

shapeCAD: An Accessible 3D Modelling Workflow for the Blind and Visually-Impaired Via 2.5D Shape Displays

Alexa F. Siu¹, Son Kim², Joshua A. Miele³, Sean Follmer¹

¹Stanford University ²Vista Center for the Blind, ³Smith-Kettlewell Eye Research Institute

¹{afsiu, sfollmer}@stanford.edu, ²skim@vistacenter.com, ³joshua.a.miele@gmail.com

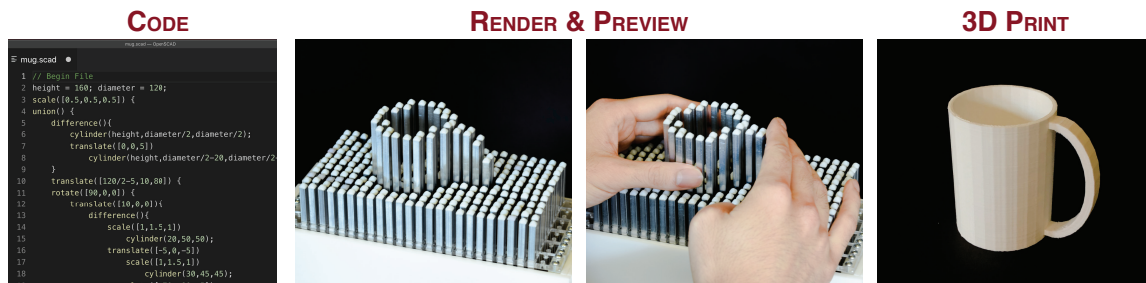


Figure 1. We investigate an accessible 3D modelling workflow (shapeCAD) where 3D models are generated through OpenSCAD, a declarative programming language for 3D modeling, and rendered at interactive speeds in a 2.5D shape display consisting of a grid of 12×24 actuated pins.

ABSTRACT

Affordable rapid 3D printing technologies have become key enablers of the Maker Movement by giving individuals the ability to create physical finished products. However, existing computer-aided design (CAD) tools that allow authoring and editing of 3D models are mostly visually reliant and limit access to people with blindness and visual impairment (BVI). Through a series of co-design sessions with three blind users of mixed programming ability, we identify accessibility challenges in existing 3D modelling scripting tools and design interactions to support dynamic feedback of scripts using a 2.5D tactile shape display. With these insights, we implement shapeCAD. Interacting with shapeCAD, BVI users are able to leverage the low resolution output from a 2.5D shape display to complement programming of 3D models. shapeCAD allows users to haptically explore and modify existing models, and to author new models. We further validate usability and user experience through an evaluation with five BVI programmers. In a short period of time, novices were able to design a range of new objects. BVI users can bring a valuable perspective to design and it is imperative to increase accessibility in tools that enable this community to also participate as designers.

ACM Classification Keywords

Human-centered computing Accessibility systems and tools

Author Keywords

Accessible Authoring Tools; Accessible 3D Printing; Tactile Graphics; Haptics; Tactile Displays; 2.5D Shape Displays

INTRODUCTION

People with blindness and visual impairments (BVI) are experienced *makers* having to be adept at using the technology at hand to solve accessibility problems they face in their daily lives. This spirit of creative problem solving and tinkering in the BVI community has mostly existed in parallel to the mainstream *Maker Movement* because most *maker* tools are inaccessible. The rise in less expensive and distributed fabrication tools, such as 3D printers and easy-to-use micro-controllers, have made it easier for a wide range of groups to engage in *making* [19]. Yet accessibility remains a challenge and the BVI community, which may benefit immensely from such tools, remains marginalized [22, 53]. Accessibility in *making* can not only provide access points for contextualized learning of many concepts considered critical for STEM but can also give BVI people the tools to participate in the vibrant maker culture as designers themselves [3, 8, 19, 53] and act independently to make the things they want and need. In this paper we focus on identifying and addressing accessibility challenges in the areas of 3D modelling and 3D printing for the BVI.

Current commercial and open source Computer-Aided Design (CAD) tools that support viewing, authoring, and editing of 3D models are mostly visually reliant and limit access for BVI people. This is a limitation not only in terms of the lack of feedback as models are created but also in terms of the user interface, as most modern programs rely on direct manipulation with graphical user interfaces which are not easily accessible. Programming-based tools for 3D modelling, such as OpenSCAD [40], address some issues of access for defining 3D models, yet feedback on the geometry of the model is not available [35]. 3D printing the model can serve as feedback itself, however, the time between iterations can take several hours; this discourages its use and methods to interactively visualize the resulting 3D model do not exist [24].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ASSETS '19, October 28–30, 2019, Pittsburgh, PA, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6676-2/19/10 ...\$15.00.

<http://dx.doi.org/10.1145/3308561.3353782>

While there has been some progress on 2D tactile arrays [14], which allow for full hand active haptic exploration, 2D representations have challenges in providing BVI people with easy understanding of 3D objects [48]. Tactile arrays have limited resolution, and information can only be extracted by a contour following procedure which is slower than when using a combination of various haptic exploratory procedures [30]. In comparison, 2.5D shape displays [13, 49, 50, 55, 62] also have tradeoffs with resolution, but can recover some of the objects' depth cues lost in 2D representations and may provide a promising approach to enable 3D object understanding.

In this work, we investigate using a 2.5D shape display to support feedback during declarative programming of 3D models. Figure 1 summarizes the workflow. Code is written to specify 3D geometry. When the code is compiled, it is rendered on a 2.5D shape display. The user can then interact with the rendered shape to verify the design and rapidly iterate on the code before 3D printing the design. In formulating this workflow, we use a participatory design process with two goals in mind: (1) to understand the needs and preferences of BVI people when using existing declarative programming tools, and (2) to design interactions with 2.5D shape displays that support easier navigation, understanding, and creation of 3D objects.

With the insights gained from the participatory design process we implement an accessible 3D modelling workflow: shapeCAD. With shapeCAD, we define interactions to navigate 3D models that allow users to obtain a detailed mental model of the object without getting lost, and interactions that support quickly iterating through code changes while obtaining haptic feedback using a 2.5D shape display. Last, we report on results from a usability evaluation conducted with four blind and one low-vision participant using shapeCAD to create 3D models. We were specifically, interested in (1) the speed with which new users can master the system to accomplish set goals, and (2) how effectively users can utilize the newly learned skills to design 3D models from scratch. Our results demonstrate that users are able to complete a set of simple 3D modelling tasks. We conclude with design considerations for increasing 3D modelling accessibility for the BVI community.

RELATED WORK

Audio & Tactile Graphics

BVI people most often rely on labels and alternative text. Several works have looked at automatic labeling of images [2, 18], spatial data such as maps [34] and even complex engineering drawings [44]. However, text-based descriptions of graphics are less precise, more error prone to interpret, and require more cognitive load than a perceptual interface that directly renders the same information through touch or vision [52, 57]. Tactile graphics have been used to complement text descriptions (e.g. thermoforming, embossing, 3D printing outlines, etc) [23], and extensive guidelines exist on best practices for communication efficiency and understandability [39]. These methods while accessible, are most successful at describing outlines and texture but fall short when communicating 3D spatial information. In practice, a one-to-one visual to haptic translation usually does not result in an efficient method as

has been shown not only for perspective drawings but also for other kinds of spatial information such as schematics [41], engineering drawings [44], and maps [11, 21].

3D printed objects are a promising possibility since a lot of the haptic multidimensional cues are preserved but these can take a long time to fabricate [54] and accessible tools for the blind to express through this medium don't exist [17]. An additional drawback is that they cannot benefit from the advantages of a computer-based system. Moreover, fabricating tactile graphics require high effort and technology-dependent knowledge, they restrict presentation of dynamic information, and most often require the presence of a support person to offer additional information or context typically through audio descriptions [32]. In this work, we seek to understand what are effective methods for encoding and communicating spatial 3D geometry information in accessible ways for BVI people.

Haptic Encoding & Perception

Several perceptual studies have proposed models on the visual-haptic perception and processing for common 3D objects. Humans are remarkably good in identifying objects through haptics [26] employing a combination of various exploratory procedures [30]. Lawson and Bracken compared identification of 3D objects with varying depth cues (from full 3D information to just 2D outlines) and found that identification was more accurate and took less time when more depth cues were available [29]. A limitation is that most of these perceptual studies have focused on 3D object identification; we highlight that identification is inherently different from understanding in that object identification can happen without having an accurate mental model of the 3D object in question. In the context of this work, we are more interested in 3D object understanding rather than identification since modifying an existing 3D model or even authoring one from scratch, requires the user understand all spatial elements of the model. In this work, we investigate interactions and perceptual cues that help users form an accurate mental model of a 3D object representation.

Dynamic Tactile Arrays & Shape Displays

Tactile arrays and shape displays, which typically consist of a grid of pins that can translate up and down, can provide dynamic rendering of graphical content and have been shown to be as effective as static tactile graphics in conveying the same information [32, 36, 41, 49]. These systems can provide functionality similar to a computer-based system. Compared to other types of haptic displays such as tool based kinesthetic haptic devices [33], tactile arrays provide many benefits for gross shape perception. Interacting with tactile arrays is typically done through whole hand interaction [29] which is not possible through a single point haptic display. However, tactile arrays and shape displays vary in their resolution and number of haptic cues they are capable of conveying. Some are only capable of binary heights while others have a broad range of travel. There are several arrays with high enough resolution for Braille presentation (10-17 taxels per inch) but with the trade-off of having a small rendering region limited to representing Braille characters and only allowing binary pin heights [42]. Other tactile arrays such as the Graphiti developed by the American Printing House have been developed with a larger

rendering region at the cost of more limited resolution [14]. Compared to Braille arrays, these tactile arrays allow rendering of contours and fills for graphic representations.

Last on the spectrum are shape displays, such as shapeShift [50] which sacrifice resolution at the cost of restoring depth cues. These can represent 2.5D data through reliefs but their resolution is typically lower than other tactile arrays with restricted depth cues [13, 50, 55, 62]. Tactile arrays will likely always have resolution and workspace limitations such that the content being explored will exceed the rendering region of the display. To overcome these limitations, researchers have explored interaction techniques that allow BVI people to access large-format graphics through tactile arrays [43, 45]. In this work, we seek to design interactions with 2.5D shape displays that enable BVI to explore 3D models.

Multimodal CAD & 3D printing

Several tools have been created to allow automatic translation from 2D to 3D format compatible for 3D printing [5, 27, 59]. However, these do not allow BVI people to preview the models or add customizations and are limited to flat geometry. Several tangible user interfaces to enable physical authoring have been proposed to allow more customization and a tighter loop with the design process [1, 25, 31, 37]. While the input provides a good preview of one's own model, it is not possible to preview or modify existing models on the web. Götzelmann proposed a system for autonomous selection of 3D models available online based on specified user preferences and parameters [16].

Declarative programming 3D modelling tools, such as OpenSCAD and CraftML[61], have more accessible input methods and are compatible with current screen reading technologies. OpenSCAD has reportedly been used by blind people in authoring 3D models [24, 35]; this method uses a technique called constructive solid geometry (CSG) where complex shapes are defined by applying boolean operations (e.g. union, difference, intersection) to primitive shapes (e.g. sphere, cube, cylinder) [28]. However, several challenges still remain. Methods to interactively visualize the code don't exist, other than audio from the screen reader. This means that users must remember the code and picture what it would look like, which demands high cognitive load and becomes exponentially more difficult with complex models [35]. Another option is 3D printing for each iteration, but this can take several hours; thus the long period in between iterations to verify users' designs, discourages its use [24] and creates a disconnect between the authoring method and the finished artifact [6]. Moreover, identifying errors and debugging the code usually entails debugging of the entire source code which may be burdensome and time intensive [35]. In this work, we investigate ways to overcome some of the programming challenges by using tactile feedback through a 2.5D shape display.

PARTICIPATORY DESIGN OF ACCESSIBLE 3D DESIGN

To uncover and formulate design guidelines on interactions with tactile displays that support understanding and creation of 3D geometry, we utilized a participatory design approach to position participants as co-designers.

Summary of Workflow

Figure 1 shows an overview of the investigated workflow combining declarative programming for specifying 3D model geometry and a 2.5D shape display for dynamic previewing of the authored content. Providing dynamic feedback as the user works through their program, allows the user to quickly verify, correct errors, and iterate on a design. As a programming language, we investigate using OpenSCAD for several reasons: (1) it is open-source and mainstream in the *maker* community for sharing parametrized designs that can be easily customized [38], and believe it is important to use tools that are currently available and beyond the research stage; (2) it is a declarative programming tool for 3D modelling and thus compatible with existing screen reading technologies [35] as opposed to many systems that rely on direct manipulation through graphical user interfaces which are not accessible.

To obtain a dynamic preview of the model specified through code, we use a tactile shape display based on shapeShift [50]. The shape display consists of a grid of actuated square pins (4.8 mm) with an inter-pin spacing of 2 mm. The pins are grouped in modules of 2x24 pins that can be combined to increase the display region. The average speed is 70mm/s, thus shapes can be rendered within a second.

Methods

We conducted one-on-one recurrent co-design sessions with three users. The main goals of the co-design sessions were to: 1) identify accessibility/usability issues in current practices using script-based tools for 3D modelling, 2) understand what interactions help users navigate and understand 3D models using a 2.5D tactile shape display with limited resolution, 3) design interactions that help users transition from programming 3D models to previewing them using a 2.5D tactile shape display, and 4) refine teaching and training practices related to the proposed 3D modelling workflow.

Demographics

We recruited three users through local blindness organization mailing lists who were interested in learning more about 3D design tools. All users used screen readers as their primary assistive technology when using a computer (U1, U2, U3). All users preferred operating system was Windows; and were comfortable with both JAWS and NVDA screen readers. None had previous 3D modelling experience. U1 had taken a Java class 10 years ago. U2 had no previous programming experience. U3 had worked as a software developer.

Structure of Sessions and Analysis

Each user participated in at least one session, one of them in four sessions. Each session lasted between 1.5 to 2 hours. Sessions were interleaved between users and separated by a period of 5-10 days. We used the data and feedback collected in each session to prepare for and implement new prototype interactions for subsequent sessions.

The first session was structured to (1) familiarize the user with the shape display hardware capabilities and limitations, (2) allow users to explore the hardware by browsing through existing 3D models, and (3) read and modify OpenSCAD script examples. Through this process, we identified what areas

were challenging to the user when interacting with the system. We asked and noted users' preferences, asking probing and clarifying questions. The session typically concluded using Wizard-of-Oz techniques [10] to quickly prototype some of the ideas that had emerged from the session. The feedback was then used to implement interactions that seemed promising and shown to the next user. Subsequent sessions focused more explicitly on new features added to the prototype. All sessions were audio-recorded, transcribed, and coded for analysis. These were then organized by common themes.

Findings & Challenges

Diverse Interests & Goals for 3D Modelling

All co-designers had personal motivations for learning about 3D modelling and wanted to help advance 3D modelling accessibility to give BVI makers more independence in their various professional endeavours. As U3 noted, one of the main hindrances for not being able to learn about 3D modelling was the reliance on sighted mediators, *"I can't do it independently, if I did, I would have to call [relative]."* U1 wanted to learn about 3D modelling to help launch his personal e-commerce business, *"something like this would definitely help to verify the product renderings. Instead of a blind person having to verbally describe how to draw a mockup, this would be a much more tangible way to get almost immediate feedback and then being able to say 'I don't like that' and keep changing it until you can say 'ok this is good'"*. U2 is a teacher and was interested in 3D modelling for creating maps and teaching materials for his students. He had mentioned how one his students had been interested in computer science but *"the lack of accessible material"* had discouraged the student and felt that blind people needed more tools to make materials accessible. These point to several needs and opportunities for creating domain specific design tools in the context of digital fabrication for the BVI community. For example, tools for fostering blind and sighted collaborations in prototyping and brainstorming sessions, creating electronics project enclosures, and designing teaching aids for BVI students.

Lack of Accessibility in OpenSCAD

Our initial workflow made use of the OpenSCAD graphical user interface. One of the first challenges identified when users were introduced to the OpenSCAD interface was that several of its parts are not accessible with both JAWS and NVDA screen readers. As U1 noted, the *"interface is not keyboard friendly, I can access the menu but I can't access the different panels."* Moreover, we also found that while users were able to render and compile the scripts, the error console was also not accessible so users were not able to identify if they had an error, instead they relied on a sighted mediator to check if the code had compiled and a model had been generated.

Dynamic Tactile Representation Supports Programming and Vice Versa

The investigated workflow abstracts low level 3D model specification through a declarative programming environment (a very precise language) and high level operations (e.g. scale, rotation, translation) through a coarse low resolution tactile display. We found these two polar mediums can compensate for each other's limitations.

Having an understanding of how a model is created, through the program script, helps the user form a mental model of the object and compensates for limited resolution output of the shape display. When the code is provided with the 3D model rendered in the shape display, users are able to more quickly identify its different parts, *"looking at code helps inform the image and vice versa"* (U1). Without the code, the low resolution output of the pins makes the task of identifying objects more difficult, as U3 noted, *"doesn't always feel like a 1:1 representation of the model. Things appear flatter than they are."* In a shape display, horizontal and vertical edges can be well represented but edges that don't align with the orientation of the pins will appear aliased. Thus, for example, the round edges of a cylinder will feel more square.

On the other hand, providing just the code can be a mentally demanding task for the user to understand and form an accurate mental model of the actual 3D geometry, *"definitely for a mind [of] design and detail if looking just at the code"* (U1). *"If I just had the file, I would probably just have some guess of what it would look like but it wouldn't be very accurate or I would forget as I read through it"* (U2).

All users felt very strongly that the immediate feedback supported their learning of the programming language, CSG operations, and also when debugging the scripts. U2 who had no previous programming experience when using the workflow felt this was the most powerful attribute, *"When you said programming, I was a bit scared but it wasn't so scary after you try it... The nice thing about it, when you change something immediately you see what happens on the display. That's a very powerful tool for understanding the code."*

Navigation Challenges

The workflow makes use of a 2.5D shape display to provide a dynamic preview of the 3D model. Since users are only able to feel one face of the object at a time, this requires creating navigation commands to fully explore the entire 3D model. In this context we refer to navigation commands as operations that rotate, translate, and/or scale the rendered 3D object.

Predictable Commands Are Important for Tracking Changes. When commands are predictable, the user is able to clearly track the state of the object. We found 90° rotation increments were predictable by the user in contrast to finer resolution of 10 – 30°. Similarly, for scaling, users often wanted to know the objects' scale to use as reference. U3 noted that this was similar to *"exploring a map, where knowing discrete zoom scales is extremely useful"*. Overall, this showed us that despite only being able to feel one plane of the object at a time, users are still able to integrate the different pieces of information to form an accurate mental model of the object, *"it's like a tactile display, where you can only see one section at a time"* (U2).

Resetting the model to a known position. When learning about the system, users would explore the model and after a series of operations they would get lost. Providing a way to reset the model to its initial position allowed users to quickly ground themselves back to the orientation matching the code and verify their understanding of the 3D model's axis.

Considerations for "visibility" and spatial mapping of UI elements. In the initial prototype we introduced the navigation controls for rotation, translation, and scaling as a series of keyboard commands. However, we quickly found that this lead to having to remember too many keyboard shortcuts and as more features were added, it became more difficult for users to remember all the features in addition to all the keyboard commands that were already used with their assistive technology. Users commented on the need for something more spatial that feels *"as if I'm actually turning the object"*.

Continuous scaling allows users to reveal more details. Users exploration strategy typically entailed first exploring all sides of the 3D model at a scale that fit within the shape display limits to get an overall picture of the model's shape. Next, users would zoom into various parts when they wanted to reveal more details. As was found in a previous study [51], users referred to this strategy as *"bringing the part into focus"*.

Challenges Related to Hardware Limitations

Using a 2.5D shape display for rendering 3D models has a few limitations due to technical challenges. For example, depending on the orientation, overhangs will appear as solid. Moreover, objects may be aliased when rendered because of the limited pin resolution. We sought to understand how these limitations affected users' understanding of 3D models and what system features could help mitigate them. As U2 commented, *"there's a learning curve for using the system. The most difficult part is visualizing the shape and understanding the shape display hardware."*

Understanding When Changes Are Out of the Rendering Region. When learning about the system, users were encouraged to make changes to the sample files to understand how they changed what was rendered on the display. Occasionally we found users' understanding of the model was not initially correct and when they modified the file, nothing would change on the rendered object. This was because users were modifying geometry that was clipping in the display but not being aware of this, they confused it with the program being unresponsive. Giving users some context of where the 3D geometry is clipping based on new changes could be helpful for grounding. U2 expressed that this was initially one of the more challenging parts because, *"one part is not there, or it's there but you need to understand it's also below the base of the display"*. U3 expressed a similar sentiment, *"It would be nice to know the direction of where changes are clipping."*

Quickly Integrating Views to Get an Overall Mental Model. To obtain a full mental model of the 3D geometry, users must use the navigation commands to feel and integrate the different sides of the object. This is similar to exploring a real object, as U3 commented, *"if I had a real object, I would be turning it around as much as possible to get a full understanding of the shape."* It can be helpful having multiple views shown at once so that users don't need to perform as many navigation steps, *"it would be less work if I could just feel both [top and bottom] at the same time"* (U2).

Understanding Where There Are Overhangs or Internal Cuts. 2.5D rendering using physical pins is limited to rendering of

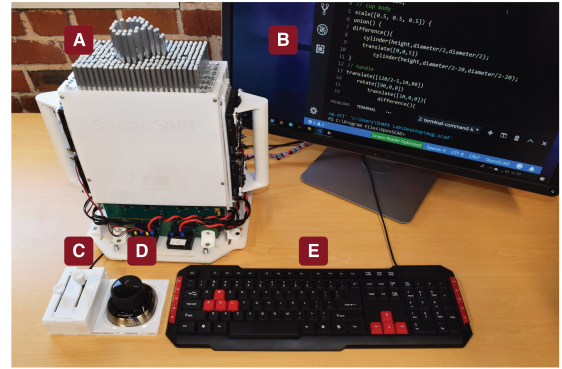


Figure 2. Overall setup of shapeCAD comprised of A) a 2.5D shape display, B) OpenSCAD programming language, C) slider inputs for control of zoom and cross section view, D) 3D mouse for translation and rotation, E) and computer keyboard.

reliefs and prevents overhangs or internal cuts. For example a mug rendered upright will appear with its handle filled rather than hollow because pins actuate vertically (Figure 1). While a mug handle is more easily understandable that it should be hollow, this is not clear for all objects, as U3 commented, *"having pins being solid, can be confusing. I need to think 'I know that there is a space in those pins not showing from the side'"*. Having a mechanism to communicate this to the user can be important for understanding of the 3D model.

SHAPECAD SYSTEM IMPLEMENTATION

Based on the findings identified through the co-design process, we refined our system and implemented shapeCAD. The system consists of an accessible programming environment for writing, and compiling OpenSCAD scripts (Fig. 2A). The output 3D model is rendered on a 2.5D shape display (Fig. 2B). A set of sliders, 3D mouse, and keyboard are used to interact with the rendered model and verify the design (Fig. 2C-E).

Microsoft Visual Studio Code (VSC) [v1.33.1] is used as the programming environment for modifying OpenSCAD [v2015.03-2] scripts (Fig. 2B). An extension was written for creating built-in commands that compile and render OpenSCAD files directly from the VSC editor. The extension was written to overcome several of the accessibility challenges with the standard OpenSCAD graphical user interface that were found through the co-design sessions. VSC was tested throughout the co-design process and chosen because of its accessible interface, and ease for users to install custom extensions. The extension is available online for download and installation: <https://github.com/alexasiu/shapeCAD-Extension>.

The main application that interfaces with the shape display was implemented in Unity. This application manages updating the shape display when a user compiles a file, providing audio feedback, and listening for user input from the keyboard, sliders, and 3D mouse (Fig. 2C-E). When a file is rendered the application loads it, computes the model's centroid and uses it to render the model centered on the display. If the model has previously been rendered, it preserves its previous transformation. An audio confirmation with a description of the file name is provided when a file newly rendered or updated. While OpenSCAD is unitless, the application by default assumes units of *mm* and scales objects appropriately.

Navigation (Translation, Rotation, & Scaling): When navigating a 3D model, users are able to translate, rotate, scale, and reset the model. Translation and rotation operations are commanded using a 3D mouse [3D Connexion 3DX-700066] (Fig. 2D) with six-degrees-of-freedom. Only one operation can be performed at a time; this is accomplished by filtering the 3D mouse input to just take the dominant degree-of-freedom.

Rotation operations are applied in 90° increments so users are able to track the state of the model as was found through the co-design sessions. Translation on the other hand is continuous; audio warnings are provided when the user is translating the model out of the rendering region. This was found useful to prevent users from accidentally losing the model. For scaling the model, the user can press the plus and minus keys on the keyboard for continuous scaling or use a 3-position slider for discrete scaling (Fig. 2C). The discrete scaling allows users to quickly toggle to a known scale for making comparisons while the continuous scaling allows users to reveal more details in the model. Lastly, for resetting the model back to its initially rendered orientation and scale (1:1), the user can press the *Enter* key on the keypad. If the user modifies the model's source code and re-renders it on the display, the model preserves its last orientation and scale.

Revealing Hidden Differences: When the user updates a 3D model and the changes are not visible in the shape display rendering region, an audio warning is provided to indicate so. The user can then toggle the 'reveal hidden differences' mode to get haptic feedback on where the clipping is occurring. This feedback is provided by vibrating the pins closest to the hidden changes. In Fig. 3 the user initially renders a cylindrical cup (3A). The user modifies the file to add a handle (3B), and when it is rendered an audio warning is given that the changes to the model are clipping. The user can then turn on the reveal hidden differences mode which makes the pins corresponding to the handle vibrate (3C). This is because half of the handle is clipping below the rendering region. Toggling this and all of the following modes is accomplished using the numeric keypad; shortcuts were chosen to not interfere with the screen reader and to be located in a spatially convenient location as determined through the co-design session.

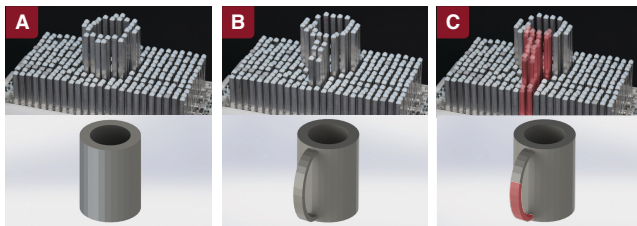


Figure 3. Audio and haptic warnings are given to users when a change they have made to the 3D model is not visible in the shape display. A) The user initially renders a cylindrical cup. B) The user makes changes to the program to add a handle, and when it is rendered an audio warning is given that changes to the model are clipping. C) The user can then turn on the 'reveal hidden differences' mode which vibrates the pins corresponding to the handle since half of the handle is clipping (in red).

Split View: One approach we use to help users more quickly integrate the different views of the 3D model is to render two

simultaneous views as shown in Figure 4. The split view mode can be toggled using the keyboard (Fig. 2E). This mode splits the display in half to render the top-down view of the model on the left, and the bottom-up view of the model on the right. In Figure 4, the user can quickly understand that the rendered mug is open on the top but closed from the bottom.

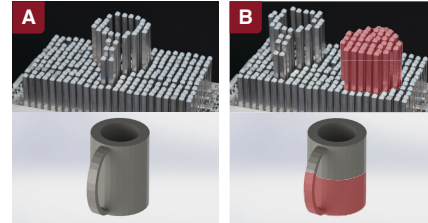


Figure 4. Users can toggle the split view mode to show two simultaneous views of the 3D model. This mode splits the display rendering region to show both a top-down view, and a bottom-up view of the model.

Section View: In commercial CAD systems, users are able to see overhangs and internal cuts using transparency views. However, in this context where *transparent tactile views* are not possible, alternate means for users to be able to identify overhangs and internal cuts are essential for understanding. To allow discoverability of these kinds of geometry, shapeCAD has a mode that allows the user to slice through the 3D model. Similar to split view, this mode is toggled on/off by the user and the slice plane is controlled with a slider (Fig. 2C). Figure 5 shows an example where a mug has been sliced at different points; this allows the user to verify the mug's body is hollow and explore other internal features.

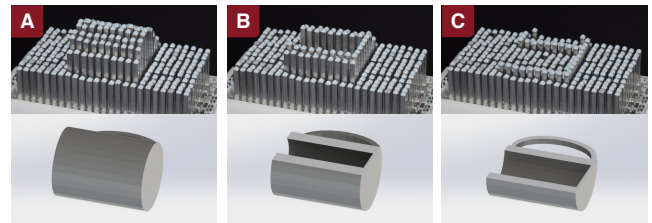


Figure 5. Users can toggle section view to render cross sections of the 3D model they are viewing. (A), (B), and (C) show different cross sections of a mug. The slice plane is controlled by the user through a slider.

EVALUATION

We were interested in users' opinions about shapeCAD its effectiveness and ease of use, workflow adaptability, ease of understanding existing models and completing a desired task or operation, and practicality of the tool for designing new 3D geometry. To assess these, we conducted an evaluation where participants were introduced to shapeCAD and then asked to complete a series of controlled tasks with a set goal as well as an open-ended task where participants created a new model from scratch.

Demographics

We recruited another five participants (P1-P5) for the evaluation through local blindness organization mailing lists. Four of the participants were blind and one was visually impaired (central vision loss). One user (P2) had 3D modelling experience with OpenSCAD. Four of five users (P1, P2, P4, P5)

had extensive programming experience, having either studied computer science as a college major or worked as software developers at some point in their careers. We specifically sought to recruit users with programming experience to be able to run a one-session study. From our early co-design sessions and pilot studies, we found that teaching the system to non-programming experts took additional training time to introduce some programming concepts. All participants received a 50 USD gift card in compensation for their time. Transportation costs that incurred were also compensated.

Materials & Methods

The study was composed of a pre-survey, a training phase, and the experiment. The experiment had three tasks, two were controlled 3D modelling tasks with a specified goal (up to 20 minutes) and the last one was an open-ended task driven by participant's goals (up to 25 minutes). Each task was followed by a survey. Participants were asked to think aloud as they worked through each task, in particular, we asked them to verbalize what questions were raised in their minds, what challenges they were facing, and what strategies they had or were planning. Video and audio were recorded for analysis.

Pre-Survey & Introduction. Two days before the scheduled study, participants were given a pre-survey (13 questions) with demographics questions, use of assistive technology, experience in programming and 3D modelling, and interest in 3D modelling and printing. An overview of the system and documentation on selected OpenSCAD functions was also provided. The selected functions were: sphere, cylinder, cube, rotation, translation, union, and difference.

Training Phase. At the start of the study, the experimenter explained the study protocol, obtained IRB consent, and started the training phase. Participants were given a walkthrough of shapeCAD using script examples, followed by a practice task. Participants were introduced to the OpenSCAD syntax and learned how to use the different system features. In total, the training phase exposed participants to five different 3D models for viewing and exploring. After the walkthrough, participants were given a 3D model to practice the controlled task they would be completing. They were told to use the training as an opportunity to ask additional questions about both the shapeCAD and/or the protocol. This practice task was different from the ones used in the actual tasks. In total this training phase lasted up to 40 minutes.

Controlled 3D Modelling Tasks: Transformation & Construction. Users were given a 3D printed goal model and an associated script file. The model generated from this script did not match the goal and it was participants' task to modify the script so it would match the goal. The first task was a transformation task that tested understanding of shape placement, orientation, and scale. The second task was a construction task; this task tested two additional concepts related to CSG: choosing primitive shapes, and applying the appropriate boolean operations. Both tasks initial model and goal model are shown in Figure 6 and were modelled based on introductory TinkerCAD [58] starter lessons. The purpose of these tasks were to assess whether users were able to understand existing code, and 3D models and modify them to achieve specific goals.

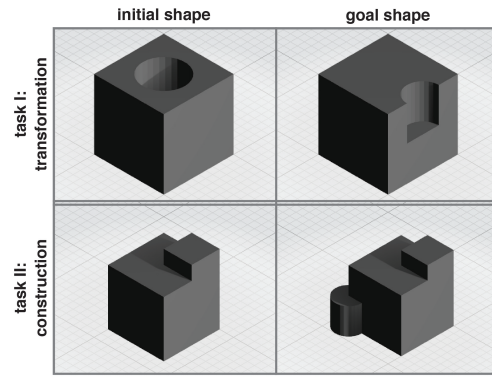


Figure 6. Transformation task (top row) and construction task (bottom row) used for the controlled tasks in the evaluation. Participants were shown a 3D printed model of the goal shape and a script that defined the initial shape. Participants task was to update the script to match the initial shape to the goal shape.

Task Difficulty Survey. We used the Single Ease Question (SEQ) after users completed each task. The SEQ is a Likert-type rating scale to evaluate how difficult users find a task typically conducted after performing a usability task [46].

Free Form Creation Task. The last portion of the study allowed users to create a new model from scratch. As a starting point, participants were given the prompt of designing a container of anything and shown 3D printed examples of a cup with a flange, a desk organizer, and a toothbrush holder. The purpose of this task was to assess whether participants were able to ideate new models based on personal goals. In relation to 3D modelling and CSG concepts, we specifically looked at whether participants were able to decompose an idea into primitive shapes and apply correct boolean operations.

Free Form and General Post-Survey. After the free form task, participants also completed a survey related to their experience in the task. This survey asked about the object participants wanted to create, the challenges they faced, how well the result matched what they originally envisioned, and their confidence and satisfaction levels. Afterwards, users completed a more general survey. This survey contained questions from the standard usability scale (SUS) [4] to obtain a general usability score, questions on ease of use and usefulness of specific system features based on the Questionnaire for User Interaction Satisfaction (QUIS) [7] to identify critical parts that affect usability, and questions on workload (NASA-TLX) [20]. There were also open-ended qualitative questions on the shapeCAD's best and worst attributes, and how likely participants would be to continue using it.

Quantitative Results

Controlled Tasks Accuracy and Time. P1-P4 completed all controlled tasks successfully. P5 did not accurately complete the construction task, having placed the cylinder at an opposite corner (Fig. 6). On average, participants took 6.4 minutes ($\sigma = 2.96$) to complete the transformation task and 14.6 minutes ($\sigma = 4.88$) to complete the construction task.

Free Form Task Satisfaction and Completion. P1-P4 reported being extremely satisfied with the results and thought the objects matched what they initially sought to create. P5 reported

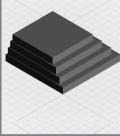

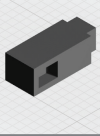







P1	P2	P3	P4	P5
				
				
15.42 min	18.78 min	21.09 min	12.84 min	8.01 min
"staircase stand"	"tall glass cup"	"truck with carriage"	"a cube with a dome on top"	"a cylinder cup"

Figure 7. Objects created by all participants during the evaluation’s free form task, the time it took to create them, and participant’s description of the object. The first row shows the rendering of the 3D model and the second row shows the object 3D printed using a MakerBot Replicator. All participants reported being satisfied with the final results except P5.

being not quite satisfied with the results. Figure 7 shows both a rendering and 3D printed versions of the objects participants created, the time it took to complete them, and the reported description of the object. On average, participants took 17.03 minutes ($\sigma = 5.12$) to complete the task.

Task Difficulty and Workload. The average normalized SUS score for subjective system satisfaction was 70 ($\sigma = 15.10$). Participants rated task difficulty on a scale from 1 (Extremely Difficult) to 5 (Extremely Easy). The free form task was rated as the most difficult ($\mu = 2.0, \sigma = 0.7$), followed by the construction task ($\mu = 2.8, \sigma = 1.3$), and last the transformation task ($\mu = 3.6, \sigma = 1.14$). Participants were also asked to rate workload measures of Mental Demand, Physical Demand, Temporal Demand, Effort and Frustration on a scale from 1 (Very High) to 5 (Very Low). In addition, participants rated their Performance on a scale from 1 (Terrible) to 5 (Excellent). Results show the task required high effort ($\mu = 2.6, \sigma = 0.89$) and was mentally demanding ($\mu = 2.0, \sigma = 0.70$). However, participants did not feel frustrated ($\mu = 4.4, \sigma = 0.89$), they rated both temporal ($\mu = 4.4, \sigma = 0.89$) and physical demand very low ($\mu = 4.8, \sigma = 0.45$) and in general perceived their performance as good ($\mu = 4.4, \sigma = 0.89$).

Ease of Use & Features Usefulness. Participants rated how easy they found using shapeCAD on a scale from 1 (Extremely Difficult) to 5 (Extremely easy). "Toggling between features" was rated highest ($\mu = 4.8, \sigma = 0.45$), followed by "Using navigation controls for rotation" ($\mu = 4.4, \sigma = 0.55$), "Using navigation controls for translation" ($\mu = 4.0, \sigma = 1.22$), and "Understanding rendered shapes" ($\mu = 4.0, \sigma = 0.71$). The last set of questions asked participants to rate how useful they found each of the shapeCAD’s features on a scale from 1 (Not at all useful) to 5 (Extremely useful). "Resetting the model" was rated the most useful feature ($\mu = 5.0, \sigma = 0.0$), followed by "section view" ($\mu = 4.4, \sigma = 0.55$), "scaling" ($\mu = 4.4, \sigma = 0.89$), "split view" ($\mu = 4.4, \sigma = 0.89$), "audio cues" ($\mu = 4.2, \sigma = 0.84$), and "revealing hidden differences" ($\mu = 3.2, \sigma = 1.10$).

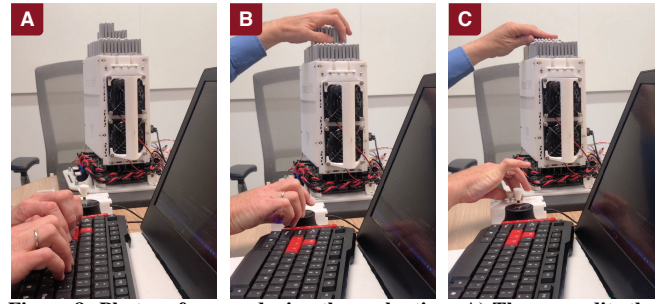


Figure 8. Photos of a user during the evaluation. A) The user edits the 3D model code and renders it. B) Then uses the 3D mouse to orient the object and C) the section view slider to visualize internal geometry.

Qualitative Results & Discussion

While completing the study tasks, we asked participant to think aloud. These kinds of protocols have been used extensively for uncovering challenges novices face in learning a new system [12]. We use a similar approach to analyze the qualitative data collected from video recordings of the study [12, 47]. The sections below organize these findings by themes. Figure 8 shows a participant completing a task using the keyboard to modify the code (A), the 3D mouse to navigate the model (B), and the sliders to reveal cross-sections of the model (C).

Strategies for Understanding, Updating, and Verifying

Throughout the study, we observed common strategies adopted by all participants when exploring new 3D models. The rotation and translation commands were typically used to get a slower but more accurate overview of the model. Participants would examine each face of an object, rotating it through all its sides. This was typically done when first exploring a new object to identify what parts mismatched from the goal object or to verify the accuracy of their completed design, "I'm trying to isolate the cylinder to see if it's in the right place [while rotating and translating the model in Task 2]" (P2).

On the other hand, when iterating on changes to the code and observing how it affected the physical output, participants would rely more heavily on the split view and section view features. P4 commented that these features allowed him to more "quickly view the figure from different angles..." as opposed to using the translation and rotation commands which provided a more holistic view. In particular, we found these features were helpful when participants had a specific error to diagnose or geometric detail to verify. For example, P2 commented when using section view during the construction task, "I think the cylinder should be hanging in space because I turned the object sideways so I'm using the cross section view to make sure this is true". This confirms similar needs as those found through the co-design sessions for additional interactions that allow users to obtain more views of the model without having to perform navigation operations. Three out of the five users (P3-P5) described the various features to manipulate the 3D models as some of shapeCAD’s best attributes, "I really liked the functions the split views and section view" (P3).

Participants relied on different zoom levels when wanting to view in more detail certain parts of the object, "What I'm doing now, is I think it's right but I'm scaling it up to get more resolution [when verifying completion of Task 1]" (P2). The

discrete zoom slider allowed participants to do this quickly and to easily toggle back to a 1:1 scale when needed. Similarly, P1 created a small object (5 cm × 5 cm × 3 cm) and was typically verifying the design at a scale 1.5x the original.

Similar to the co-design sessions, we found that resetting the model to a known position and scale was extremely important. In general, participants would constantly use the reset command after applying a number of operations or before toggling back to the code. This is also reflected in the survey where resetting the model was rated as the most useful feature. Compared to the co-design sessions, we found less use for the continuous scaling. This could be a limitation in the study design because the tasks people performed, and objects they created did not have very small geometric details. Similarly, participants rated the reveal hidden differences feature as the least useful. This could be because the particular tasks in the study were too simple or because the users needed more training to familiarize with the feature, as P4 commented, *"It's something I would need to play more with."*

Recurrent Challenges

While the navigation strategies participants used to complete the tasks were similar to those found through the co-design, we found additional challenges that still need to be addressed.

Challenges In Visualizing Difference Operations. Participants that used the difference operation during the free form tasks (P2, P3, P5) ran into problems related to locating the shape they wanted to subtract or cut from the main shape (Figure 7). As P2 described it, *"it's difficult when you think you've made a shape but you can't find it. Or there's a shape you want to subtract from another one but you can't find the negative space. It would be nice to toggle between the normal rendering and just showing the thing you want to subtract, like switching in space."* P3 shared a similar sentiment when creating the truck carriage and P5 when creating the cylindrical cup and properly applying the cut out for the cup, he was able to apply it but not centered on the outer cylinder shell.

Challenges in Understanding the Code. Participants also faced some challenges in matching the code representation and the rendered representation. While it was easy to identify in the shape display the feature they wanted to change or trace the operation they wanted to apply, this was not always as easily translatable to the programming domain. P5 commented, *"So I know the object I want to get to. I know I need to get to the translate command that is in the middle of the code and I think I'm there but I'm not there and then I make changes to the wrong thing."* These challenges can be partly attributed to familiarity with the programming language, *"The more difficult part was writing the syntax correctly, but I had the right idea."* (P4), or need for more training time using shapeCAD. In the survey all participants reported the tasks took a high mental demand and required high effort.

A strategy participants took when they did not understand the code, was a "process of elimination" strategy; as P1 commented when working through a task, *"let's render and see what comes up"*. Participants would make a small change to the script and check the model to verify their understanding of

"how the axis align with the model" (P2). This demonstrates the usefulness of feedback for both viewing 3D models and for helping users better understand the code and also supports the findings that were addressed through the co-design sessions. P1 described this almost immediate feedback as the shapeCAD's best attribute for giving blind people more independence in design, *"No trouble with hardware or flow of the system. The feedback will make it easier. Makes it possible for a blind person to create."*

Despite the high effort to learn, all participants expressed their interest in continuing to use shapeCAD and a desire for more accessible 3D design tools, *"I would definitely use it because I have an interest in 3D models professionally. I think it has a lot of potential. At the very least I would be able to view. But if I learned maybe I could create the next rocket engine"* (P1). Participants were mostly interested in creating 3D models for the use of other BVI individuals. P1-P5 were interested in the creation of maps and learning materials, *"Creating educational materials for the blind"* (P1).

DESIGN IMPLICATIONS

Overall our results demonstrate that shapeCAD is accessible to BVI people. Despite the limitations of 2.5D shape displays, e.g. low resolution and limited views, novice users are able to effectively use the feedback to complete a series of basic 3D modelling tasks. Based on our process and study results, we summarize important design considerations when designing accessible 3D modelling systems for the BVI that support programming and feedback via 2.5D tactile shape displays:

Enhancing Low Resolution Tactile Representations With Additional Context. Navigation of 3D models in a low resolution display can be challenging, however, supporting it with additional context can help users make better sense of the representation. In this system we supported users' understanding of 3D models rendered on a 2.5D shape display with scripts written in OpenSCAD. The tactile feedback gives user a better overview of the 3D model's spatial layout, but the code allowed users to verify the details of shapes and sizes when geometry appears aliased due to the low resolution.

Supporting Both Quick and Detailed Exploration of 3D Models. Navigation commands, e.g. rotation and translation, allow users to explore a 3D model from different views to obtain a full mental model. However, these methods may require high mental demand and can be slow; which may be unsuitable for quick iterative prototyping. Thus it is also important to provide additional features that allow users to get a quicker, albeit less complete, understanding of the output model. In our implementation we enable this through two features: split view which gives users two simultaneous views of the 3D model, and section view which allows users to slice objects and identify overhangs and internal geometry.

Creating Direct Mappings Between 3D Model Representations. One of the main challenges we observed was in translating manipulation commands in the physical world to its equivalent code representation. Often participants pointed out on the shape display the operations they wanted to apply but had trouble translating it into code. As P3 commented, *"directly*

pushing and pulling the pins could be nice and then seeing the code after that. Then it would be reversed from the current configuration, I would make the shape then see the code". These highlight the importance of supporting users transitioning from the code to the physical representation and vice versa. shapeCAD mostly supports translating from the code to physical representation. But the opposite transition from the physical output to explicit lines of code could also be helpful in providing a tight loop between the different representations.

Reducing the Learning Curve and Supporting Active Learning. All participants agreed shapeCAD had a steep learning curve in the number of things they had to learn to get started but they also felt that they could keep learning, *"I would need practice to get really good, but I can see that I could really do this"* (P4). The quick feedback from the 2.5D shape display enables users to take a more active learning stance when they are stuck in writing or understanding the program. In the evaluation, we observed when participants were lost, they could experiment with modifying a parameter in the code, and the rendered output helped them better understand how the two were related. Systems that allow for learning by experimentation allow flexibility for users to more easily recover when stuck and promote self-directed learning [15, 56].

Increasing Independence in the Design Process. A clear theme from both co-designers and evaluation participants was the general lack of design tools that BVI people can use independently. If blind people are to design and build the tools they need and want, they must have direct access to the tools needed to design such things. The BVI community can bring a valuable perspective to design and it is imperative to increase accessibility in tools that enable participation; as P4 commented, *"I would like to be able to help designers in my department to produce 3D models... a lot of the work we do is in tactile design and that's something that I currently don't have access to because the tools we use aren't accessible. So the fact that I would be able to use a tool that I could control and could actually create and design 3D models is just revolutionary and very exciting"*.

LIMITATIONS & FUTURE WORK

In this work we specifically investigate the use of a 2.5D shape display for providing feedback to BVI people when programming to complete a series of 3D modelling tasks. There are inherent limitations in using the specific hardware with shapeCAD such as the resolution of pins and the size of the rendering region chosen. While studies have investigated the optimum resolution for a haptic graphic display to be 10 taxels per inch (tpi) [9], they have also emphasized the need for user familiarization with the device and the tradeoffs with different resolutions has not been investigated. This resolution is in comparison to a 3.5 tpi used with shapeCAD. Some of the interactions might differ with different rendering resolution and speeds. For example, perceptual studies have shown that a tight coupling between tactile information specifying a rotating object and the corresponding mental model are critical for understanding [60]. Because rendering speed limitations, users preferred discrete rotations when exploring a model with

shapeCAD. However, with sufficiently fast rendering, it is possible that continuous rotation could be easier to interpret.

In the evaluation conducted, we focus on a short two-hour session with complete novices, thus the level of complexity for the tasks presented is only introductory. Further work is needed to see how shapeCAD extends when working with more advanced geometry. In addition, several of the users felt their unfamiliarity with the programming language was a limitation in completing the tasks. While teaching and training can be a big factor, perhaps other methods could be investigated that reduce the large efforts required from users in getting started. For example, direct modelling techniques could provide more intuitive inputs for manipulating the 3D models instead of manipulating them through code.

Future studies could also assess the potential of a tool like shapeCAD as a platform to increase engagement in STEM. The Maker Movement has often been used as an entry point for contextualized learning [3]. As one co-designer, who had no previous programming background, pointed out, he felt the interactivity and quick feedback loop made the idea of programming less daunting from what he originally thought.

Lastly, while we have focused on a general tool for 3D modelling, all co-designers and evaluation participants had a domain-specific interest for learning about 3D modelling. Extensions could be made to directly support the specific creations such as tactile maps, graphics, and project enclosures. In most cases users wanted to take the role of designers, to create content by and for the BVI community. There are many possibilities for increasing access to design tools and broadening participation of an eager community ready to engage.

CONCLUSION

Few 3D user interfaces have been created for the BVI community to engage in design and *making*. In this work, we present several challenges faced by this community in accessing 3D design and investigate one possible workflow, towards increasing accessibility. With shapeCAD, we demonstrate a set of interactions that allow us to use 2.5D shape displays to provide near-immediate feedback when authoring and editing 3D models. In a short period of time, novices in our evaluation were able to ideate a range of objects. Increasing access to design tools can open up a new medium for expression and creative potential that others across the world already engage with. Unless BVI makers have direct access to design and fabrication tools, they are forced to work through sighted intermediaries, reducing agency, availability, and creativity. A dynamic, real-time display capable of producing 2.5D representations of the work in progress provides essential feedback during the design process and makes the process more efficient and independent.

ACKNOWLEDGEMENTS

We thank all co-designers and users for their time and feedback, and Parastoo Abtahi, Elyse Chase, Inrak Choi, and Michael Lin for their help with editing figures and providing feedback. This work was supported by the NSF GRFP (#DGE-114747).

REFERENCES

- [1] Peter Bennett, Mike Fraser, Duncan Boa, and Ben Hicks. 2017. InstructiBlocks: Designing with Ambiguous Physical-Digital Models. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 551–558.
- [2] Jeffrey P Bigham, Ryan S Kaminsky, Richard E Ladner, Oscar M Danielsson, and Gordon L Hempton. 2006. WebInSight:: making web images accessible. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 181–188.
- [3] Paulo Blikstein. 2013. Digital fabrication and ‘making’ in education: The democratization of invention. *FabLabs: Of machines, makers and inventors* 4 (2013), 1–21.
- [4] John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
- [5] Craig Brown and Amy Hurst. 2012. VizTouch: automatically generated tactile visualizations of coordinate spaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. ACM, 131–138.
- [6] Erin Buehler, William Easley, Samantha McDonald, Niara Comrie, and Amy Hurst. 2015. Inclusion and Education: 3D Printing for Integrated Classrooms. *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '15* (2015). DOI: <http://dx.doi.org/10.1145/2700648.2809844>
- [7] JP Chin, VA Diehl, and KL Norman. 1988. Questionnaire for User Interaction Satisfaction (QUIS). *Human-Computer Interaction Lab, University of Maryland at College Park* (1988).
- [8] Jonathan Cohen, Jonathan Cohen, W. Monty Jones, Shaunna Smith, and Brendan Calandra. 2017. Makification: Towards a Framework for Leveraging the Maker Movement in Formal... *Journal of Educational Multimedia and Hypermedia* 26, 3 (2017), 217–229.
- [9] Damian Copeland and Janet Finlay. 2010. Identification of the optimum resolution specification for a haptic graphic display. *Interacting with Computers* 22, 2 (2010), 98–106. DOI: <http://dx.doi.org/10.1016/j.intcom.2009.11.001>
- [10] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz studies—why and how. *Knowledge-based systems* 6, 4 (1993), 258–266.
- [11] Julie Ducasse, Anke M Brock, and Christophe Jouffrais. 2018. Accessible interactive maps for visually impaired users. In *Mobility of Visually Impaired People*. Springer, 537–584.
- [12] K Anders Ericsson and Herbert A Simon. 1984. *Protocol analysis: Verbal reports as data*. the MIT Press.
- [13] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation. *UIST '13 Proceedings of the 26th annual ACM symposium on User interface software and technology* 13 (2013), 417–426.
- [14] American Printing House for the Blind. 2018. Introducing Graphiti - A Revolution in Accessing Digital Tactile Graphics. (2018). <https://www.aph.org/graphiti/>
- [15] Denis Gillet, Anh Vu Nguyen Ngoc, and Yassin Rekik. 2005. Collaborative web-based experimentation in flexible engineering education. *IEEE Transactions on Education* 48, 4 (2005), 696–704.
- [16] T Götzelmann. 2018. Autonomous Selection and Printing of 3D Models for People Who Are Blind. *ACM Transactions on Accessible Computing (TACCESS)* 11, 3 (2018), 14.
- [17] Timo Götzelmann, Lisa Branz, Claudia Heidenreich, and Markus Otto. 2017. A Personal Computer-based Approach for 3D Printing Accessible to Blind People. In *Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 1–4.
- [18] Darren Guinness, Edward Cutrell, and Meredith Ringel Morris. 2018. Caption crawler: Enabling reusable alternative text descriptions using reverse image search. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 518.
- [19] Erica Rosenfeld Halverson and Kimberly Sheridan. 2014. The Maker Movement in Education. *Harvard Educational Review* 84, 4 (Dec 2014), 495–504. DOI: <http://dx.doi.org/10.17763/haer.84.4.34j1g68140382063>
- [20] SG Hart and others. 1988. Development of NASA-TLX: Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati (eds.), *Human Mental Workload*. (1988).
- [21] Leona Holloway, Kim Marriott, and Matthew Butler. 2018. Accessible Maps for the Blind: Comparing 3D Printed Models with Tactile Graphics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 198.
- [22] Amy Hurst and Shaun Kane. 2013. Making making accessible. In *Proceedings of the 12th international conference on interaction design and children*. ACM, 635–638.
- [23] Youssef Bou Issa, Mustapha Mojahid, Bernard Oriola, and Nadine Vigouroux. 2009. Accessibility for the blind: An automated audio/tactile description of pictures in digital documents. In *2009 International Conference on Advances in Computational Tools for Engineering Applications*. IEEE, 591–594.

- [24] Shaun K Kane and Jeffrey P Bigham. 2014. Tracking@ stemxcomet: teaching programming to blind students via 3D printing, crisis management, and twitter. In *Proceedings of the 45th ACM technical symposium on Computer science education*. ACM, 247–252.
- [25] Jeeun Kim, Abigale Stangl, and Tom Yeh. 2014. Using LEGO to model 3D tactile picture books by sighted children for blind children. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*. ACM, 146–146.
- [26] Roberta L. Klatzky, Susan J. Lederman, and Victoria A. Metzger. 1985. Identifying objects by touch: An expert system. *Perception & Psychophysics* 37, 4 (01 Jul 1985), 299–302. DOI: <http://dx.doi.org/10.3758/BF03211351>
- [27] Richard E Ladner, Melody Y Ivory, Rajesh Rao, Sheryl Burgstahler, Dan Comden, Sangyun Hahn, Matthew Renzelmann, Satria Krisnandi, Mahalakshmi Ramasamy, Beverly Slabosky, and others. 2005. Automating tactile graphics translation. In *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 150–157.
- [28] David H Laidlaw, W Benjamin Trumbore, and John F Hughes. 1986. Constructive solid geometry for polyhedral objects. In *ACM SIGGRAPH computer graphics*, Vol. 20. ACM, 161–170.
- [29] Rebecca Lawson and Sarah Bracken. 2011. Haptic Object Recognition: How Important are Depth Cues and Plane Orientation? *Perception* 40, 5 (2011), 576–597. DOI: <http://dx.doi.org/10.1068/p6786> PMID: 21882721.
- [30] Susan J Lederman and Roberta L Klatzky. 1987. Hand movements: A window into haptic object recognition. *Cognitive Psychology* 19, 3 (1987), 342 – 368. DOI: [http://dx.doi.org/https://doi.org/10.1016/0010-0285\(87\)90008-9](http://dx.doi.org/https://doi.org/10.1016/0010-0285(87)90008-9)
- [31] Ben Leduc-Mills and Michael Eisenberg. 2011. The UCube: a child-friendly device for introductory three-dimensional design. In *proceedings of the 10th international conference on interaction design and children*. ACM, 72–80.
- [32] Fabrizio Leo, Elena Cocchi, and Luca Brayda. 2017. The effect of programmable tactile displays on spatial learning skills in children and adolescents of different visual disability. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25, 7 (2017), 861–872.
- [33] Thomas H Massie, J Kenneth Salisbury, and others. 1994. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, Vol. 55. Citeseer, 295–300.
- [34] Joshua A Miele, Steven Landau, and Deborah Gilden. 2006. Talking TMAP: Automated generation of audio-tactile maps using Smith-Kettlewell’s TMAP software. *British Journal of Visual Impairment* 24, 2 (2006), 93–100.
- [35] Kazunori Minatani. 2017. Finding 3D CAD Data Production Methods that Work for People with Visual Impairments. In *HCI International 2017 – Posters’ Extended Abstracts*, Constantine Stephanidis (Ed.). Springer International Publishing, Cham, 548–554.
- [36] Paolo Motto Ros, Vittorio Dante, Luca Mesin, Erminio Petetti, Paolo Del Giudice, and Eros Pasero. 2014. A new dynamic tactile display for reconfigurable braille: implementation and tests. *Frontiers in neuroengineering* 7 (2014), 6.
- [37] Stefanie Mueller, Anna Seufert, Huaishu Peng, Robert Kovacs, Kevin Reuss, François Guimbretière, and Patrick Baudisch. 2019. FormFab: Continuous Interactive Fabrication. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 315–323.
- [38] Yuenyong Nilsiam and Joshua M Pearce. 2017. Free and open source 3-D model customizer for websites to democratize design with OpenSCAD. *Designs* 1, 1 (2017), 5.
- [39] Braille Authority of North America. 2010. Guidelines and Standards for Tactile Graphics. (2010). <http://www.brailleauthority.org/tg/>
- [40] OpenSCAD. 2015. OpenSCAD - The Programmers Solid 3D CAD Modeller. <https://www.openscad.org>. (2015). Accessed: 2019-05-02.
- [41] Thomas Pietrzak, Andrew Crossan, Stephen A Brewster, Benoît Martin, and Isabelle Pecci. 2009. Exploring geometric shapes with touch. In *IFIP Conference on Human-Computer Interaction*. Springer, 145–148.
- [42] Denise Prescher. 2014. Redesigning input controls of a touch-sensitive pin-matrix device. In *Proceedings of the international workshop on tactile/haptic user interfaces for tabletops and tablets*. 19–24.
- [43] Denise Prescher and Gerhard Weber. 2017. Comparing Two Approaches of Tactile Zooming on a Large Pin-Matrix Device. In *IFIP Conference on Human-Computer Interaction*. Springer, 173–186.
- [44] Deepak Ramteke, Gayatri Kansal, and Benu Madhab. 2014. Accessible Engineering Drawings for Visually Impaired Machine Operators. *Assistive Technology* 26, 4 (2014), 196–201.
- [45] Ravi Rastogi, TV Dianne Pawluk, and Jessica Ketchum. 2013. Intuitive tactile zooming for graphics accessed by individuals who are blind and visually impaired. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 21, 4 (2013), 655–663.
- [46] Jeff Sauro and Joseph S Dumas. 2009. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1599–1608.

- [47] Zachary Schmitt and Svetlana Yarosh. 2018. Participatory design of technologies to support recovery from substance use disorders. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 156.
- [48] Yutaka Shimizu, Shinya Saida, and Hiroshi Shimura. 1993. Tactile pattern recognition by graphic display: Importance of 3-D information for haptic perception of familiar objects. *Perception & Psychophysics* 53, 1 (1993), 43–48.
- [49] Masami Shinohara, Yutaka Shimizu, and Akira Mochizuki. 1998. Three-dimensional tactile display for the blind. *IEEE Transactions on Rehabilitation Engineering* 6, 3 (1998), 249–256.
- [50] Alexa F Siu, Eric J Gonzalez, Shenli Yuan, Jason B Ginsberg, and Sean Follmer. 2018a. shapeShift: 2D Spatial Manipulation and Self-Actuation of Tabletop Shape Displays for Tangible and Haptic Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 291.
- [51] Alexa F. Siu, Joshua Miele, and Sean Follmer. 2018b. An Accessible CAD Workflow Using Programming of 3D Models and Preview Rendering in A 2.5D Shape Display. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '18)*. ACM, New York, NY, USA, 343–345. DOI: <http://dx.doi.org/10.1145/3234695.3240996>
- [52] Nancy Staggers and David Kobus. 2000. Comparing Response Time, Errors, and Satisfaction Between Text-based and Graphical User Interfaces During Nursing Order Tasks. *Journal of the American Medical Informatics Association* 7, 2 (2000), 164–176. DOI: <http://dx.doi.org/10.1136/jamia.2000.0070164>
- [53] Katherine Steele, Brianna Blaser, and Maya Cakmak. 2018. Accessible Making: Designing Makerspaces for Accessibility. *International Journal of Designs for Learning* 9, 1 (2018), 114–121.
- [54] Saiganesh Swaminathan, Thijs Roumen, Robert Kovacs, David Stangl, Stefanie Mueller, and Patrick Baudisch. 2016. Linespace: A Sensemaking Platform for the Blind. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2175–2185. DOI: <http://dx.doi.org/10.1145/2858036.2858245>
- [55] Faisal Taher, John Hardy, Abhijit Karnik, Christian Weichel, Yvonne Jansen, Kasper Hornbæk, and Jason Alexander. 2015. Exploring interactions with physically dynamic bar charts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3237–3246.
- [56] Ruhan Ozkardes Tandogan and Akinoglu Orhan. 2007. The Effects of Problem-Based Active Learning in Science Education on Students' Academic Achievement, Attitude and Concept Learning. *Online Submission* 3, 1 (2007), 71–81.
- [57] Barker Temple. 1990. The benefits of the graphical user interface: A report on new primary research. The benefits of the graphical user interface: A report on new primary research. The benefits of the graphical user interface: A report on new primary research. (1990).
- [58] TinkerCAD. 2019. Direct Starters. <https://www.tinkercad.com/learn/project-gallery;collectionId=OPC41AJJKIDWDV>. (2019). Accessed: 2019-05-02.
- [59] Thomas P Way and Kenneth E Barner. 1997. Automatic visual to tactile translation. II. Evaluation of the TACTile image creation system. *IEEE Transactions on Rehabilitation Engineering* 5, 1 (1997), 95–105.
- [60] Maryjane Wraga, Monique Swaby, and Catherine M Flynn. 2008. Passive tactile feedback facilitates mental rotation of handheld objects. *Memory & cognition* 36, 2 (2008), 271–281.
- [61] Tom Yeh and Jeeun Kim. 2018. CraftML: 3D Modeling is Web Programming. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 527.
- [62] Kai Zhang and Sean Follmer. 2018. Electrostatic Adhesive Brakes for High Spatial Resolution Refreshable 2.5 D Tactile Shape Displays. In *Haptics Symposium (HAPTICS), 2018 IEEE*. IEEE, 319–326.