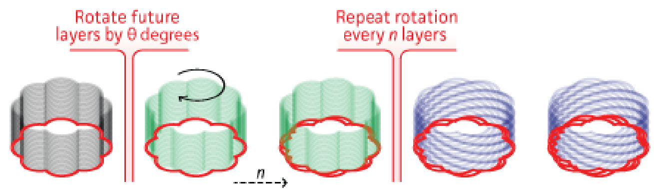

Figure 14: Intervention with a gesture, rotate with fingers, tweaks current toolpath on-the-fly. Another intervention with clapping can define the printing history

5.4.3 G-code Parsing & Splicing. When the G-code file is created, the system automatically parses it into 2D movement vectors in the same layer, by capturing toolpath commands (Ω, parsed by lines starting with G0, G1), and saves meta printing parameters (ϵ, lines starting with E, F, M, etc.) respectively. This step is for taking in-situ design inputs that might affect each layer in different ways. If a designer interrupts printing, the design input is represented in a transformation matrix (θ), then applied to this vector ($\Omega^\intercal \times \theta$), as shown in Figure 13.

5.4.4 Apply Transformation Matrix. As illustrated in the basketry example (Figure 14), hand gestures presented one of 'additional input's to enable continuous interaction with physical objects [Leithinger et al. 2011]. Our system transforms users' direct input as captured by camera sensors into a transformation matrix. The G-code parser saves the header movement of each layer in a 3D vector with the same z-value. Once the rotation matrix is created, the vector is transformed by multiplying it.

When new G-code is generated, the system splices new G-code according to these movement vectors (Ω'), recovering meta information from ϵ (E and F, fan on/off, printing temperature, etc.). Now the printer gets this new G-code, replacing the old G-code it was executing.

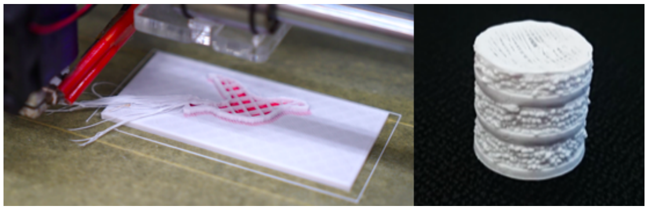

Figure 15: A line drawing gesture in the mid-air can present strands effects (left), and physical input to increase extrusion rate (stepping up the extrusion motor speed) generates interesting texture effects (right)

5.4.5 Expressive G-codes and Customized Printing Parameters. 3D printing is controlled by 3D printer's printing parameters, for example, the ratio of extruded material to movement speed. These parameters are printing time parameters that handle the quality of printing not the shape of object. Prior work tested tweaking G-codes to directly change the appearance of 3D shapes [Laput et al. 2015; Takahashi and Miyashita 2017], demonstrating the 3D printer as a tool capable of promoting its own expressivity. However, these techniques have not been reused by end users, because managing G-codes directly to change printer's behaviors is not a trivial task in traditional onscreen CAD systems. We encoded three expressive printing templates, stranding, drooping, and drooping with droplets. When a user draws lines in the mid-air, the system captures the direction of the movement and inserts the template coded in a few G-Code lines into the current queue (See Figure 15 left). After finishing, the process returns to the original task and complete printing.

Using expressive G-code templates, the system enables integrating non-traditional 3D printing effects. This technique is especially useful to present textures to promote users' tactile experiences from 3D printed objects [Kim and Yeh 2015]. Also, as the system stores vector information in one layer, a gesture to 'auto-complete' can be set by the adding the predefined templates at each point in polygon vectors, stored by G-code parser, calculating the direction by multiplying the normal of each point in constant intervals.

Our system also supports later replication and iteration. Thus, the system supports to save the final G-code created by users' on-the-fly design action to replicate the model if needed. Design actions that incorporate important design parameters can be reversed into an STL form for sharing [Makeprintable 2015].

6 DISCUSSION

This work is part of our long term vision to enable direct interaction between a human and fabrication machines, and to ultimately change the personal digital fabrication pipeline. Here we discuss points that we and the future fabrication community will need to take into account.

6.1 Supporting Exploration over Efficiency

Digital fabrication made great advances in speed, material, and precision. Recent efforts have focused on using a 3D printer to validate their design by physically fabricating the digital model. The 'efficiency' of fabrication is not the target of our work. Rather, our focus is on the exploratory nature of design, by involving the printer as a tool during this process. Although current 3D printing does not support physical undo unless special hardware such as milling pin integrated, similar to crafting or model making with physical materials, we imagine that Compositional 3D Printing will enable makers to be on-site during production, effecting real-time design decisions based on ongoing observations and reflections.

6.2 Designing New Mappings

Our middleware provides multiple channels to interaction designers, so they only have to care about designing a new input expression, rather than the internal algorithm to perform mapping. To expand the design space, future researchers can start from designing interaction styles to adequately present on-the-fly interactions (top-down). Yet, an interaction designer, familiar with their own modeling skills from existing tools (e.g., how to attaching, clipping, or cutting are expressed by new input styles) might be limited. We envision that another type of users can take a different approach, thinking of new types of machine behaviors, and developing those interactions that express their desired features (bottom-up). In that case, understanding the production time parameters (i.e. focus height, speed and power in laser cutter to create atypical functions on the special materials [Mueller et al. 2014]) is the most significant factor. This makes it possible to imagine future hardware systems, such as robotic arms to mold pottery on a turntable. This might entail number of production time factors, such as the x, y, and z-motor control of each arm or the rotation speed of the turntable. By exploring that effects the combinations of these parameters can create, an interaction designer might be inspired to propose well-suiting interactions for those effects on-the-fly.

6.3 Expanding to other Fabrication Machines

Our design scenarios were implemented for FDM 3D printing, as it is the most well-known, affordable, open-sourced design machines; in contrast to laser cutters and milling machines that still cost several thousand dollars and possess closed internal mechanisms. In comparison, our middleware system manipulates machine movement in G-code, a common language for any CNC machines. As the middleware parses the machine language into movement and meta-parameters, any fabrication machines with toolpath planning by x,y,z vector sets are compatible. Design expression for additive and subtractive fabrication might differ. By understanding the right interaction methods and the machine parameters they affect (as in Figure 9), our system can be used in the desired direction, which in turns helps widen the design space for this machine. This would be an ideal way of expanding the design space of this paradigm well beyond what currently exists for 3D printing to open-ended interactive fabrication.

6.4 Fab Machines as an Intelligent Assistant

The ultimate goal of developing a middleware and design space for Compositional 3D Printing, empowering a user to continuously design with improvised design expressions, is to facilitate users' participation in the entire pipeline including the physical production. In this work we explore the possibility of casting digital fabrication machines such as 3D printers as a music mixer. We envision a new pipeline that catalyzes co-working with fabricators as an intelligent design assistant. Fabrication tools must understand the design context, perceive the human's actions, and proactively assist with arbitrary design actions by incorporating its precise, rapid computation in the production process. To carry the benefits of humans' artisanship to the digital fabrication world, machines must support the designer's fine control of tasks, and to allow refinements of design by being intelligent, supportive, and revealing of their own processes.

7 CONCLUSION

We introduced Compositional 3D Printing as a new paradigm for digital fabrication where a designer can "compose" a 3D model by interacting with a 3D printer. We developed a system functions as middleware to support compositional 3D printing and presented several workflows as a part of design space, demonstrating an expanded design space of 3D modeling with six workflows.

ACKNOWLEDGMENTS

We thank Homei Miyahita and Abigale Stangl for their insightful comments, William Ridley and Jen Liu for elaborating the paper. This research is funded by NSF Grant No. IIS-1453771 and the Adobe Research Fellowship.

REFERENCES

Patrick Baudisch and Stefanie Mueller. 2017. Personal Fabrication. *Foundations and Trends® in Human–Computer Interaction* 10 (2017).

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 *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. 29–39.

Cylon.js. 2016. JavaScript framework for robotics, physical computing, and the Internet of Things using Node.js. https://cylonjs.com/. (2016).

Norman K Denzin and Michael D Giardina. 2016. *Qualitative Inquiry-Past, Present, and Future: A Critical Reader.* Taylor & Francis. https://books.google.com/books?id=1F-TDAAAQBAJ

Anhong Guo, Jeeeun Kim, Tom Yeh, Scott Hudson, Jeniffer Mankoff, and Jefferey Bigham. 2017. Facade: Auto-generating Tactile Interfaces to Appliances. In *Proceedings of the 35th Annual ACM Conference on Human Factors in Computing Systems (CHI '17)*.

Takeo Igarashi. 2010. Computer graphics for all. *Commun. ACM* (2010).

Yuki Igarashi, Takeo Igarashi, and Jun Mitani. 2011. Beady: Interactive Beadwork Design and Construction. In *SIGGRAPH Asia 2011 Sketches (SA '11)*. Article 17, 17:1–17:2 pages.

Gabe Johnson, Mark D. Gross, Jason Hong, and Ellen Yi-Luen Do. 2009. (2009).

Jeeeun Kim, Anhong Guo, Tom Yeh, Scott Hudson, and Jeniffer Mankoff. 2017. Understanding Uncertainty in Measurement and Accommodating its Impact in 3D Modeling and Printing. In *Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17)*.

Jeeeun Kim and Tom Yeh. 2015. Toward 3D-Printed Movable Tactile Pictures for Children with Visual Impairments. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. 10.

Gierad Laput, Xiang 'Anthony' Chen, and Chris Harrison. 2015. 3D Printed Hair: Fused Deposition Modeling of Soft Strands, Fibers, and Bristles. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. 593–597.

Daniel Leithinger, David Lakatos, Anthony DeVincenzi, Matthew Blackshaw, and Hiroshi Ishii. 2011. Direct and Gestural Interaction with Relief: A 2.5D Shape Display. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. 541–548.

LIA. 2014. 3D Sculpting. http://liasomething.tumblr.com/. (2014).

Makeprintable. 2015. Reverse All your GCode with G-code Reverser. https://makeprintable.com/blog/makeprintables-g-code-reverser/. (2015).

Yuki Mori and Takeo Igarashi. 2007. Plushie: An Interactive Design System for Plush Toys. *ACM Trans. Graph.* 26, 3, Article 45 (July 2007).

Stefanie Mueller, Bastian Kruck, and Patrick Baudisch. 2014. Laser Origami: Laser-cutting 3D Objects. *interactions* 21, 2 (March 2014), 36–41.

Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive Construction: Interactive Fabrication of Functional Mechanical Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. 599–606.

Hyunjoo Oh, Jeeeun Kim, Cory Morales, Mark Gross, Michael Eisenberg, and Sherry Hsi. 2017. FoldMecha: Exploratory Design and Engineering of Mechanical Papercraft. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction (TEI '17)*. 131–139.

Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. 2016. On-The-Fly Print: Incremental Printing While Modelling. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. 887–896.

Huaishu Peng, Amit Zoran, and François V. Guimbretière. 2015. D-Coil: A Hands-on Approach to Digital 3D Models Design. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. 1807–1815.

Dilip K. Prasad, Maylor K.H. Leung, Chai Quek, and Siu-Yeung Cho. 2012. A novel framework for making dominant point detection methods non-parametric. *Image and Vision Computing* (2012). https://doi.org/10.1016/j.imavis.2012.06.010

Ron Rael. 2016. GCODE.Clay | Emerging Objects. http://www.emergingobjects.com/project/gcode-clay/. (2016).

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In *ACM SIGGRAPH 2004 Papers (SIGGRAPH '04)*. 309–314.

Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. 2011. SketchChair: an all-in-one chair design system for end users. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction.* ACM Press, 73. https://doi.org/10.1145/1935701.1935717

Donald A. Schön. 1992. Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems* 5, 1 (mar 1992), 3–14. https://doi.org/10.1016/0950-7051(92)90020-g

Haruki Takahashi and Homei Miyashita. 2017. Expressive Fused Deposition Modeling by Controlling Extruder Height and Extrusion Amount. In *Proceedings of the 35rd SIGCHI Human Factors on Computing Systems (CHI '17)*.

Christian Weichel, John Hardy, Jason Alexander, and Hans Gellersen. 2015. ReForm: Integrating Physical and Digital Design Through Bidirectional Fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. 93–102.

Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen. 2014. MixFab: A Mixed-reality Environment for Personal Fabrication. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. 3855–3864.

Karl D.D. Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D. Gross. 2011. Interactive Fabrication: New Interfaces for Digital Fabrication. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. 69–72.

Clement Zheng and Ellen Yi-Luen Do. 2018. Mechamagnets: Tactile Mechanisms with Embedded Magnets. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '18)*. 57–64.

Amit Zoran and Joseph A. Paradiso. 2013. FreeD: A Freehand Digital Sculpting Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. 2613–2616.