Banksybot: CNC Decoration of Existing Objects with Physical and Virtual Authoring Tools

Jingyi Li, Michelle Chang, Daniel Lim, Valkyrie Savage, Björn Hartmann
University of California, Berkeley
{noon, michang, limdan7}@berkeley.edu, {valkyrie, bjoern}@eecs.berkeley.edu

ABSTRACT
Design tools for digital fabrication open allow creating *new* artworks, toys, and tools. However, computer numerical control (CNC) machines may also be used to *modify* existing objects. We investigate how to assist artists with *sgraffito*—reinterpreting pre-existing objects with engraved or drawn surface decorations. In our Banksybot system, an artist 3D-scans an object, uses our design tool to author decorations, and applies them to the object with a CNC machine. We describe and compare three methods for authoring designs: 1) fully virtual, using a 3D drawing program to prepare a design offline; 2) "live" virtual, in which virtual art is replicated by the machine in real-time; and 3) physical proxy, in which users physically draw on a 3D-printed copy of the object. Each mode has unique affordances, like undo/redo, reflection-in-action with the piece, or working tangibly at a different scale. Designs are executed using a modified consumer 3D printer and a commercial 3-axis CNC router. We contribute methods to generate 3D-printed mounts, segment strokes to maximize decoration quality, create toolhead routings, and generate machine code for artist-created designs. We created several example objects with Banksybot and collected informal feedback from five artists who used our authoring methods.

Author Keywords
digital fabrication; design tools; creativity support

ACM Classification Keywords
H.5.2 User Interfaces: Digital Fabrication

INTRODUCTION
While digital fabrication allows for tremendous flexibility in creating *new* objects, existing objects have inherent functional, historical, aesthetic or emotional value that cannot simply be duplicated or fabricated from scratch. These objects exhibit a plethora of material types, shapes and qualities (see objects in Figure 1): this physical variety is not yet matched by current digital fabrication techniques, which can only process a few types of raw materials.

We aim to assist artists and makers in creating *sgraffito* works combining existing objects and digital practices. Found objects

Figure 1. Objects processed using Banksybot: a clay fish sculpture with a programmatic scale decoration, a piece of driftwood engraved with locations of personal importance, a wax skull decorated in the style of sugar skulls, and a wooden model train carved to be an alligator.

augmented with new designs were hallmarks of artists such as Picasso and Braque; today we see similar work from graffiti artists like Banksy. Others blend digitally fabricated and found components like Inomata's turtle shell [6] and Horrigan's cups [5]. Research has likewise explored this combination, as in Zoran's "Hybrid reAssemblages" [23] and Chen's Encore [2] (see Figure 2). We take inspiration from their work and contribute design software and hardware to make sgraffito art accessible to a variety of practitioners.

In this paper, we focus on tools to help artists create surface designs—both engraving and drawing—on objects with arbitrary shapes. Our work generalizes the approach of existing plotting and drawing machines, which are restricted to either planar objects (e.g., plotters); ellipsoids (e.g., Eggbot[1]); or cylinders (e.g., laser engraving with rotary attachments). We also explore three different authoring methods leveraging both physical and digital methods of specifying a design.

We introduce Banksybot: a pipeline for creating engravings on existing objects using physical and digital specifications. Artists first 3D scan the object they wish to decorate. Banksybot then allows them to specify their decoration pattern in one of three ways. First, artists can *virtually* decorate their object in a browser-based 3D drawing interface, which allows integration of existing art, and affords undoing. Second, artists can work in *"live"* mode, in which designs are digitally authored but committed to the object in near-real-time as the artist works, allowing *reflection-in-action* as the artist reacts to the developing piece [14]. Finally, artists can work with a *proxy* object: after scanning the original, the artist prints a

[1] `http://egg-bot.com`

in preparation for submission to DIS 2017

Figure 2. Zoran combines pottery shards and fabricated shells for reassembly (left). Inomata 3D prints a turtle shell shell with new textures (center). Horrigan engraves cups and adds gold leaf (right).

surrogate—which may be scaled up or down—on which they physically draw their design.

Once the design is authored, Banksybot generates toolpaths and machine code for engraving or painting on the original object. Since precise positioning is important for accurate reproduction, Banksybot generates 3D printable mounts through constructive solid geometry (CSG) operations to fixture objects in a known position and orientation. The system also leads the artist through a multi-point calibration process to align physical and virtual coordinates. Some decorations may require the object to be reinserted in multiple orientations so all points can be reached; Banksybot automatically segments artist strokes to be drawn with maximum fidelity.

To demonstrate our pipeline's compatibility with consumer hardware, we modified a low-cost 3D printer with a swappable tool holder, and one rotational and two linear degrees of freedom. We also demonstrate our approach with a 3-axis CNC router. Both have limitations in the types of surfaces they can reach, but these could be overcome with industrial 5-axis machines in the future.

We contribute the following:

- A novel process which allows artists to add digitally-fabricated surface textures and images to existing objects
- Banksybot, a system which implements this pipeline
- An evaluation of three authoring techniques for this pipeline: virtual, live, and proxy
- A collection of author- and artist-created objects processed using Banksybot, demonstrating the wide variety of objects that can be modified with our tool.

RELATED WORK

Banksybot draws on prior research in techniques for combining digital fabrication with existing objects, as well as artist support tools and physical authoring tools for digital designs.

Digital Fabrication + Existing Objects

3D prints have been explored as a tool for *assembling* [10] or *rebuilding* [23] existing objects. In Encore, Chen, et al. [2] explored 3D printed enhancements to existing objects. Their tool allows for printing additions on the surface of existing objects, which is similar in spirit to the Banksybot project: giving new life to objects with existing desirable qualities. However, rather than single-point 3D printed augmentations, our project focuses on surface engraving.

Shader Printer [12] allowed creating images on arbitrary object surfaces using a laser and bi-stable thermochromic ink, but this requires coating the object and only permits flat, two-color decorations. Similarly, computational hydrographic printing [21] allows colors to be transferred onto non-porous surfaces; Banksybot additionally allows engravings, without requiring objects to be submerged in water.

Physical Authoring Tools for Digital Designs

Physical input for manipulating 3D models has seen significant attention in prior work. Knep's dinosaur input device leveraged a physical kinematic model for authoring animations [7], and Hinckley's neurosurgical props leveraged surgeons' physical experiences handling skulls [4]. ModelCraft allows architects to specify CAD operations on a paper printout of a model [16], and Sheng, et al., built a system for artists leveraging a sponge as a proxy for virtual clay [15]. Our *proxy* mode allows users to use a solid 3D printed version of the target object to design surface engraving.

Some prior work has also looked at real-time interactions with digital drawing and fabrication: Constructable allows users to draw on a workpiece with laser pointers to specify cuts [9], while Flying Pantograph tracks a user's stylus as they draw on a tablet [8]. However, these are both 2D interactions, and the artist cannot hold the piece in-hand while authoring their designs. Interactive fabrication experimented with a variety of techniques for creating new fabricated artifacts [18]; our aim is to integrate existing objects.

ReForm's bi-directional fabrication [17] and Makers' Marks [13] use physical authoring to build up shapes from scratch. In contrast, we compare different types of physical and digital authoring tools, and focus on reuse of existing objects.

Artist Support Tools

A variety of tools aim to support artists: in painting using projectors [3], in sketching using sketch warping [20], in sculpting with projected feedback [11], or in carving with smart tools [22]. In contrast to this prior work, Banksybot explores several authoring tools for artists to express themselves in a variety of media, rather than correcting or guiding their actions towards some pre-specified goal.

USING BANKSYBOT

To describe user interactions with Banksybot, we follow an artist as they create a series of engraved objects. Our artist scans each object they want to decorate: a sculpted clay fish, a cast wax skull, and a piece of wood shaped like Michigan (see Figure 3a).

Virtual Mode

The artist loads the fish's scanned geometry into their design tool. They import a vector graphic of a scale pattern and experiment with different placements on the model before committing to a particularly interesting layout that warps with the curvature of the fish.

Figure 3. To use Banksybot, an artist first selects and scans an object to decorate (a). They can drag and drop images in a virtual authoring tool (b), draw in live mode to commit the art in near-real time to the piece (c), or print an enlarged version of the object for physical sketching (d).

When they hit “done”, Banksybot generates mounts to orient the fish and fix it in place. After they are 3D printed, the artist carefully fixtures the mounts to and places fish inside their machine. Banksybot then leads the artist through a calibration process, where they select a point on-screen, then manually move the machine’s toolhead to that point on the physical object plus a specified “inset,” several times in a row. With this calibration complete, the system engraves the fish scales (see Figure 3b).

Live Mode

After mounting and calibrating the skull in the machine, the artist places a flower vector on the side of its head using the virtual interface. They watch as the machine engraves the design on the skull, and notice a small chip next to one of the eyes. Liking the effect, they draw more eyelashes around both eyes to match. The artists then decides more patterns on the piece will increase visual pop, and doodles over the empty surface as the machine follows them (see Figure 3c).

Proxy Mode

Our artist would like to engrave hand-drawn sketches on the piece of wood. Since they don’t want to mar the surface with pigment, and since the wood is too small to easily draw on precisely, they 3D print an enlarged copy.

Using a marker, the artist draws on the proxy print: a house over their hometown of Ann Arbor, a mortarboard over their school in Kalamazoo, and a tree near Traverse City where they got married. They scan the proxy object, then extract the drawn paths using the Magic Wand in Photoshop. Banksybot determines the 3D locations of these annotations on the model, and again generates a pair of mounts, leads the artist through calibration, and executes the design (see Figure 3d). This time, to engrave the wood more deeply, the artist uses a tabletop CNC router.

BANKSYBOT: A PIPELINE FOR CNC DECORATION

The Banksybot software takes a 3D-scanned object (as either an STL or OBJ file[2]) as input and produces machine instructions for decoration as output. Important components are reachability analysis, authoring tools, machine code generation, mount generation, and calibration routines for CNC machines. The workflow is slightly different for each authoring technique (see Figure 4). Many different CNC machines can be used to execute the decorations, and we also present technical specifications for our own three-axis CNC decoration machine.

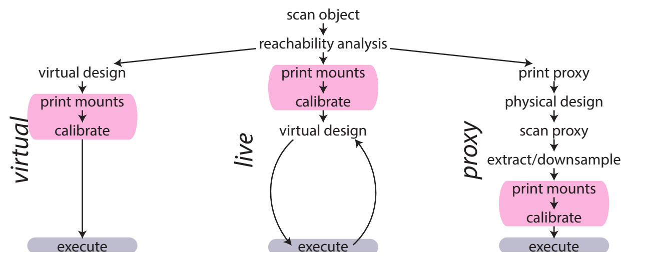

Figure 4. The workflow for our system always begins with scanning an object and performing reachability analysis, and ends with design execution. However, each authoring tool uses different intermediate steps.

Figure 5. Some surfaces cannot be reached to perform decoration. In our virtual editing tools, these are rendered in pink.

Reachability Analysis

Due to machine constraints, some surfaces of some objects may not be reachable by the tool head. For example, closer parts of an object may obscure and block access to parts behind them. We calculate reachability by raycasting: for each $1mm$ vertical step, we cast rays $1mm$ in diameter (roughly the size of our smallest tool) towards the core of the object. If the ray intersects the object’s surface multiple times, we mark triangles that are not the outermost intersection as unreachable (see Figure 5) and display them in pink.

Authoring Tools

Virtual Authoring

Our browser-based virtual authoring tool displays the artist’s scanned object geometry overlaid with sketched virtual lines. The current prototype, implemented in three.js[3], supports drawing and erasing functions, SVG import, and multiple colors and tools. Imported graphical paths are subsampled into a series of points ($.5mm$ apart), which are cylindrically mapped

[2] `https://en.wikipedia.org/wiki/Wavefront_.obj_file`

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

onto the object's surface. Users define the bounding box of the projection, allowing for translation, rotation, and perspective transformation. The drawing tools can also directly edit image points. Using multiple tools segments drawings into multiple toolpaths, which are supported through layers.

Live Authoring
In live mode, artists use a similar interface, but instead of creating a full design offline, the machine executes the design to the object as the artist continues working. Every 10 seconds, we export a tool path containing points that have not been edited for at least 10 seconds. In our prototype, we "Wizard of Oz" this interaction by saving the points to a folder shared between the computer running the design tool and the computer connected to the machine, and manually loading and executing the file on the machine. Points are color-coded to indicate their state: dark grey points have already been sent to the machine for execution and are fixed, while white points are still editable by the artist.

If users create art which cannot be executed in the object's current orientation, the system asks to either reorient the object or delay execution of those strokes. Once the object is reoriented, the uncompleted points are routed and sent to the machine.

Proxy Authoring
For proxy mode, artists work with a 3D printed copy of an object, which may be scaled from the original. They draw directly on this copy using whatever tools they prefer and scan the annotated proxy.

After selecting the appropriate pixels from the OBJ file's texture images, users export a corresponding binary mask representing visible annotations. Using mapping data contained in the OBJ file alongside these mask images, Banksybot calculates the 3D locations of all annotations, expressed as a collection of points, which are then imported to the virtual software for further editing. As extracted points are located on the surface of the proxy object rather than the original, they are re-scaled to fit the original.

Machine Code Generation
Once artists complete their designs, our software converts the art into GCode, compatible with our custom maker-class machine, CNC routers, and other machines.

The GCode represents a toolpath that the engraver, pen, or paintbrush follows while performing the decoration task. To minimize decoration time, we wish to construct a short routing and thus model our artist's design as a traveling salesman problem ("TSP"); this is similar to prior work in generating continuous line illustrations from images [1, 19]. For virtual and live modes, artist "strokes" are defined as a set of points $p_1 \dots p_n$ between mousedown and mouseup events. We model each stroke as a node in a fully-connected graph, with edge weights determined by 3D move distance (including avoiding the object) between the end point of one stroke and the beginning point of another.

Because we cannot denote "strokes" in the same way on scanned proxy objects, we use the extracted pixel-based points as nodes. Since we extract one point per triangle in the mesh, even small designs can end up with thousands of points, resulting in a very long-running TSP. We thus subsample designs to be no finer than our machine can execute, collapsing points within .4mm of each other.

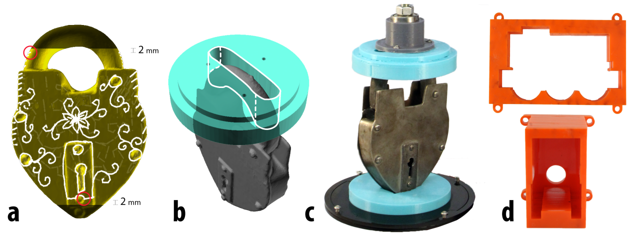

Figure 6. To make mounts, we select the extreme upper and lower points in a decoration orientation and add a small offset for tool size (a), then subtract a projection of the intersection of the object and mount from the mount itself (b) to ensure concave objects may be inserted (c). Our CNC-compatible mount is laser cut and features screw holes for mounting (d).

Once we have simplified our search space, we need to determine which orientation(s) are necessary for engraving, which depends upon the machine's available axes. To ensure decoration quality, we keep the tool as close as possible to normal during decoration (see Figure 8, left). We currently support two orientations: one along the object's major axis, and one along its minor axis (calculated by an ellipsoidal approximation). For each point, we take the minimum of the dot product of its normal and the two object axes to determine which strokes will be executed in which orientation.

Finally, we solve the segmented TSP. Our generated GCode consists only of move (`G1`) actions, which trace the strokes and retract the toolhead back to the X origin to move between them.

Mounts and Calibration
Since Banksybot supports use of existing objects and has artists hand-mount tools like pens—which may be set to different depths per run—we require calibration to ensure that our generated toolpath correctly matches the object's physical positioning. We use two techniques for this: custom 3D printed mounts which fix the object in place, and user-specified calibration points to determine skew and orientation.

Mounts
For physically fixing the object in a predictable position, one pair of 3D printable mounts for each orientation. The mounts need to (1) provide maximum stability, (2) allow for objects to be inserted and removed with ease, and (3) expose maximum surface area for decoration.

We thus create mounts that cradle as much of the object as possible without interfering with engraving. For each orientation, we select the extreme upper and lower points in the routing and create an offset of $2mm$ to account for the width of the tool itself (see Figure 6a). We create top and bottom blocks covering the object up to those points, and subtract the object's geometry from the blocks.

To ensure that the object can be inserted and removed (particularly if it is non-convex), we use projection. We project the widest horizontal cross-section of the intersection of the object and its mount, such that if an object has a non-convex area it can still be inserted and removed smoothly (see Figure 6b & c). For our tabletop CNC router, we only generate a bottom mount;

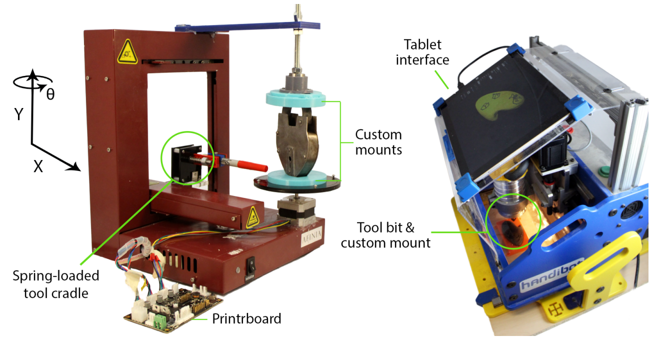

Figure 7. Our custom CNC decorating machine (left) uses repurposed servos and an open-source control board. It has spaces to mount 3D printed object holders to ensure object stability, and our spring-loaded tool cradle can accept engraving, drawing, or painting tools. We also use a HandiBot (left) to engrave in 2.5D.

we also add screw-holes to affix the mounted object to a board during the run (see Figure 6d).

Calibration Points

Mounts account for object position, but cannot account for initial conditions, particularly the hand-inserted decoration tool's depth. For maximum adaptability to existing materials and to offset inaccuracies in the scanning process, we allow users to specify how deep points should be pushed inward. This helps specify the CNC machine's stroke width, and our custom machine has a spring-loaded toolhead to absorb extra force from the additional surface contact.

Artists physically calibrate the machine by matching a small set of virtual and physical points on their object (minimum 4 non-co-planar points, with accuracy increasing for each additional point). We capture the physical coordinates for each point, and use least-squares matching to generate a 3D transformation matrix mapping digital to physical points. We apply this matrix to each digital point in the GCode before execution.

A Low-Cost CNC Sgraffito Machine

Our primary goal is to contribute a collection of software tools supporting artists with sgraffito. However, as current consumer CNC machines are largely focused on 3D printing, we also demonstrate how to construct a three-axis decoration machine from common consumer 3D printer parts. We require a machine that can address the entire surface of an object, allows swappable toolheads, has mount points for our generated mounts, and is robust to scanning inaccuracies in an object's captured geometry.

Our machine uses two linear axes and one rotational axis. It is created from the motorized stage of an Afinia 3D printer and an extra stepper motor. We use an open-source PrintrBoard for control (see Figure 7). We attach each tool—a diamond engraver, fine-tipped sharpie, or paintbrush—to a custom tool holder. The machine accepts generic GCode which is interpreted by our modification of Marlin[4], a 3D printer firmware library.

Achievable Accuracy

Several sources of inaccuracy can affect decoration: the 3D scanner, the 3D printed mounts, and the actual machine. Our

[4]`https://github.com/Printrbot/Marlin`

Figure 8. The smallest reasonably achievable line spacing with Banksybot is $1.6mm$ (left). We try to keep the tool's angle of attack as close to normal as possible, as the pen moves off the object when closer to the sphere's poles (right).

scanner claims $.127mm$ ($.005in$) accuracy; however, when we performed 15 pairwise distance measurements on our skull object, the scan is on average $.54mm$ offset from the original object. When printed, it is $.58mm$ thicker than the scanned object. The 3D printer we use to fabricate our mounts, a Stratasys uPrint SE Plus, has a layer height of $.254mm$ ($.010in$) and a minimum XY feature size of $1.194mm$. Our machine's spring-loaded tool holder can deflect $0.73°$ on the X and $0.29°$ on the Y axis when dragged across the surface of an object. The tip of our pens cannot make a dot smaller than $0.95mm$, and our rotational stepper makes steps of $1.8°$.

To explore accuracy limitations in practice, we created a couple of simple reference objects with hatchmark patterns of various curvatures (see Figure 8). More engineering could improve our machine's accuracy in the future: rather than constructing the most accurate possible machine, we aimed to create a simple one capable of demonstrating the Banksybot pipeline.

EVALUATION

We describe a qualitative evaluation comprising an informal user study comparing our authoring tools, as well as a collection of author-created objects demonstrating Banksybot's flexibility.

Informal User Study

To evaluate our authoring tools, we invited five artists (ages 20–46, thee fremale) to our lab and had them create an object with each tool. They came from a range of artistic practices: two digital, two new media, and one material. Artists had an average of 3.5 years of experience with digital fabrication, though we did not select for this.

For ease of comparison, each artist created the same object using the same tool: in virtual mode they decorated an antique padlock, in live mode they decorated a Japanese Kokeshi doll, and in physical proxy mode they decorated a rubber ducky (see Figure 9). After giving a brief overview of the system, we randomized task performance order. All 3D scanning was performed by the authors before and after the studies to allow the artists to focus on the authoring tools. Due to the time involved in printing mounts after creating a design, we executed the virtual and proxy designs after the artist left (the live design was executed as they drew).

A majority (3 out of 5) of users preferred virtual mode, citing favorable affordances of erasing/undoing and importing vector

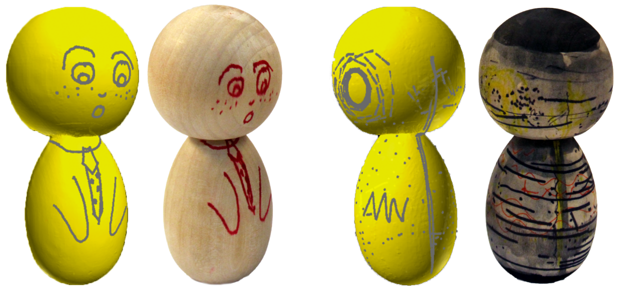

Figure 9. Study users created digital (yellow) designs for Kokeshi dolls in live mode. Left is a literal decoration. Right is a performance piece, where the artist treated the "machine as a material," placing dots in the web interface so the machine would "spin for [its] master" as he made black marks to show the doll's interaction with both the machine and human.

art, which let them work with more familiar design tools like Adobe Illustrator. Even with the tangible aspects of proxy mode, users favored virtual mode because they "didn't have to worry about smearing" the pens or "awkwardly holding" the 3D print to draw at angles. Most users, despite rating live mode as most likely to be used in the future and the most fun to use (4.8 and 4.2 on a 5 point Likert scale, respectively), reported they felt they were in a "race against the machine" and disliked the "attention divide between the virtual and physical drawings." Our digital artists prioritized the ability to undo strokes, while our new media ones valued the hybrid digital/physical interactions with their objects.

In particular, one new media artist began to draw alongside the machine during live mode, "following a rule" to make dots when the machine did and lines when the machine rotated (see Figure 9, right). This hybrid performance highlights a novel interpretation of the CNC machine as a *material*, while still giving artists direct, iterative control of their work. While most participants enjoyed the tangibleness of 3D-printed proxies, they "could not imagine" the final result on the actual object due to material differences: one artist commented 6 times on and adapted her design to the "interesting topographies" of the print layers, but was unsure if it would carry over to the smooth rubber ducky.

Example Objects

We created a collection of example objects showcasing designs on natural, hand-shaped, hand-cast, and machine-made substrates. We elaborate on the objects described in *Designing with Banksybot* as well as a toy train.

Michigan Driftwood Pendant. The driftwood piece showcases proxy mode, allowing the artist to create a design that flows with the material's physical ridges. Engraved on our Handibot CNC router, it shows our pipeline is compatible with machines beyond our custom one, and validates our CNC mount design.

Clay Fish. Our hand-shaped clay fish was scanned in its leather-hard stage (i.e., when shrinkage due to drying is complete). We then engraved the hexagonal scale designs, which were programmatically generated using Processing[5], and finally baked the figure to set the designs. This object also highlights the importance of user-controlled "insets:" as the clay is not perfectly hard and can deform during mounting, it is critical that we move the tool in further than the initial scan dictates.

Día de los Muertos Wax Skull. Our hand-cast wax skull was engraved in live mode. Its design showcases the benefits of *reflection-in-action* as the artist notices a chip in the wax and reacts to it by filling in additional eyelashes.

Alligator Train. We engraved a wooden train engine to look like an alligator, and included the signature of the train owner (Alison) on top. This design highlights the utility of our generated mounts, as the train cars are on wheels and would simply roll away without mounting, and many of the functional benefits of using existing objects: the train cars have embedded magnets mixed with wood and plastic, which could not be 3D printed.

DISCUSSION AND CONCLUSION

The increasing availability of both 3D scanning and printing enable a whole range of novel physical → digital → physical design processes. We have begun to explore digitization as a tool for artistic expression, starting with the decoration of existing objects. Banksybot has artists use such objects as a literal part of a final composition—expanding the materials available for "digital fabrication" to include not just featureless stock material like filament or wooden "blanks," but objects with intrinsic functional or aesthetic value.

Through our initial user studies with artists, we uncovered many interesting features of our toolchain and authoring tools. Users saw our machine as an *interpreter* of their designs, and were excited about bringing more "chance" to their practice in "collaborative authorship," such as one user's hybrid performance of using the machine as a controller rather than decorator. Banksybot allows users to go beyond the standard of using a machine to exactly execute a digital design, opening spaces for more expressive, meaningful, and safe human-CNC interactions.

Our virtual authoring tools could also benefit from additional simulation. For instance, we currently display all lines as having uniform thickness. Our machine does not execute them as such: lines where the pen dwells longer are thicker than lines where it moves quickly. Likewise, artists modulate line thickness in pieces by varying tool pressure. While some combinations of tools and materials work well (e.g., diamond engraver and wax), others produce undesirable artifacts (e.g., markers bleeding onto wood). To aid in sgraffito, we would like a simulated or empirically-gathered library of appropriate parameter settings for these tool-surface interactions.

Furthermore, our current system only allows literal, direct authoring of lines. We would like to investigate *symbolic* authoring: for example, in proxy mode, users could draw a cross to denote a specific SVG to import, similar to Makers' Marks [13], or specify a perimeter line for a texture to fill.

In conclusion, we presented Banksybot: a system which allows artists to modify existing objects with fabricated surface art. We developed three unique authoring tools—virtual, live, and proxy—for design specification, and tested them with artists from a variety of practices. We hope to see more work combining digital fabrication and existing objects for artistry.

[5] `http://processing.org`

ACKNOWLEDGMENTS

This material is based on work supported by the NSF under Grant No. DGE 1106400 and a Samsung ThinkTankTeam award.

REFERENCES

1. Bosch, R., and Herman, A. Continuous line drawings via the traveling salesman problem. *Oper. Res. Lett. 32*, 4 (July 2004), 302–303.
2. Chen, X. A., Coros, S., Mankoff, J., and Hudson, S. E. Encore: 3d printed augmentation of everyday objects with printed-over, affixed and interlocked attachments. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, ACM (2015), 73–82.
3. Flagg, M., and Rehg, J. M. Projector-guided painting. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, ACM (2006), 235–244.
4. Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, ACM (New York, NY, USA, 1994), 452–458.
5. Horrigan, A. Amalgam collection. `http://cargocollective.com/anthonyhorrigan/Amalgam-Collection-2012`, 2012. Accessed: 2016-04-05.
6. Inomata, A. World outside your world. Video (3min. 2sec. / NTSC 16:9), 2011. Accessed: 2016-04-11.
7. Knep, B., Hayes, C., Sayre, R., and Williams, T. Dinosaur input device. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, ACM Press/Addison-Wesley Publishing Co. (New York, NY, USA, 1995), 304–309.
8. Leigh, S.-w., Agrawal, H., and Maes, P. A flying pantograph: Interleaving expressivity of human and machine. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, ACM (2016), 653–657.
9. Mueller, S., Lopes, P., and Baudisch, P. Interactive construction: Interactive fabrication of functional mechanical devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, ACM (2012), 599–606.
10. Mueller, S., Mohr, T., Guenther, K., Frohnhofen, J., and Baudisch, P. faBrickation: Fast 3d printing of functional objects by integrating construction kit building blocks. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, ACM (New York, NY, USA, 2014), 187–188.
11. Rivers, A., Adams, A., and Durand, F. Sculpting by numbers. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 157:1–157:7.
12. Saakes, D., Inami, M., Igarashi, T., Koizumi, N., and Raskar, R. Shader printer. In *ACM SIGGRAPH 2012 Emerging Technologies*, ACM (2012), 18.
13. Savage, V., Follmer, S., Li, J., and Hartmann, B. Makers' marks: Physical markup for designing and fabricating functional objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, ACM (2015), 103–108.
14. Schön, D. *The Reflective Practitioner*. Basic Books, Inc., New York, New York, 1983.
15. Sheng, J., Balakrishnan, R., and Singh, K. An interface for virtual 3d sculpting via physical proxy. In *GRAPHITE*, vol. 6 (2006), 213–220.
16. Song, H., Guimbretière, F., Hu, C., and Lipson, H. Modelcraft: Capturing freehand annotations and edits on physical 3d models. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, ACM (2006), 13–22.
17. Weichel, C., Hardy, J., Alexander, J., and Gellersen, H. 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, ACM (New York, NY, USA, 2015), 93–102.
18. Willis, K. D., Xu, C., Wu, K.-J., Levin, G., and Gross, M. D. Interactive fabrication: New interfaces for digital fabrication. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '11, ACM (2011), 69–72.
19. Wong, F. J., and Takahashi, S. Abstracting images into continuous-line artistic styles. *Vis. Comput. 29*, 6-8 (June 2013), 729–738.
20. Xie, J., Hertzmann, A., Li, W., and Winnemöller, H. Portraitsketch: Face sketching assistance for novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, ACM (New York, NY, USA, 2014), 407–417.
21. Zhang, Y., Yin, C., Zheng, C., and Zhou, K. Computational hydrographic printing. *ACM Trans. Graph. 34*, 4 (July 2015), 131:1–131:11.
22. Zoran, A., and Paradiso, J. A. Freed: A freehand digital sculpting tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, ACM (2013), 2613–2616.
23. Zoran, A., Shilkrot, R., Nanyakkara, S., and Paradiso, J. The hybrid artisans: A case study in smart tools. *ACM Trans. Comput.-Hum. Interact. 21*, 3 (June 2014), 15:1–15:29.