Plain2Fun: Augmenting Ordinary Objects with Interactive Functions by Auto-Fabricating Surface Painted Circuits

Tianyi Wang, Ke Huo, Pratik Chawla, Guiming Chen, Siddharth Banerjee, Karthik Ramani *
School of Mechanical Engineering, Purdue University
West Lafayette, IN 47907, USA
{wang3259, khuo, chawla0, chen1956, banerj27, ramani}@purdue.edu

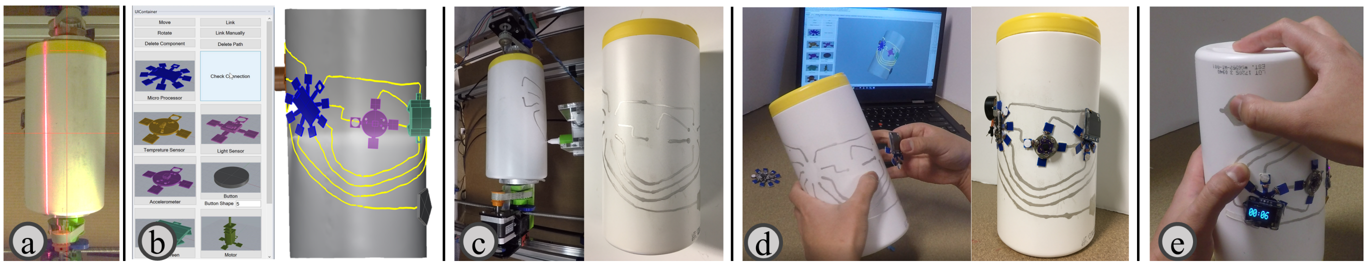

Figure 1. Workflow of Plain2Fun: users first scan a chosen object on the machine (a); we provide a design tool for users to place the components on the obtained 3D model and generate the conductive paths (b); a digital machine is used to draw the paths on the object with a conductive pen (c); then the users simply align and attach the soft housing bases onto the object (d); finally, a physical prototype on the existing object with customized interactive functions is finished (e).

ABSTRACT
The growing makers' community demands better supports for designing and fabricating interactive functional objects. Most of the current approaches focus on embedding desired functions within new objects. Instead, we advocate repurposing the existing objects and rapidly authoring interactive functions onto them. We present Plain2Fun, a design and fabrication pipeline enabling users to quickly transform ordinary objects into interactive and functional ones. Plain2Fun allows users to directly design the circuit layouts onto the surfaces of the scanned 3D model of existing objects. Our design tool automatically generates as short as possible circuit paths between any two points while avoiding intersections. Further, we build a digital machine to construct the conductive paths accurately. With a specially designed housing base, users can simply snap the electronic components onto the surfaces and obtain working physical prototypes. Moreover, we evaluate the usability of our system with multiple use cases and a preliminary user study.

*School of Electrical and Computer Engineering (by courtesy)

DIS '18, June 9–13, 2018, , Hong Kong

DOI: https://doi.org/10.1145/3196709.3196791

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation (e.g. HCI): User Interfaces, Interaction Styles

Author Keywords
DIY; Computational Fabrication; Creativity Support Tool.

INTRODUCTION
The DIY makers community is rapidly growing primarily because of the proliferating digital fabrication tools such as 3D printer, laser cutter, and CNC machines. For mechanical design, makers can easily access numerous 3D models from online sharing community (e.g., Thingiverse [37]) or customize 3D shapes with easy-to-use computer-aid-design (CAD) software. Thus, in most cases, the physical realization of mechanical functions is well supported by both hardware and software. On the other hand, makers can also embed interactive functions including sensing, lights, displays, and sound into the fabricated objects [30, 32, 42]. Yet, current design and fabrication tools (i) do not support electronics design on the artifact directly, (ii) assume users have a high level of knowledge from different domains, and (iii) are not user-friendly to support creating interactive functions.

Although technologies like Voxel8 [40] demonstrate the idea that combines circuits fabrication with 3D printing, they are not popular in the mainstream of personal fabrication approaches yet. More importantly, the methods which integrate the fabrication of objects and the construction of electronics only cater for the demand of inventing a new design, rather than utilizing an existing object. On the other hand, a post-assembly approach provides more flexibility in terms of incorporating off-the-shelf electronic components on existing plain objects. A typical solution for the post-assembly is grouping the components on a printed circuit board (PCB). However, designing a working PCB requires experiments on prototypes and extensive knowledge of the software. For prototyping, makers usually hook up the circuit on a breadboard with conductive wires. Moreover, mounting the PCB or the breadboard on the existing objects involves extra mechanical design of fixtures. Further novices often choose breakout board packages which are hook-up ready for prototyping circuits conveniently [9]. Thus, we develop a distributed surface mount method where users attach the different breakout boards on the object surfaces directly. In doing so, we greatly simplify the prototyping process and reduce the time substantially.

We propose Plain2Fun, a pipeline that allows novice users to design and construct an interactive prototype on a existing object's surface, as shown in Figure 1. To achieve our goal, we need to resolve three main challenges: (i) attaching the rigid and flat breakout boards on the curved surface; (ii) connecting the components with conductive paths on the surface in a user-friendly manner; and (iii) assisting users to generate the 3D conductive paths in a virtual environment and ensure a working prototype design.

First, we design and 3D print a housing base to extend the breakout boards with split and flexible pads. The split structure and flexibility of the material allows the base to better conform along the curved surface. We solder the pins of the rigid and flat breakout boards onto wires. The wires are also wrapped on the base and serve as contact points to the conductive paths.

Second, we automate the construction of conductive paths as much as possible to ensure the accuracy and lower the users' workload. Among all the common conductive path construction methods, including copper tape and conductive ink, the conductive ink is the most compatible one and easy to build using a computer numerical control (CNC) machine. Using low-resistance conductive ink pens which are commercially available [34], we build a 4 degree-of-freedom (DOF) machine to draw conductive paths accurately on the objects. In addition, our machine consists of a laser scanner for in-situ capture of the 3D shape of the objects.

Third, we develop a design environment to support users to customize functions such as placing the housed breakout boards of the electronic components on the scanned 3D model. We also computationally generating the drawing path and machine code for fabrication accordingly. We create a weighted geodesic algorithm to find a path with the shortest distance while avoiding intersecting with the components and existing paths. After interfacing with the machine and constructing the paths, users simply align the selected components onto the drawn paths and snap the base on the surfaces to complete the physical prototype.

In summary, we build towards the broad goal of supporting users to easily prototype interactive objects. Specifically, Plain2Fun is the first attempt, to the best of our knowledge, to enable authoring functions onto already existing plain objects with an interactive design tool that operates along with a digital fabrication machine. Following is a list of our contributions:

- A pipeline to author interactive functions on existing objects and construct prototypes by simply sticking a flexible housing base on the objects.
- Development of a digital machine for drawing conductive paths automatically.
- Building an interactive design tool facilitating users to customize circuits layout, generating conductive paths while avoiding intersections, and preventing mis-connections.
- Example use cases demonstrating the usages of Plain2Fun on different objects and user studies evaluating the usability.

RELATED WORK

3D Interactive Object Fabrication

Numerous works have been proposed to support makers to design and fabricate 3D interactive objects. As a major fabrication method in makers' community, enabling 3D prints with interactivities have been studied. By leveraging the internal structure of the objects, optics [43] and acoustic [13] based interactions have been enabled. Further, the functional components have been embedded within the empty chamber inside the 3D prints [11, 29, 30]. Previous works also showed integrated fabrication approaches where the construction of conductive materials is combined with 3D printing. Hudson [14] integrated conductive threads with fabric-based 3D printing. Capricate [32] embedded capacitive sensing capability using a multi-material printer with carbon-based conductive filament, while Voxel8 [40] has demonstrated printing conductive inks. Surfcuit [38] generated channels and holes on the 3D prints and required users to manually place copper tapes in the channels and solder the components. Roquet et al. introduced an origami folding method to integrate flexible paper circuits and 3D printed pieces together as an interactive object [8]. However, these approaches primarily focused on design and also fabricate new objects from scratch. As opposed to duplicating new objects with the same physical affordance, makers may simply want to add interactivity to the existing objects. Here, Plain2Fun supports users to re-purpose the existing objects and rapidly transform them into interactive ones.

Augmentations on Existing Objects

Recent works have studied augmenting existing objects by attaching new physical parts onto or directly 3D printing around the objects [6, 16, 36]. Moreover, RetroFab extended the idea of mechanical hijacking and automated the physical control interfaces [27]. Apart from mechanical augmentations, previous works also explored wireless or touch interactions with the existing objects. PaperID presented tagging objects with Radio Frequency Identification (RFID) tags and detecting users' interactions via an RFID reader [19]. TRing leveraged magnetic sensing and enabled users to interact with objects which were embedded with a magnet [45]. Further, capacitive and acoustic based sensing techniques have also been exploited to enable touching interactions with objects [28, 24, 46]. To avoid modifying the existing objects, stretchable soft sensors/displays have been introduced as an extra and conformal layer on the 3D objects [42, 44]. Inspired by the form factor of a soft layer, we propose to house the off-the-shelf breakout boards on flexible bases and attach them on the surfaces of object so that the assembly remains simple and requires no modifications on the objects. Besides, using off-the-shelf components allows users to choose functions with larger freedom including various sensors, and outputs with lighting and sound, etc.

Designing and Constructing Circuits on Surface

Despite the existence of general-purpose PCB design tools (e.g., Eagle [4]), more novice-friendly systems have been developed. Anderson et al., proposed an interactive system to generate layout design for breadboards based on users' high-level inputs [3]. As the use of breadboards is largely limited because its bulkiness, researchers investigated to deploy the circuits with more flexibility. For example, copper tape has been used widely for constructing conductive paths on flat surfaces [8, 26, 31]. Moreover, low resistance capacitive ink based circuits can be constructed with drawing pens manually [7, 21]. While PaperID allows for accurate drawing with a stencil [19], many previous works used ink-jet printer to construct precise traces on foldable or flexible substrates [23, 35, 39, 15]. As for 3D objects, Voxel8 [40] prints conductive inks together with plastic filament. In SurfCuit [38], stripes of copper tapes were manually placed in the channels on the 3D prints and connections were crafted by soldering on the stripes. More recently, a water transfer printing based technology has been proposed to transfer conductive patterns from a special multi-layer film onto objects [18, 12]. In Plain2Fun, we develop a customized 4 DOF CNC like machine with a capacitive silver ink pen to draw traces directly on surfaces of existing objects.

As complete design and fabrication workflows targeted at novice and casual makers, some of the above proposed approaches provided computational supports in their corresponding design tools [22, 26, 35, 38]. SurfCuit was the closest work to ours as it provided a circuit design tool on 3D surfaces. We manage to avoid intersections between paths by proposing a weighted geodesic algorithm while SurfCuit resolves the intersections interactively by users. Further, SurfCuit provided circuit design with low-level independent components such as resistors, capacitors, etc. Instead we adopt breakout board level components which are hook-up ready to avoid complex schematic design. Moreover, in SurfCuit, the components were soldered onto copper tubes and above the surface whereas in our case, the flexible housing base attaches to the surface. By introducing a gradient weight distribution around the bases, we also avoid intersections between the paths and the components.

PLAIN2FUN

Plain2Fun is a design and fabrication pipeline supporting users to add interactive functions to plain daily objects. We compose this pipeline with three major modules: (i) a digital fabrication machine which perform 3D scanning of the object and circuit construction with a conductive ink pen; (ii) a library of electronic components which are housed on soft bases and ready to be attached on the surface; and (iii) a design tool for users to customize the circuit layout directly on the 3D surface and interface with the fabrication tool.

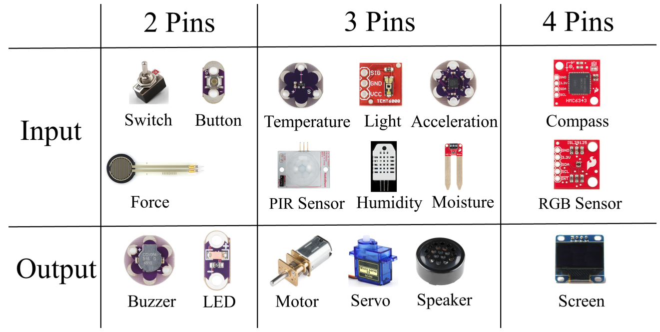

Figure 2. Electronic components library.

Here we explain the pipeline together with a typical use case of Plain2Fun as illustrated in Figure 1. A creative maker has a plain wet tissue container and would like to transform it to a digital sand timer like device and preserve the container's original shape and function. To deliver the function of digital timer, a quick formulation for implementation could include a micro-controller, a display, a speaker, an accelerometer and a capacitive button. Numerous online vendors provide breakout boards for each of the electronic components. The traditional way usually involves hooking them up together on a bread board and then attaching the whole board onto the object. However, Plain2Fun supports simply attaching the components directly on the surface of the object thus allowing for a more flexible way of deploying the circuit.

With Plain2Fun, users first scan the plain object on the machine (Figure 1 (a)), then import the scanned 3D model into our design tool (Figure 1 (b)). The users select the components from the library and place them onto the surface of the object in a drag-and-drop manner. Based on the function of each component, the design tool provides color coded pins to suggest possible connections. Then the users can either connect the pins with manually routed paths or auto-generated ones. Our routing algorithm calculates the shortest paths on the surfaces between pins while avoiding intersections with the components and the existing paths. Further, the digital machine constructs the conductive paths with a silver ink pen (Figure 1 (c)). Lastly, the users simply align the physical housing bases with the drawn paths and then attach the bases onto the surfaces to finish the physical prototype (Figure 1 (d)).

Electronic Components Library

To deliver interactive functions, makers usually need to create a recipe with a micro-controller unit (MCU), a power unit, and some functional components for input/output. We screen the components based on two rules: (i) components with proper breakout board package which is hook-up ready; and (ii) breakout boards with small footprint as the surface area of the objects maybe limited. We select a normal MCU breakout board (Adafruit Trinket M0 [2]) which retains enough pinouts on a small footprint. We search a wide range of off-the-shelf functional components which are commonly used as inputs and outputs from online vendors such as Adafruit [1], Sparkfun [34], etc. As shown in Figure 2, we selected a total of 17 functional components.

Based on the collected components, we analyze the pin configuration and classify the connection strategies. As the connection between the battery and the MCU is fixed, we primarily refer to the pin configuration of the MCU and the inputs/outputs.

- One-pin capacitive pad directly connects to an I/O pin.
- Two-pin usually involves a ground (G) and an I/O pin (IO).
- Three-pin consists of a G, a power pin (P) and an IO.
- Four-pin includes a G, a P and two IO for I^2C bus (i.e., SDA, SCL).
- MCU has a pair of G and P and 5 IO of which two pins can be used as an I^2C bus.

Note that the P and G can be commonly shared by different components. In extreme cases, our selected MCU supports up to 5 independent functional components which consumes all 5 general IO, or one I^2C component which uses I^2CSDA and I^2CSCL pins plus 3 regular components which use the rest 3 general IO.

Topology Management

Unlike prototyping a circuit with a breadboard where users connect the components with wires, a surface mount approach usually requires one to resolve the intersection problem. For normal double-sided PCBs, we use through holes to avoid intersections between conductive paths. However, this solution cannot be applied on surfaces of existing objects without drilling holes. Instead, our approach resembles a single-sided PCB design where the existing conductive paths divide the surface into multiple closed sub-areas which block the pins inside the areas to reach out. We develop a topology management to ensure an existence of solutions to connect as many components as possible properly without introducing intersections.

To conform and attach the breakout boards onto curved surfaces, we expand the flat boards by housing them on polygonal flexible bases as shown in Figure 3. We further split the housing base to a multi-arm structure. This way, the bases can better conform onto highly curved surfaces. To better describe the topology problem, we denote each component as a circular sequence of pins along the counter-clockwise direction, e.g., 2-pin equals $\{G, IO\}$. Considering a circular permutation [41], we derive that a 2-pin component has 1 possible configuration, while a 3-pin has 2 possibilities (i.e., $\{P, G, IO\}$ and $\{IO, G, P\}$), and a 4-pin has 6. A proper connection between a functional component and a MCU needs to avoid intersections in two level: (i) internal intersections caused by self-needed pin connections, and (ii) external intersections which interfere the existing paths from other functional components. To avoid internal ones, the MCU pin sequence must contain a subsequence which matches the sequence of the functional component's pins in a reverse way. For example, to connect a 3-pin component with a counter-clockwise circular sequence $\{P, G, IO\}$ to the MCU, the MCU must contain a subsequence as $\{IO, G, P\}$. Second, the to-be-connected pins must stay in a geometrically connected area. We tackle the topology management as follows.

- We first examine the pin sequences of the original MCU breakout board, i.e., $\{P, IO/SCL, IO, IO/SDA, G, IO, IO\}$. We can extract valid subsequences for 2-pin ($\{G, IO\}$), 3-pin ($\{P, G, IO\}$, $\{IO, G, P\}$), and 4-pin components ($\{P, SCL, SCA, G\}$) respectively. Then a naive design would be directly extending the components with the exact number of pins and the corresponding sequences, as shown in Figure 3 (a, d).
- A closer inspection at the naive design uncovers some issues. We illustrate a failure case for the 3-pin naive design in Figure 3 (b) where after connecting a 3-pin with $\{P, G, IO\}$ to a subsequence $\{IO, G, P\}$ on the MCU, no more subsequence with $\{IO, G, P\}$ is available in any connected area. This means we cannot connect another 3-pin component with $\{P, G, IO\}$. To address this kind of failure cases, we augment a 3-pin configuration with an extra G to form a 4-pin axis symmetric configuration as shown in Figure 3 (a). With such augmentation, we are able to find a strategy to connect 5 independent functional components which consumes all general IO pins as shown in Figure 3 (c).
- For a naive 4-pin design as shown in Figure 3 (d, e), after connecting an I^2C component, a general IO pin will be blocked and thus wasted. We augmented the 4-pin configuration with an extra pair of P and G to resolve this issue as demonstrated in a successful case in Figure 3 (f). This new configuration also enables the connection of a series of I2C components using I2C bus. We can replace the regular 4-pin component in Figure 3 (f) with a 6-pin I2C component, and connect all its functional pins to the first I2C component. Furthermore, a third I2C component can be added between the second one and the MCU in the same way.
- Further, we augment the original 7-pin configuration of MCU with an extra pair of P and G to introduce more flexibilities for connections (Figure 3 (g)).
- The pin sequence of an augmented MCU can be written as $\{P, IO, IO, IO, P, G, IO, IO, G\}$. For the components with more than four pins, we design their pin sequences as $\{G, P, IO, IO, IO\}$ (5 pins), $\{IO, G, P, IO, IO, IO\}$ (6 pins) or $\{IO, IO, G, P, IO, IO, IO\}$ (7 pins). Therefore the vacant pins on the MCU can be in the same geometrically connected area with a pair of ground and power pins and are ready to be connected with other 3-pin or 4-pin components.

Conductive Path Generation

With the proposed topology management, Plain2Fun guarantees the existence of proper connection solutions. However manually routing between the pins could be painful and time-consuming for users. We propose a path generation algorithm to support the users. In order to reduce the resistance of the conductive paths, each path should be as short as possible. On a surface, the shortest path between any two points is defined as a geodesic. Moreover, in a surface mount approach, a conductive path is not allowed to intersect with existing paths

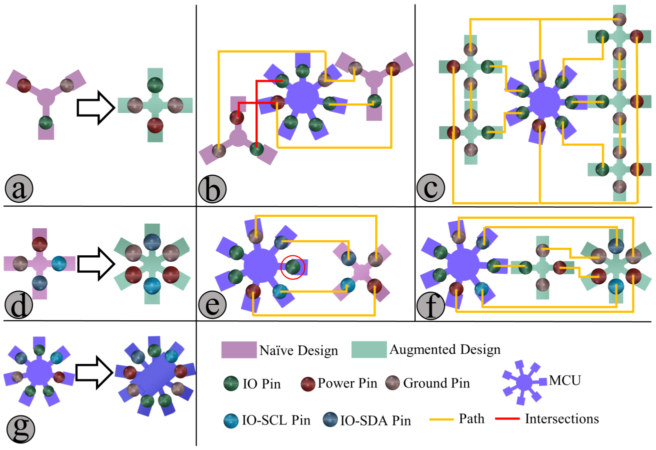

Figure 3. Topology management strategy: augmenting the pin configurations of the housing bases (a, d, g); failure cases with naive designs (b, e); and proper connections with augmented designs (c, f).

or electronic components. Thus, in addition to the ordinary geodesic computation, our algorithm needs to deal with intersections. Besides, the paths should be generated in real time to preserve good user experience.

To resolve the above challenges, we apply the geodesic algorithm described in [17] on a weighted mesh. In the original algorithm, by adding Steiner points on edges of the mesh and further segmenting between these Steiner points, we construct an augmented graph **G**. **G** consists of the Steiner points, the segments, as well as the original mesh edges and vertices. Then the shortest path in **G** can be found using Dijkstra's algorithm. Instead of calculating the path lengths using normal Euclidean distances, we introduce weights to each face of the original mesh and define a weighted length.

A weight value w_{s_i} is assigned to each face s_i of the mesh. For a path **P** in **G**, its weighted length is calculated as:

$$L_P = \sum_{p_i \in \mathbf{P}} w_{p_i} l_{p_i} \qquad (\forall p_i \in s_i, w_{p_i} = w_{s_i})$$

where p_i refers to the short segments that composes path **P**. This way, we can assign different weights to the faces thus differentiate the existing paths and the components covered areas from the empty surface areas as shown in Figure 4. By applying the weighted distances, the Dijkstra's algorithm finds a "shortest" path which is steered away from the occupied areas. To this end, the algorithm generates as short as possible paths while avoiding intersections. The Dijkstra's algorithm is implemented using a min-heap structure and its time complexity for generating a path is $O(nlog(n))$. In practice, on a mesh with 33000 faces, it takes 0.3s on average to generate a path on a desktop with an Intel i7-6700k processor.

We initialize $s_i = 1$ for all faces. Whenever a user add a component on to the surface, we set the weights of the faces right beneath it as a high constant value (2048). For the faces in peripheral regions of a component, their weight values decrease exponentially from 2048 to 256 with respect to the distance between the face centroid and the component center

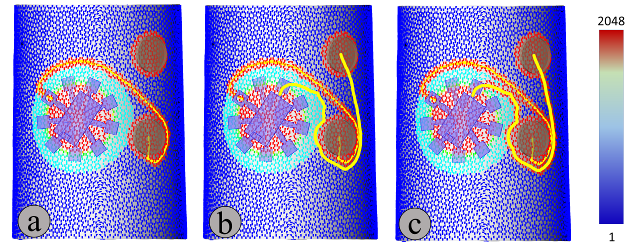

Figure 4. Weight distribution of a mesh while adding a conductive path. (a) The weight map before adding the path. (b) A new path (yellow line) is generated using weighted geodesic algorithm and avoid touching the existing components and paths. (c) Weight distribution of this mesh is updated after adding a path.

(Figure 4 (a)). Once a path is generated, we set the faces beneath a thickened belt along the path weighted as 2048. The thickened belt has a width of 5mm, which is twice the width of the path drawn by the conductive pen. (Figure 4 (b, c)). As shown in Figure 4 (a), due to the weights assigned to the peripheral regions of a component, the generated path is steered away from these regions thus leaves sufficient room for adjacent pins. Overall, the result demonstrates our weighted geodesic algorithm successfully generates as short as possible paths while avoiding intersections.

IMPLEMENTATION

Digital Fabrication Tool

As shown in Figure 5, we build a CNC machine which integrates scanning and drawing functions. The machine consists of a drawing head mounted with a conductive ink pen (CSIP-998), a customized laser scanner, and a 4-DOF motion system. To move the drawing head horizontally, we install two timing belt driving systems along X and Y axes. A pair of universal clamps is used to fixate the object and rotate it along the center axis. We can adjust the clamps' openness by moving them along X axis and lock the positions during scanning and drawing. Further, since we aim at drawing strokes on a 3D surface, another DOF is needed on Z axis to work together with the rotational DOF. A pair of ball screw driving systems are used to move the objects up and down. We use 6 step motors to deliver 4 DOF movements, i.e., 2 separate motors for X and Y axes, 2 synchronized motors for rotational and 2 synchronized motors for Z axis. All the motors are controlled with motor driver boards (Pololu A4988 [25]) through an Arduino Mega 2560. The overall size of the machine is $\sim 60 \times 60 \times 60$ cm and the cylindrical volume for holding an object is limited with a height of 25cm and a radius of 12.5cm.

In Plain2Fun, users first adjust the clamps' openness to fit an object, place the object, and then tighten the clamps. The anti-slip sticky pads on the clamps further increase the frictions to hold the objects. In our test, we are capable of holding heavy objects (e.g., 2kg). Since we pre-calibrate the scanner with respect to the machine coordinate system, we simply move the objects and align it with the laser scanner. The laser scanner is modified based on an open source firmware and software [10] which runs on a Raspberry Pi and connects with a desktop through Wi-Fi. After scanning, we retrieve the scanned model

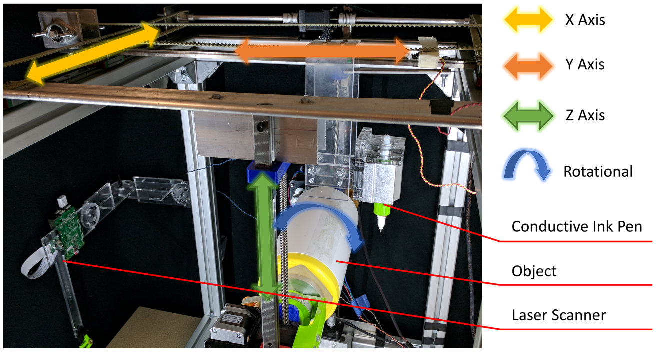

Figure 5. Our digital fabrication tool with 3 translational and 1 rotational DOF.

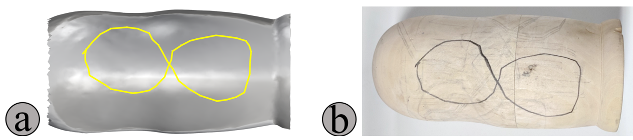

Figure 6. Comparison between the designed shape and the drawn paths.

from the Raspberry Pi and repair the mesh using Netfabb [5]. Users design their circuit layout on the 3D model using our design tool. We store the generated paths as dense points, and transform them into a cylindrical space, then calculate the movements for the drawing header and the object (r, X, Z) based on the cylindrical coordinates. The fabrication tool interfaces with the design tool through serial communication and execute the movement commands.

We evaluated the drawing accuracy and path conductivity of the fabrication tool. First we tested the positioning accuracy and repeatability of the drawing results using the digital machine. We generated an "8" shape curve on a scanned model of a wooden doll and use the machine to draw this curve on the doll repeatedly 10 times with a pencil (Figure 6). The width of the pencil trace is about 0.5mm, so this quick qualitative examination showed that our machine had reasonable positioning accuracy and repeatability (below 0.5mm) for drawing paths on curved surfaces.

Then we tested conductivities of conductive paths on different materials. We selected three objects made of common materials including glass, wood, and plastic. On each object, we drew 12 curves with the conductive ink pen (30mm $\times$4, 60mm$\times$4, 120mm$\times$4), and measured the resistance. The result showed that the average resistance on these three materials remain similar and low (glass with $\bar{r} = 1.63\Omega/cm, SD = 0.4\Omega/cm$, wood with $\bar{r} = 1.07\Omega/cm, SD = 0.4\Omega/cm$ and plastic with $\bar{r} = 1.65\Omega/cm, SD = 0.5\Omega/cm$). Thus, our machine with conductive ink pen can be used to construct low current circuits for prototyping interactive objects.

Soft Housing Base Fabrication

We recall the split multi-arm structure design which greatly enhances the soft housing bases to be better conformed onto heavily curved surfaces. In this section, we mainly address

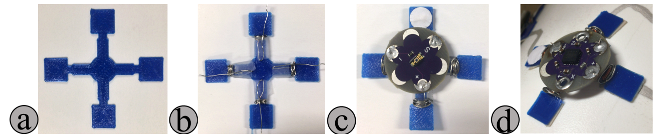

Figure 7. Soft housing base fabrication.

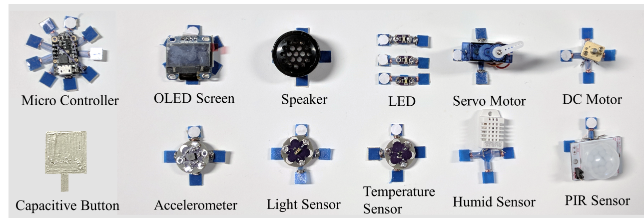

Figure 8. A library of electronic components

the fabrication of the base and housing of the breakout board. We first 3D print the multiple arm thin substrate (0.75mm) using Thermoplastic Urethane (TPU) filament (Figure 7 (a)). Then we wrap bare metal wires on each arm which serve as electronic contact pads (Figure 7 (b)). We solder the breakout boards onto the loose end of the metal wires according to the pin sequence studied in Topology Management (Figure 7 (c)). Strong Double side tape pads are attached on to the square end areas of the housing base to maintain physical contact with the surfaces of objects. With such design, users can simply peel off the protect layer from the double side tapes and stick the component onto the object surfaces (Figure 7 (d)). To facilitate users aligning the housing bases with the circuit layout in the design environment, we further mark one of the pins with a white dot on its physical and virtual models. Figure 8 shows a group of breakout boards housed on the bases. Because we intend to develop a kit for makers, in this prototypical implementation, we prepare the housing bases for users ahead. Although the flexible PCB technology is available for massive production, in our implementation, we choose to fabricate external bases for housing various off-the-shelf breakout boards.

To examine the contact property between the soft housing base and conductive paths on the objects, we attach multiple housing bases (8 pins) onto three objects (Figure 9) with different material (glass, wood, plastic) and different surface curvatures. On each objects, we randomly chose four positions and use our fabrication machine to draw short conductive paths as contact points at each position. After all bases were attached, we measured the conductivity between the metal wire on housing base and its corresponding conductive path on the object using a multimeter. All 32 pins on each object were successfully connected to the conductive paths with an average resistance 2.27Ω ($SD = 0.7\Omega/cm$). Therefore, our surface attaching bases suffice the basic contact requirements.

User Interface

The user interface of our design tool is implemented as a plug-in for Rhino 5 CAD software using *C#* language and Rhino Common SDK and runs on Windows. Our tool provides

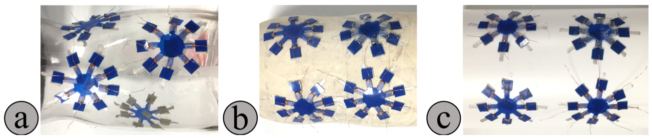

Figure 9. Attach 8-pin housing bases on objects with different material (glass (a), wood (b), plastic (c)) and shapes.

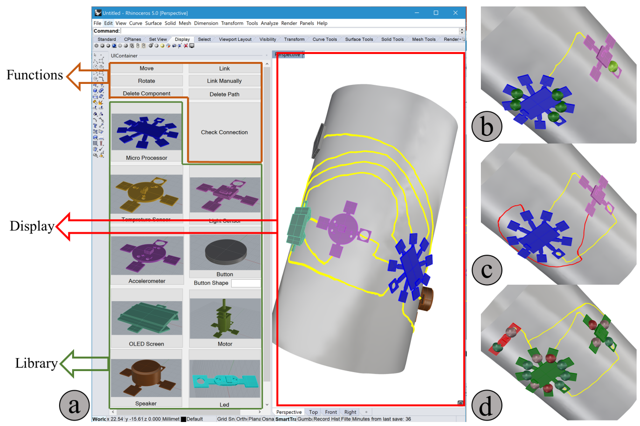

Figure 10. User interface of the design tool: UI layout (a), path generation (b), intersection warning (c), and connectivity check (d).

users with the functions to operate electronic components, to operate conductive paths, and to facilitate users planning a proper connection. In addition, our tool interfaces with the digital machine for maintenance and fabrication purposes.

Operating components. As shown in Figure 10 (a), we provide a library of functional electronic components in the user interface (UI). Users import the selected component by drag and drop the model from the library into the display region. Then by clicking and placing the model on the surface of the object, the component model is transformed to the click point on the surface and re-oriented to the normal direction at this point. We constrain the original translation and rotation operations in Rhino so that the movement of the component always remains on the surface.

Operating paths. To connect two components, users choose one pin on each of them to add a new route. Our tool prevents users to connect pins which do not match. The users have two options to create paths, i.e., manual and automatic path generation. Our algorithm calculates an as short as possible path while avoiding intersections with existing paths and the components. Also, the users can link manually by clicking points on the surface the object. If any newly added path intersects with existing paths as shown in Figure 10 (c), it will be marked as red to inform the user. Further, if the user moves any component which is already connected with some conductive paths, these paths will be recalculated based on the new position of the component. Similarly, if any component is moved onto some existing conductive paths, these paths will be regenerated and circumvent the component.

Facilitating functions. Users can visualize the connection suggestions by clicking "check connectivity" button anytime. We provide color coded connection advices, i.e., the pins which match with each other will be marked with the same color. For example, in Figure 10 (d), a red pin of the functional component represents the power pin, and all available power pins from the MCU and other components are marked as red also. And only when the functional component is correctly connected, the component will turn green (the 4-pin one in Figure 10 (d)), otherwise, it remains red (the 2-pin one).

EXAMPLE USE CASES

Our design and fabrication work flow can be potentially adapted to different contexts. Here, we selectively demonstrate four use cases including decorating existing objects with lights and sound (Figure 11 (a)), creating articulating components for interactions (Figure 11 (b)), sensing and reacting to the status of the objects (Figure 11 (c)), and leveraging the movements of the objects to trigger activities (Figure 11 (d)).

A Candle Holder Celebrating Holidays. In this case, we mainly add outputs components including a speaker and an LED array to the objects. While the speaker is playing Christmas music, the LEDS are lighted up successively from top to bottom. The electronic features and the original function of the candle are combined seamlessly to create a joyful atmosphere. Also, this example demonstrates that our design tool can generate organized conductive paths even if the components are cluttered up on the object.

A Status Pointer. As shown in Figure 11 (b), by touching on the capacitive button, the servo motor rotates the pointer to different labels. A user can inform other people about his/her status by rotating the pointer to a certain label. By using Plain2Fun, the user not only can use static components like speakers and screens, but also can use moving components like motors, since the housing base can provide strong mechanical support as well as stable electrical connections. Also, the user can extend the object interface easily by adding more passive markers together with the components.

A Mug with a Temperature Display. We create an interactive cup (Figure 11 (c)), which has three LEDs and a temperature sensor. The LEDs indicate the temperature of the water inside the cup, e.g., more LEDs will be lighted up with a higher temperature. Since Plain2Fun is additive and noninvasive, the original function of the cup remains intact.

An Electronic Sand Timer. In this use case, we showcase that the input and output components work together to deliver new functions. We build an electronic sand timer (Figure 11 (d)) on a wet wiper can with an accelerometer, an OLED screen, a speaker and a capacitive button. The user presses the button to set time, and turn the can up-side-down to start counting down. This electronic sand timer provides two types of input: touching and movements of the objects. In this use case, we showcase that the input and output components work together to deliver new functions.

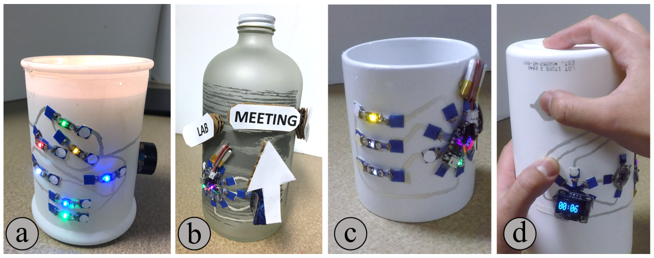

Figure 11. Example use cases: a candle holder with a surface mounted Christmas tree (a), a status indicator on a bottle (b), a temperature display for a coffee cup (c), and a digital sand timer (d)
.

PRELIMINARY USER STUDY

We conducted a preliminary user study to evaluate the usability and the expressiveness of our system.

Procedure

We invited 6 (5 male) participants (22 - 28 years old) with 5 engineering students and 1 student from non-engineering fields (economy). One of the participants had tried circuit prototyping with micro-controllers, while the rest had no experience in such activities. Four participants had no prior knowledge of CAD softwares for circuit design or 3D modeling. Our user study consisted of two sessions, design session and physical assembly session. Before the study, we introduced the overall concept of Plain2Fun with the example use cases, and demonstrate how to use the design tool and how to attach the housing bases onto the objects. After the two sessions, the participants were asked to fill out a questionnaire to document their experiences. Overall, the study lasted about 1.5 hr for each participant.

Design session. Before this session, we gave users a 20 min. tutorial on the UI and basic interactions. Then each user spent $\sim$ 15 min. to practice with the design software. In this session, we designed small three tasks to evaluate the usability of our design software and elicit the user experience of designing circuit on 3D surface. Namely, (**T1**) designing a circuit layout with a MCU and five four-pin components, (**T2**) designing one with a MCU, a six-pin component and three four-pin components, and (**T3**) freely choosing any components to form a functional circuit. We record the completion times for the three tasks separately. **T1** and **T2** are two most complex circuits supported by our current component library. For **T1** and **T2**, we provided two suggested connection strategies which are similar to Figure 3 (c) as references. To complete **T3**, we encouraged users to explore the component library and customize their own interactive functions. Further, the circuit layout in **T3** was designed on the model of a user-chosen object and was used for the physical assembly session also.

Physical assembly session. In this session, we tested the performance of the drawing machine with users' circuit layouts and gathered their experience of the physical assembly process. Based on the design results from **T3**, we used the machine to construct conductive paths on the objects. Then we provided users with the corresponding electronic components which have been housed on the bases ahead and asked them to finish the physical prototype by attaching the bases onto the surfaces. After the users completed the prototypes, we examined the connectivities of the circuits, programmed the MCU based on the working logic from the users, and then checked the interactive functions.

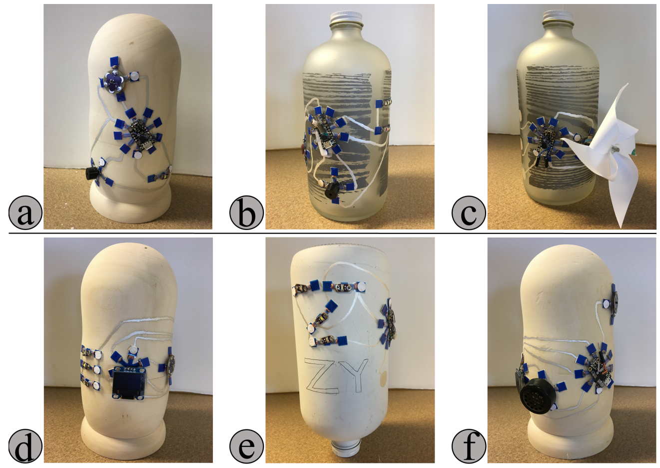

Figure 12. A collection of the physical prototypes from the user study.

Results

In the design session, all six users were able to find correct connecting strategies for **T1** and **T2** using the provided references. For **T1**, the average time was 7 min. 5 sec., and four users completed the layout within 6 min. The average time to complete **T2** was 5 min. 21 sec., where five users completed it within 6 min. The two users who had previous 3D modeling knowledge completed the circuits faster than the other users. Both of them took less than 5 min. We suspected they benefited from their familiarity with the 3D model manipulations. For **T3**, users came up with various designs and completed the circuits without any references. Users spent 10 to 15 min. on **T3**.

During the physical assembly session, none of the users had difficulties in attaching components at the right positions. All the users completed their interactive objects within 6 min. Then we measured the conductivities between the pins of the components and the conductive paths with a multimeter. The results showed that the each pair of the pins on all six objects was successfully connected and had a resistance below 5Ω. Figure 12 shows a collection of the customized functional objects made by users.

Findings

Overall, this study demonstrated the feasibility of Plain2Fun system to help a novice user to decorate objects with interactive circuits. Most of the users were able to quickly learn how to operate our design software and attach components and completed their interactive object within given time. Although we provided a limited library of components, users still created diverse functional prototypes as shown in Figure 12. The result from the post-study questionnaire (Figure 13) showed

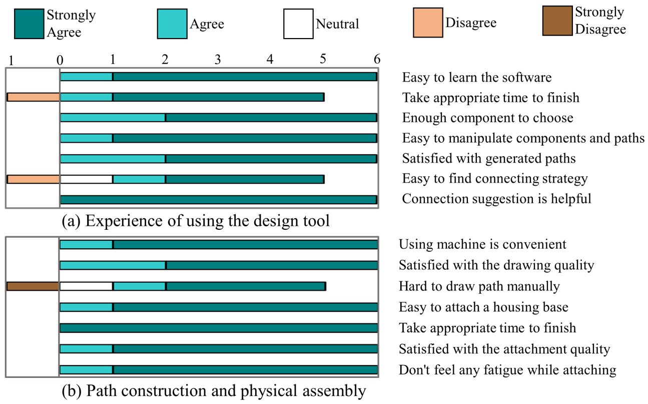

Figure 13. User feedback on design experience (a) and assembly experience (b).

that most of the users agreed that both of the designing and attaching components were easy for them. Below, we discuss the main insights we gained from our observation and the feedback from the users.

Software usability. Most users were satisfied with conductive paths generated automatically by the software (Figure 13 (a)) and preferred to use automatic path generation over manual routing. Although we let users to try both approaches during tutorial, only one user tried to manually draw paths by himself during the design session. This indicated the convenience and reliability of our automatic path generation. All users mentioned the connecting suggestions were useful during the layout designs. Only one user reported negative opinions on finding the connection strategy. In general, the preliminary results confirmed our design software's usefulness.

Connection robustness. Although we measured the conductivities of the circuits after their assemblies and the initial results indicated good connections, yet the connections were not robust especially during severe movements. For example, when we programed the MCU, we need to carry the objects around and plug/unplug the cables and a couple of connections were broken because of such movements. For these broken connections, we needed to manually enhance the contact points with the drawing pens. Overall, after we programmed the MCU and fixed some fragile contacts, all the objects behaved as the users designed. Currently the housing bases were merely for the prototypical implementation and we expect to improve the connection robustness by introducing a flexible PCB in our next iteration.

Design pipeline. The working physical prototypes as shown Figure 12 demonstrated that the users could realize their creative ideas with the help of Plain2Fun pipeline. Some users generated their ideas from their actual needs in real life. User 2 wanted an alarm clock that could wake her up at sunrise, so she built an alarm (Figure 12 (b)) which can be triggered by a light sensor. User 6 wanted to create a thermostatic environment for his pet. As a result, he built a temperature monitor (Figure 12 (f)) with a temperature sensor, a screen to display temperature value, and a speaker to ring alarms when it was too hot or too cold. Despite delivering the interactive functions, some users also considered their design from an aesthetic perspective. User 3 decorated an object with a good looking electric windmill (Figure 12 (c)). User 5 put four LEDs in an interesting position (Figure 12 (e). When the LEDs were off, the housing base of these LEDs looked like a letter "Z", and the LEDs looked like a letter "Y" when they were on. These two letters represented an idol of user 5.

Potential applications. Users unanimously agreed that repurposing the existing objects through Plain2Fun is an "*interesting and creative*" idea. User 3 acknowledged that *It is exciting that I could design my own object that works as I expected in such an easy way*". Moreover, we asked users to suggest potential applications after they used this system. Surprisingly, some users were able to propose many novel applications such as "*with some remote control modulus, make it an IoT product*"(User 4), "*Birthday gift. I believe a gift like this would be definitely impressive*" (User 5) and "*Children education and recreation*"(User 3).

LIMITATION AND FUTURE WORK

Circuit auto-planning. Although the connecting suggestion function in our software provides sufficient information to ensure users to connect each component correctly, the users still have to connect each pair of pins manually. This causes extra working load, consumes more time, and prevents user from devoting time into their creative thinking. Based on our observations during the user study, most users preferred to first import all the desired components at once and arrange components to the right position before they started to connect the pins. So for future improvements, we can embed the connecting strategy into a circuit auto-planning algorithm as discussed in the Topology Management section. This algorithm deals with all connections at once and optimizes the paths together. Furthermore, we can integrate graphic programming interface (similar to [3]) into our 3D circuit design tool thus include the programming into our pipeline. This will encourage more users who have limited programming skills to enjoy DIYing.

Flexible electronic components. In the component library, a soft housing base is used to bridge the hard straight breakout board and the curved surface. However, this triple layer structure (housing base - breakout board - functional component) makes the component bulky and often deteriorates the appearance of the object. For example, a temperature sensor chip has a footprint of $4mm \times 2mm$, the breakout board reaches $20mm \times 20mm$, and a housing base for this breakout board exceeds $40mm \times 40mm$. So for next step, we would like to redesign the housing bases using flexible PCB to reduce the size and improve its appearance. Also, the flexible PCB form factor increases the contact area and potentially improves the connection robustness. Furthermore, we can introduce more kinds of sensors and actuators into our library and create an ecology for the makers with massive production similar to CircuitScribe [7] and LiliPad [33].

Geometry of objects. The digital fabrication tool of Plain2Fun can adapt to objects with convex or slightly concave smooth cylindrical surfaces, such as easter eggs, russian dolls and wine bottles. However, this tool is not capable of operating on complex surfaces with holes, cavities or edges for the following reasons: (i) the holes and cavities may create occusions and cause defects on the laser-scanned model, (ii) complex structures, like a handle on a cup, may interfere with the conductive ink pen while drawing and (iii) conductive ink cannot stay firmly on sharp edges. To solve these problems, we can use multiple laser scanners to avoid occlusions, and add one more rotation degree of freedom to the conductive ink pen to fabricate on more complex shapes.

Real time registration. Unlike [20] using 3D printed mounts to fixate their objects on the clamps, we integrate the scanner with the machine to calibrate the position of the object while scanning. A major drawback of this method is that once the user removes the object, the previous calibration becomes invalid, since the user cannot put the object back to exact same position. Thus the user must re-scan and re-design the object if he wants to iterate the design with modifications. In the future, we can add a RGBD camera module to register the real object and the pre-scanned model in real time. With the real-time registration, the users can continue their design in a consistent coordinate system.

Durability. Users can freely tear off the housing bases without damaging the object. Further users can use organic solvents to erase the conductive paths. This way, users can reuse the original object and iterate their designs. Currently, the assembled interactive objects remain functional after a few days even. To extend the durability, we consider to use insulating paint to cover the conductive paths and glue the components after users finish their final design. Moreover, we can leverage color painting to further beautify the appearance.

CONCLUSION

In this paper, we presented Plain2Fun, a fast design and fabrication pipeline which allows users to add interactive functions to plain everyday objects with customized circuits layouts. To achieve this goal, we developed an interactive design software, a fully automatic digital machine for conductive path generation, and a library of components with soft housing bases for easy attachment. Plain2Fun allows users to customize the circuits directly on the scanned 3D model of the object. We validated that the digital machine with a conductive pen is capable of constructing low current circuits. Through multiple example use cases and a preliminary user study, we further verified that the physical prototypes made with Plain2Fun delivered the desired functions. We believe that a wide range of applications can be enabled by applying Plain2Fun and other novel ideas can be derived from this workflow.

ACKNOWLEDGMENTS

This work was partly supported by the National Science Foundation (MRI #1726865, IGERT #1144843, UNS #1512217) and the Donald W. Feddersen Chair Professorship from the School of Mechanical Engineering. Any opinions, findings, or recommendations presented are those of the author(s) and do not necessarily reflect the views of the funding agency.

REFERENCES

1. Adafruit. 2017a. Adafruit. (2017). Retrieved December 1, 2017 from = https://www.adafruit.com/.
2. Adafruit. 2017b. Adafruit Trinket M0. (2017). Retrieved December 1, 2017 from = https://www.adafruit.com/product/3500.
3. Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2017. Trigger-Action-Circuits: Leveraging Generative Design to Enable Novices to Design and Build Circuitry. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 331–342.
4. Autodesk. 2017a. Eagle. (2017). Retrieved December 1, 2017 from = https://www.autodesk.com/products/eagle/overview.
5. Autodesk. 2017b. Netfabb: Additive manufacturing and design software. (2017). Retrieved December 1, 2017 from = http://www.freelss.org/.
6. Xiang'Anthony' Chen, Stelian Coros, Jennifer Mankoff, and Scott E Hudson. 2015. Encore: 3D printed augmentation of everyday objects with printed-over, affixed and interlocked attachments. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 73–82.
7. CircuitScribe. 2017. CircuitScribe. (2017). Retrieved December 1, 2017 from = https://www.circuitscribe.com/.
8. Claudia Daudén Roquet, Jeeeun Kim, and Tom Yeh. 2016. 3D Folded PrintGami: Transforming Passive 3D Printed Objects to Interactive by Inserted Paper Origami Circuits. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, 187–191.
9. EngineerZone. 2015. What is a Breakout Board? (2015). Retrieved December 1, 2017 from = https://ez.analog.com/community/university-program/blog/2015/09/28/what-is-a-breakout-board.
10. Freelss. 2017. Freelss: The Free 3D Printable Laser Scanning System. (2017). Retrieved December 1, 2017 from = https://www.autodesk.com/products/netfabb/overview.
11. Wei Gao, Yunbo Zhang, Diogo C Nazzetta, Karthik Ramani, and Raymond J Cipra. 2015. RevoMaker: Enabling multi-directional and functionally-embedded 3D printing using a rotational cuboidal platform. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 437–446.
12. Daniel Groeger and Jürgen Steimle. 2018. ObjectSkin: Augmenting Everyday Objects with Hydroprinted Touch Sensors and Displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 134.
13. Liang He, Gierad Laput, Eric Brockmeyer, and Jon E Froehlich. 2017. SqueezaPulse: Adding Interactive Input to Fabricated Objects Using Corrugated Tubes and Air Pulses.. In *Tangible and Embedded Interaction*. 341–350.

14. Scott E Hudson. 2014. Printing teddy bears: a technique for 3D printing of soft interactive objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 459–468.

15. Yoshihiro Kawahara, Steve Hodges, Benjamin S Cook, Cheng Zhang, and Gregory D Abowd. 2013. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 363–372.

16. Yuki Koyama, Shinjiro Sueda, Emma Steinhardt, Takeo Igarashi, Ariel Shamir, and Wojciech Matusik. 2015. AutoConnect: computational design of 3D-printable connectors. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 231.

17. Mark Lanthier, Anil Maheshwari, and Jörg-Rüdiger Sack. 1997. Approximating weighted shortest paths on polyhedral surfaces. In *Proceedings of the thirteenth annual symposium on Computational geometry*. ACM, 274–283.

18. Brice Le Borgne, Olivier De Sagazan, Samuel Crand, Emmanuel Jacques, and Maxime Harnois. 2017. Conformal Electronics Wrapped Around Daily Life Objects Using an Original Method: Water Transfer Printing. *ACS applied materials & interfaces* 9, 35 (2017), 29424–29429.

19. Hanchuan Li, Eric Brockmeyer, Elizabeth J Carter, Josh Fromm, Scott E Hudson, Shwetak N Patel, and Alanson Sample. 2016. PaperID: A technique for drawing functional battery-free wireless interfaces on paper. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5885–5896.

20. Jingyi Li, Jennifer Jacobs, Michelle Chang, and Björn Hartmann. 2017. Direct and immediate drawing with CNC machines. In *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication*. ACM, 11.

21. Joanne Lo and Eric Paulos. 2014. ShrinkyCircuits: sketching, shrinking, and formgiving for electronic circuits. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 291–299.

22. Joanne Lo, Cesar Torres, Isabel Yang, Jasper O'Leary, Danny Kaufman, Wilmot Li, Mira Dontcheva, and Eric Paulos. 2016. Aesthetic Electronics: Designing, Sketching, and Fabricating Circuits through Digital Exploration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 665–676.

23. Simon Olberding, Sergio Soto Ortega, Klaus Hildebrandt, and Jürgen Steimle. 2015. Foldio: Digital fabrication of interactive and shape-changing objects with foldable printed electronics. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 223–232.

24. Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. 2013. Touch & activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 31–40.

25. Pololu. 2017. (2017). Retrieved December 1, 2017 from = https://www.pololu.com/product/1182.

26. Jie Qi and Leah Buechley. 2014. Sketching in circuits: designing and building electronics on paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1713–1722.

27. Raf Ramakers, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2016. Retrofab: A design tool for retrofitting physical interfaces using actuators, sensors and 3d printing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 409–419.

28. Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 483–492.

29. Valkyrie Savage, Colin Chang, and Björn Hartmann. 2013. Sauron: embedded single-camera sensing of printed physical user interfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 447–456.

30. Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. 2015. Makers' Marks: Physical markup for designing and fabricating functional objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 103–108.

31. Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 579–588.

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

33. SparkFun. 2017a. LilyPad. (2017). Retrieved December 1, 2017 from = https://www.sparkfun.com/categories/135.

34. SparkFun. 2017b. SparkFun. (2017). Retrieved December 1, 2017 from = https://www.sparkfun.com/.

35. Tung D Ta, Fuminori Okuya, and Yoshihiro Kawahara. 2017. LightTrace: auto-router for designing LED based applications with conductive inkjet printing. In *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication*. ACM, 3.

36. Alexander Teibrich, Stefanie Mueller, François Guimbretière, Robert Kovacs, Stefan Neubert, and Patrick Baudisch. 2015. Patching physical objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 83–91.

37. Thingiverse. 2017. Thingiverse. (2017). Retrieved December 1, 2017 from = https://www.thingiverse.com/.

38. Nobuyuki Umetani and Ryan Schmidt. 2017. SurfCuit: Surface-Mounted Circuits on 3D Prints. *IEEE Computer Graphics and Applications* 38, 3 (2017), 52–60.

39. Nirzaree Vadgama and Jürgen Steimle. 2017. Flexy: Shape-Customizable, Single-Layer, Inkjet Printable Patterns for 1D and 2D Flex Sensing. In *Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 153–162.

40. Voxel8. 2017. Voxel8. (2017). Retrieved December 1, 2017 from = https://www.voxel8.com/.

41. Eric Weisstein. 2017. Circular Permutation. (2017). Retrieved December 1, 2017 from = http://mathworld.wolfram.com/CircularPermutation.html.

42. Michael Wessely, Theophanis Tsandilas, and Wendy E Mackay. 2016. Stretchis: Fabricating highly stretchable user interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 697–704.

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

44. Sang Ho Yoon, Ke Huo, Yunbo Zhang, Guiming Chen, Luis Paredes, Subramanian Chidambaram, and Karthik Ramani. 2017. iSoft: A Customizable Soft Sensor with Real-time Continuous Contact and Stretching Sensing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 665–678.

45. Sang Ho Yoon, Yunbo Zhang, Ke Huo, and Karthik Ramani. 2016. TRing: Instant and Customizable Interactions with Objects Using an Embedded Magnet and a Finger-Worn Device. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 169–181.

46. Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 1–14.