

ClipWidgets: 3D-printed Modular Tangible UI Extensions for Smartphones

Aaron Visschedijk
njb438@alumni.ku.dk
University of Copenhagen
Denmark

Carlos E. Tejada
ct@di.ku.dk
University of Copenhagen
Denmark

Hyunyoung Kim
hyki@di.ku.dk
University of Copenhagen
Denmark

Daniel Ashbrook
dan@di.ku.dk
University of Copenhagen
Denmark



Figure 1: ClipWidgets are interactive widgets attachable to a dedicated phone case. Markers on the widgets are reflected on a conical mirror in the phone case that makes them visible for the phone camera when inserted (a). We demonstrate three applications of ClipWidgets: a game controller (b); a music-mixing interface (c), and a mathematical equation explorer (d).

ABSTRACT

Touchscreens provide a platform for adaptable and versatile user interfaces making them a popular choice for modern smart devices. However, touchscreens lack physicality. Existing solutions to add

tangible user interfaces to smart devices often require complicated assembly or occlude part of the touchscreen. We propose ClipWidgets: 3D-printed modular tangible UI extensions for smartphones. ClipWidgets uses a conical mirror and a custom phone case to redirect the field of view of the rear camera of a smartphone to the phone's periphery. This allows the phone to optically sense input from modular passive 3D-printed widgets that are attached to the phone case. We developed three different widget types (button, dial and slider) that require minimal calibration and minimal assembly. To demonstrate the functionality of ClipWidgets we used it to prototype three different applications: a game controller, a music interface and an interactive graph tool.



This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike International 4.0 License.

TEI '22, February 13–16, 2022, Daejeon, Republic of Korea
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9147-4/22/02.
<https://doi.org/10.1145/3490149.3501314>

CCS CONCEPTS

• **Human-centered computing** → **User centered design**.

KEYWORDS

User Interfaces, Prototyping, 3D printing, Mobile Computing

ACM Reference Format:

Aaron Visschedijk, Hyunyoung Kim, Carlos E. Tejada, and Daniel Ashbrook. 2022. ClipWidgets: 3D-printed Modular Tangible UI Extensions for Smartphones. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '22)*, February 13–16, 2022, Daejeon, Republic of Korea. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3490149.3501314>

1 INTRODUCTION

Since their introduction, touchscreens have become the dominant interaction paradigm for many different technologies due to their reliability, versatility, and directness. However, tangible interaction methods retain important advantages over touch-only interfaces, including eyes-free manipulation, virtuosity, and high performance manipulation [22] that are yet to be found in touchscreen applications.

In the quest to achieve both the versatility of touchscreens and the benefits of physicality, researchers have explored techniques to increase both the flexibility of tangible devices and the physicality of touchscreens. Researchers in the field of shape-changing interfaces [1] have created tangible controls which can change in size [20], provide haptic feedback [52], or even morph from one type of control into another [19]. Such efforts are still largely restricted to laboratories, due to the complexity of fabricating and controlling shape-changing devices. Another approach has been to increase the physicality of touchscreen-based devices. One major approach in this area have been to use tangible tokens which can be detected by the touchscreen's capacitive sensing system [39, 42] or rear camera for projection-based systems [4, 54]. These methods work well and support a variety of interaction types [9] but must be placed on the screen itself, reducing the available display area. An alternative method has been to use an external camera to track objects in the space around the touchscreen [3, 27], but such approaches are subject to lighting variation and occlusion, and cannot provide a wide variety of interaction types.

In this paper, we present ClipWidgets, a modular, tangible widget system that enhances the physicality of touchscreen devices while avoiding occluding the screen. The components are easily fabricated, requiring only a standard 3D printer, and require minimal assembly and CAD knowledge. The widgets clip to the sides of a 3D-printed case, placing them in close proximity to the screen's high-resolution display. User interaction with widgets causes printed markers inside the case to rotate, which the device camera then detects. To allow interaction with widgets placed at any edge location, a conical mirror provides the camera with a view of the entire periphery of the case. ClipWidgets contributes a unique set of desirable features:

- tangible input for smart devices without screen occlusion;
- the ability to sense different types of input;
- the ability to simultaneously sense multiple inputs;
- no need for special-purpose or external hardware; and
- easy to fabricate and modify.

2 RELATED WORK

Our work relates to fabricating interactive objects and extended interfaces for mobile devices.

2.1 Interactive Fabricated Objects

3D printing is a valuable prototyping tool as it can rapidly turn 3D models into physical objects. However, most 3D-printed objects are not inherently interactive, as limitations of printing technology restricts output to material which can be extruded, powders which can be bonded, or liquid which can be cured. Researchers have devoted significant effort to overcoming these limitations via three main approaches: embedding non-printable components, incorporating multiple types of materials, and using a single printable material type coupled with specifically designed geometry.

Embedding non-printable components allows a maker to take advantage of a huge ecosystem of reliable, inexpensive electronic components and standardized software. However, doing so necessitates creating cutouts and clearances in the 3D-printed model to embed components and wires, requiring significant time as well as significant knowledge of CAD software and electronics. Multiple projects attempt to alleviate some of the burden on designers by automating aspects of this design process, for example by automatically creating case designs based on component layout [53], replacing fiducial markers on scanned physical models with component mounting locations [18, 37], or providing simplified design tools that accommodate entire commercial devices such as smartwatches or phones [24]. The drawback to these approaches is in their complexity, requiring significant manual assembly of the user, and the requirement to re-design and -print the entire object whenever a component type, size, or location changes. ClipWidgets extends this idea of embedding an entire device, but adds modularity, enabling quick and easy experimentation with interactor location and type.

A second approach to add interactivity to 3D-printed objects is by incorporating multiple material types, both those traditionally printable as well as materials requiring novel printer designs. One common approach is to use conductive thermoplastic to enable capacitive touch sensing on printed objects [39, 41–43]. However, these approaches sense a limited number of touch locations [39], require per-object [42] or per-user calibration [43], or operate by occluding a touchscreen [39, 41, 42]. By clever use of high-end multimaterial printers, researchers have been able to print optical fibers and lenses [55] and functional hydraulic actuators [26]. However, with their high cost, large size, and complexity of use, such printers remain out of reach for most people. Another method is to use additional materials that are not normally able to be printed, either via manual insertion during or after the print process, or by constructing or modifying a printer to handle the material. Researchers have incorporated fabric [34, 35], liquids [40], magnets [60], and wire [33]. While offering a wider range of behavior than available with off-the-shelf materials, these methods also incur a higher cost in complexity to the maker. Fabricating ClipWidgets requires only standard PLA material and unmodified hobbyist printers.

To avoid the complexities of multiple components and materials, a number of researchers add interactivity by using the structure of the printed object itself as a transducer, transforming the user's input energy into a form that can be sensed by a single electronic

sensor. One common approach is to use acoustic sensing, where user interaction causes internal resonance [25, 50], causes part of the object to vibrate [38, 45], or interferes with sound injected into the object [23, 32]. However, acoustic approaches are limited in the number and simultaneity of interactions, due to external noise and the generally poor sound conductivity of 3D-printed plastics. Pneumatic approaches, which sense how air pressure changes with user interaction [48, 51], avoid acoustic interference but are similarly limited to a small number of non-simultaneous interactions. Computer-vision-based approaches can potentially support a large number of simultaneous interactions; however, external cameras naturally suffer from occlusion by the user [46] or require the object to remain in a fixed location [44]. Sauron uses a camera internal to the fabricated object to detect motion of physical controls [36]; however, each new or changed object requires significant assembly effort, potentially involving adding mirrors to reflect occluded controls. The modular structure of ClipWidgets simplifies assembly while avoiding the necessity for re-printing the object to make changes, instead enabling easy reconfiguration of widget types and locations. It also uses an internal camera, but avoids occlusion via a single conical mirror that offers a view of the entire perimeter of the phone.

2.2 Physical Interfaces for Mobile Devices

ClipWidgets contributes to the body of research that explores methods to add tangible, physical interaction capability to touchscreen devices. Users prefer tangible controls in some contexts [6, 7], and in some cases such controls can avoid the problem [47] of users occluding the screen during interaction.

One common approach to adding physical controls to mobile touchscreen devices is to use conductive material to extend the touchscreen's touch-detection capabilities to additional objects [14, 24, 28, 39, 42, 59]. While this method enables tangible inputs that are easy to make and use, it requires the objects to be in contact with the screen, and typically requires the user to be actively touching the object in order to track and identify it.

To avoid the problem of tangibles occluding the screen, some projects use additional electronics hardware to track objects nearby or attached to the device [12, 49]. While effective, the complexity of this approach puts it out of reach of most everyday users. More easily implementable are methods using sensors already built into mobile devices.

Most mobile devices include a magnetometer, and multiple researchers have proposed methods to track the location of permanent magnets in relation to the device [2, 5, 15, 17]. This method works well for a single interaction object, but as magnetometers can only report a cumulative magnetic field measurement, becomes more complex with multiple objects. Adding more magnetometers [10, 30] or using active electromagnets [11] increases the number of tracked objects but requires custom electronics.

Mobile devices can also sense vibration at a wide range of frequencies. Widgets used the accelerometer to enable a thin, modular widget system that differentiated between controls via characteristic vibrations generated by gear teeth within each control [58]. This approach enabled a variety of widgets that can be reconfigured at

will, but required users to train the system for each widget location and grasp type. In addition, differentiating between identical widgets in close proximity required modifying the widget's mechanism. Acoustruments [23] used the phone's microphone to track how attached physical controls modified sound emitted from the device speaker, but also required training, and supported a limited number of widgets. ClipWidgets requires no training and works with any number of identical widgets in any location.

Another approach is to use computer vision to track objects near the device. Some work in this area has investigated tracking unconstrained objects in a space surrounding the device [3, 27, 56] which requires a large distance between the camera(s) and the objects in order to achieve a wide enough field of view, making these approaches less practical for physical controls.

Several projects have used built-in phone cameras to enable off-screen interaction. CamTrackPoint uses a small 3D-printed attachment that enables the camera to act like a laptop pointing stick. Back-Mirror uses a printed reference pattern mounted on the back of a phone along with a mirror redirecting the camera view to the pattern to enable finger tracking in a small area of the phone's rear surface [57]. More similar to ClipWidgets, two systems detect physical widgets via a phone camera. Prismodule used total internal reflection in a thin acrylic template attached to the back of a phone to detect striped patterns on buttons inserted in the template [21]. While simple, the method is limited to a small number of widgets in one area of the rear of the phone, and widgets placed behind others cannot be detected. Finally, Matsushima et al. used a mirror to redirect the camera view towards color-coded buttons mounted on the back of the phone [29]. Because the mirror was flat, the system could only see buttons in one area of the phone. ClipWidgets expands on this concept to enable a wider variety of widgets that can be placed anywhere around the periphery of the phone.

3 DESIGN AND IMPLEMENTATION

ClipWidgets is a set of tangible widgets that users can clip on to the edges of a custom smartphone case. We implemented buttons, dials, and sliders, but the working principle can be extended to many other types of interactive elements. When a user manipulates a widget, a set of gears inside the widget transforms the interaction force into rotation of an associated marker inside the case. The smartphone's camera detects this motion and performs an associated action. During normal use, the widgets are securely fixed in place to one of the edges of the phone case, but are easily reconfigured by removing them and clipping them to a new location. The case incorporates a conical mirror to enable a view of widgets distributed to any location on the case edge. Figure 2 shows an overview of the system. In the sections below, we describe each of the key components in detail.

3.1 Widgets

Widgets are the main building blocks of ClipWidgets; each consists of a movable interaction component and a mechanism that translates that component's motion into marker rotation. The working principle behind the widget design is based on translating the force of the user's interaction into detectable marker rotation. In order to be in view of the camera, the markers (described in more detail

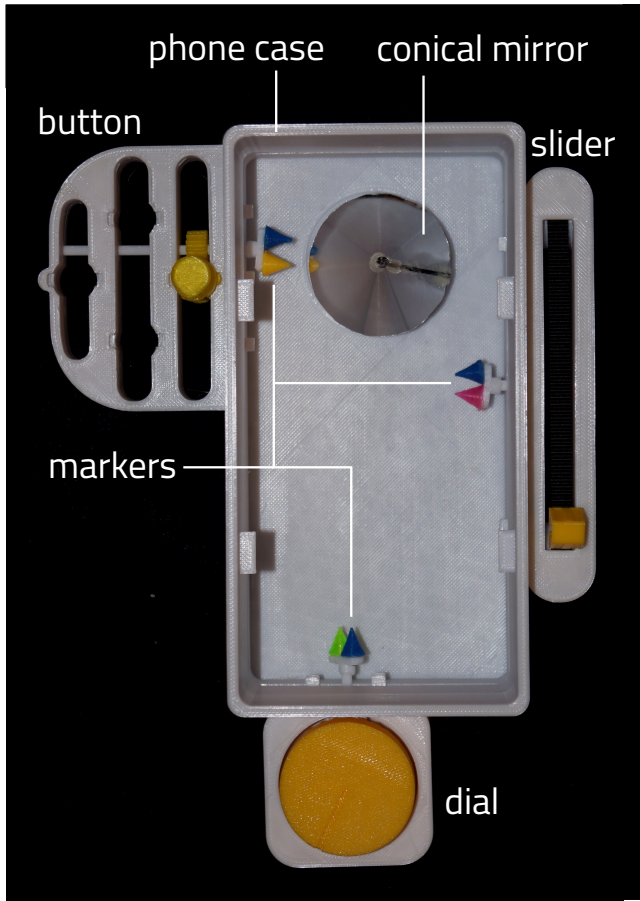


Figure 2: The design and working principle of ClipWidgets. ClipWidgets is comprised of three types of tangible widgets (button, dial, and slider), a smartphone case with conical mirror, and visual markers. When interacting with the widgets, the markers rotate. The rear camera of the phone can detect this rotation through the conical mirror.

in section 3.3) are located inside the case. Each marker is attached to a shaft leading into the body of the widget. Inside the widget, a system of gears translate the off-axis motion caused by widget manipulation into shaft rotation. The exact arrangement of gears depends on the type of movement that user interaction causes. The interface and mechanism for each widget are encased in a frame. The frame provides clips which enable users to easily attach each widget to the slots on the sides of the case.

To illustrate the possibilities of ClipWidgets, we designed three different widgets with three different motion types: a button, a dial, and a slider, each described in detail below.

3.1.1 Button widget. The button widget (Figure 3) consists of four printable parts: a button, a spur gear, a marker shaft and a frame. The button itself contains three important components (Figure 3a): the pressing interface, a spring, and a rack-gear shaft. When users depress the button, the internal rack gear turns the spur gear attached to the marker shaft, translating the linear pressing motion

into rotation (Figures 3c and 3d). When users release the pressing interface, the spring under the interface returns to its rest position, resetting the interface.

The vertical movement of each button is constrained to 3.5 mm, reducing strain on the spring and enhancing the feel of the button. The spur gear has a circumference of 31.5 mm. This means that upon a button press, the marker rotates at most 40°. Our software registers a button press when the associated marker moves by at least 20°. We measured the input delay by analyzing video footage recorded at 30 frames per second (33 ms per frame). We found from 5 observed button presses that the average input delay is 273 ms with a standard deviation of 43 ms. The input delay for the button widget is slightly longer because of the distance the button has to travel before registering.

The frame (Figure 3b) holds all the components together and provides clips to attach the widget to the phone case. It has four slots, enabling up to four buttons to be attached to a single frame. The button frame is 80 mm long, 57 mm wide (excluding the 7 mm attachment clips) and 18 mm high. When the buttons are inserted into the frame, they protrude by 18.5 mm (Figure 3c).

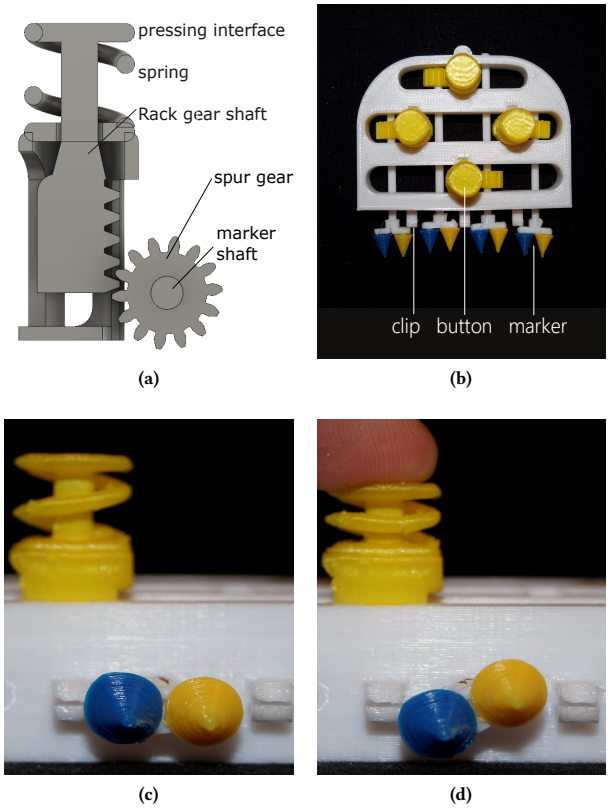


Figure 3: Cross-section view of the button widget (a) and four button widgets assembled in a frame (b). The bottom two images show the different angles of the marker when released (c) and pressed (d).

3.1.2 Dial widget. While the button widget translates linear pressing motion to marker rotation, the dial widget (Figure 4) converts rotation in one plane (parallel to the phone screen) into another (perpendicular to the phone screen). To do so, it uses a crown gear incorporated into the bottom of the dial to rotate a spur gear at the end of the marker shaft (Figure 4a). Unlike the button’s limited range of motion, the dial—and therefore the marker—can spin freely in either direction. The crown gear and spur gear have a gear ratio of 3:1, so every full rotation of the dial causes three rotations of the marker. We found from 5 observed dial inputs that the average input delay is 113 ms with a standard deviation of 18 ms.

The dial widget is made up of four printed components: a frame, the dial knob, a spur gear and the marker shaft. The dial widget is 50 mm wide and long (excluding the 7 mm attachment clips) and 26 mm high without the dial knob. The dial knob protrudes 11 mm from the top and has a 40 mm diameter (Figure 4b).

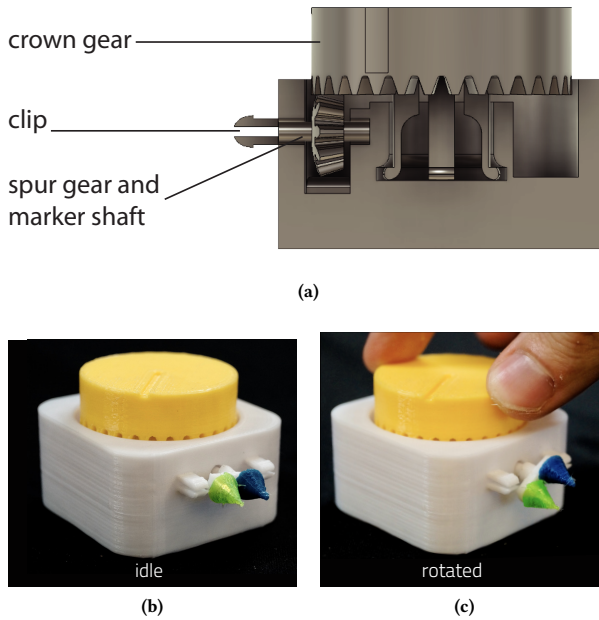


Figure 4: Cross-section of the dial widget schematic (a). Printed dial in two states (b), (c): when users rotate the dial, the crown gear on its bottom rotates the connected spur gear and marker shaft, causing the markers to rotate through 360°.

3.1.3 Slider widget. Similar to the button, the slider widget (Figure 5) converts linear motion to marker rotation; however, the slider’s physical range of motion is much larger than the button’s. We use a rack-and-pinion arrangement (Figure 5a), with the rack functionality provided by a non-printed timing belt. The remaining slider components are printed: a slider thumb attached to the timing belt, a spur gear, a marker shaft, and the assembly frame.

We designed the slider to occupy the full length of the longer dimension of the phone case, resulting in the frame measuring

126 × 20 mm in the plane of the phone screen, with a height of 50 mm. The spur gear attached to the marker shaft is approximately 17 mm in diameter, with 36 teeth matching the 1.5 mm pitch of the timing belt. The slider’s slot allows for a travel of 90 mm, resulting in a maximum marker rotation of about 600° (Figure 5b). We found from 5 observed slider inputs that the average input delay is 133 ms with a standard deviation of 24 ms.

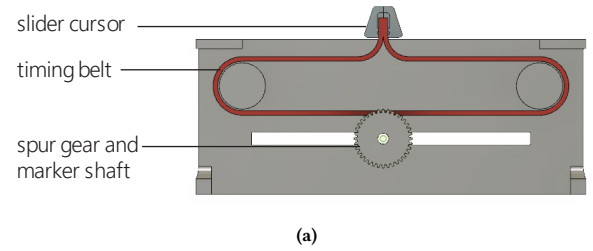


Figure 5: Cross-section of the slider widget schematic (a). Printed slider widget in two states (b). When users move the slider thumb, it pulls the connected timing belt, which rotates the spur gear and attached marker shaft.

3.2 Phone case

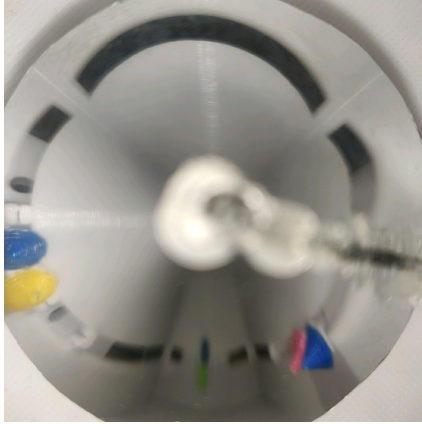
ClipWidgets uses a 3D-printed phone case to enable modular widget attachment and to aid the computer-vision-based interaction detection process. Each of the four edges of the case has 4 mm-wide slots running the length of the edge. These slots enable widgets to be attached at arbitrary points via clips, as described in Widgets.

ClipWidgets tracks widget state via computer vision detection of small markers attached to each widget. To provide line-of-sight from the camera to all possible widget locations, we use a conical mirror positioned on the phone case with the tip of the cone centered on the camera. This mirror gives a 360° view of the case edges (Figure 6). We fabricated the mirror by attaching self-adhesive d-c-fix mirror effect foil¹ to a 3D-printed cone. Our prototype system uses a 18 mm-high cone with a 40 mm diameter that is mounted at 12 mm above the bottom of the case.

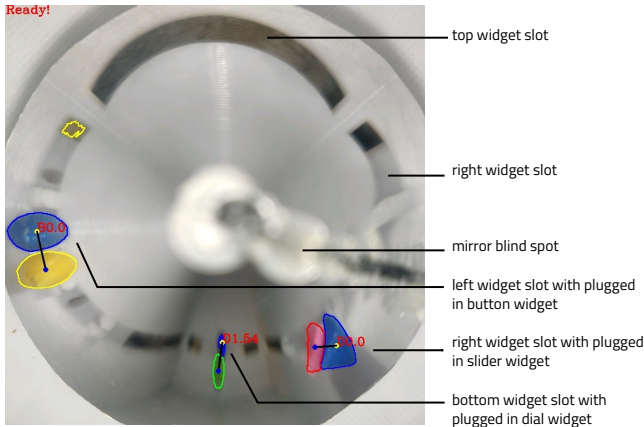
Although a commercially available conical mirror would provide a higher-quality image, our approach enables easy customization for different camera focal lengths. The need to cut the foil to wrap

¹<https://dcfixbrand.com/product/mirror-effect/>

it around the cone (Figure 1a) does create a small blind spot in the camera image (Figure 6); in our prototype, the blind spot is approximately 10° . The mirror is easily detachable from the case such that we can rotate the blind spot to a direction where no widgets are currently attached; for example, one of the corners of the case.



(a)



(b)

Figure 6: Camera view of widget markers with configuration as shown in Figure 2. Image (a) shows the raw camera view. Image (b) labels features in the image and shows widget markers (colored outlines) and angles (black lines and red text) as detected by ClipWidgets.

In our prototype we use a OnePlus 6T smartphone. The phone case is 86 mm wide and 167 mm tall to fully enclose the phone on all sides except the front, and is 70 mm deep to allow for a 48 mm gap between the camera and the bottom of the phone case. This is necessary for the camera² to have a large enough depth of field to keep both the near and far sides of the phone case in enough focus to detect the markers.

²We use the phone's bottom camera, a 16 MP Sony IMX519 with a 4.2 mm focal length and $f/1.7$ lens aperture.

3.3 Markers

To detect widget state we use small, colored markers that protrude from each widget into the inside of the case. User interaction with a widget causes the marker to rotate in a predetermined way, enabling our computer vision software to deduce the status of the widget.

A widget marker consists of two 8 mm-wide and 9 mm-tall cones glued to a small frame (Figure 7). The cones are 3D-printed with different color filaments. The frame holds the cones in place and allows it to be attached to the widget shaft. One of the cones is always blue, and the other cone is a different color to indicate widget identity. By using a blue cone and a different color cone on every widget marker, we can ensure that two cones of the same color are never right next to each other when the computer vision starts tracking them—having two cones of the same color next to each other would make it hard to detect them separately.



Figure 7: The three different markers used by ClipWidgets. The yellow marker is used for buttons, the green marker is used for dials and the pink marker is used for sliders. This color mapping was chosen arbitrarily

A widget marker rotates when the user interacts with the widget. To determine the state of the widget, the computer vision algorithm calculates the angle from the blue cone to the secondary cone, relative to the horizontal axis of the camera image. Figure 6 shows three different widget markers and their detected angles.

We use cone-shaped markers to achieve the most consistent sensing results with our computer vision algorithm. The problem with other shapes is that when widget is placed near the bottom on the long sides of the case, the camera struggles to see both at the same time when the widget marker is in a horizontal position. This can be seen in Figure 6 where the blue cone of the slider widget marker almost obstructs the camera's visual of the pink cone. We also tried a cylinder shape as well as a flat marker. Using the cylinder resulted in worse obstruction and using the flat marker made it hard to sense the marker at all, due to the angle with respect to the camera.

3.4 Widget stacking

To provide the greatest range of motion possible, our slider widgets extend over the entire length of one edge of the phone case, preventing other widgets from attaching. ClipWidgets solves this problem by providing the capability to stack widgets via a 4 mm-wide slot

identical to that in the phone case, opposite to the widget’s attachment clip. A second widget can clip into the first widget’s slot. By using a longer marker shaft that goes through the first widget into the case, we can sense its state as if it was directly plugged into the case. Figure 1c illustrates two sliders stacked together, and the shaft pass-through hole can be seen in Figure 5.

3.5 Recognition software

To analyze the camera footage, we developed computer vision software for our OnePlus 6T running on Android 10. We used OpenCV 3.4.3³ for Java to process the camera images, captured at 400×400 pixels and 30 FPS. We use the LED flash on the phone to illuminate the inside of the case for consistent color detection.

Although the conical mirror forms an anamorphic image of the interior of the phone case [16], we found de-warping the image to be unnecessary, and thus speed computation by working directly with the warped images. To detect markers, the software first converts the image from RGB to HSV color space, allowing us to segment the image by hue. We then threshold the image for the four marker colors and use OpenCV’s contour detection to find the boundary of each individual colored cone. Finally we pair the centroid of each detected blue object (the first cone on each marker) with the center of the closest non-blue object (the second cone on each marker). The software sometimes recognizes colored objects where there are no widget markers, due to camera noise and light coming through the widget slots. We filter these out by removing any detected objects that are inconsistent in size and shape during the first 100 frames.

To determine the state of the widgets we calculate the angle of the line connecting each pair of marker centroids relative to their initial angle. To prevent video noise from affecting the calculated marker angles, we round each angle to multiples of 3.6° . This removes most of the noise, while still providing 100 different values for each widget marker.

For widgets with a limited range, we assume that the initial state of each widget is its minimum value. Practically, this means sliders are at the bottom and buttons are not pressed. This provides the software with an initial status for these widgets, allowing it to unambiguously track marker state—even through multiple rotations, as with the slider. We currently only track the knob’s relative movement (i.e., increasing or decreasing), but could also use it in absolute mode by requiring a zero location for it as well.

3.6 Fabrication and assembly

Except for the timing belt used in the slider widget and the foil used for the mirror, all of our system components are fabricated on an Ender 3 Pro 3D printer, a low-cost consumer-grade printer. We use PLA filament in multiple colors. To simplify production, most of the components are fabricated in multiple parts and assembled post-print; with one exception, once assembled they do not need to be disassembled again. The exception is the marker-shaft connection: the markers are wider than the clip slot, so each needs to be removed from its drive shaft before removing a widget from the case, and reattached after clipping a widget to a new location.

³<https://opencv.org>

4 APPLICATIONS

In this section, we show three applications of ClipWidgets to demonstrate its reconfigurability and generalizability. Each application draws from the same set of 3D-printed physical widgets, but uses them in different ways.

4.1 Game Controller

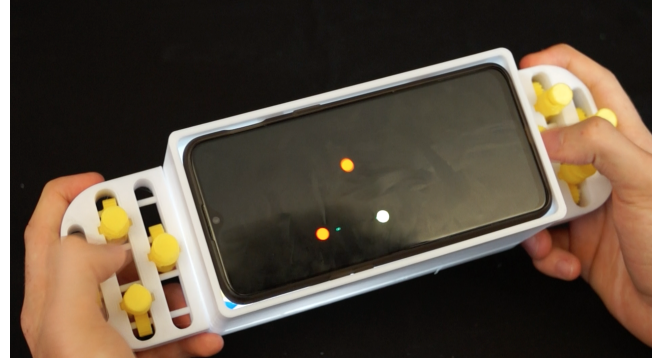


Figure 8: Game controller prototype consisting of two sets of four attachable buttons. The left four buttons are used to control the position of the player (white dot on the screen), and the right four buttons are used to shoot enemies (orange dots) in four directions.

We developed a video game controller (Figure 8) to illustrate how ClipWidgets allows a non-occluding physical interface for a mobile device. The controller consists of two sets of four buttons, each laid out in a +-shape, as direction pads. We developed a simple game to test the usability of the controller. The buttons on the left move the player in four directions and the buttons on the right shoot bullets at enemies in four directions. The interface allows users to experience the game without occluding the screen with fingers or controllers, and without purchasing special-purpose controller hardware.

4.2 Music Interface

We developed a music interface (Figure 9) to show the reconfigurability of ClipWidgets. Audio professionals often prefer physical and reconfigurable controls to support their work [19]. While other projects [8] and products [31] provide special-purpose reconfigurable mixer hardware, ClipWidgets does not require devices other than the ubiquitous mobile phone.

The music interface consists of two sliders, two dials and a button. We also developed a simple mixing app to test the use of the widgets. The app allows the user to play two separate audio tracks. The sliders control the volume and the dials control the tempo of the two tracks. Finally, pressing the button either plays or pauses both tracks. To demonstrate the reconfigurability of ClipWidgets we created two examples of how the user could build the music interface. In one example, the widgets are separated by function, having the tempo control on the left (dials) and the volume control on the right (sliders). In the other example, the widgets are separated by track, having both a slider and a dial on each side.



Figure 9: Two designs of the music interface. In both cases the two sliders control the volume and the two dials control the tempo of two separate tracks. The button on top plays/pauses both tracks. The screen shows the camera output and the current parameters of the tracks. The interface on the left (a) groups widgets by functionality whereas the interface on the right (b) groups widgets by track.

4.3 Interactive Graph Tool

We developed an interface to interact with math functions (Figure 10) to demonstrate how a ClipWidgets prototype can be used for tangible exploration of a high-dimensional space. We developed a prototype that contains three buttons, three dials and a single slider. We also developed an app that plots three math functions each containing four variables: $f_1(x) = a(x + b) + c$, $f_2(x) = ax^b + c$ and $f_3(x) = \sin(ax + b) * c$. The buttons are used to select the three math functions. The dials control the variables a , b and c of the selected math function. Finally the slider is used to zoom in and out on the x -axis of the graph.

5 LIMITATIONS

One of the strengths of ClipWidgets is its modularity and reconfigurability. The widgets can easily be added to and removed from a prototype. However, the phone case does not share this characteristic and must be customized for a given phone. Furthermore, modern smartphones use a wide variety of camera sensors mounted in different locations, making it difficult to develop a general solution with regards to both hardware and software.

The recognition software takes the angle between two detected cones on a plane. However, a rotation of the marker does not correspond linearly to the observed angle between the cones when observed from an angle. This means that as the widgets get closer to the bottom of the case, where the viewing angle gets narrower, the behavior of the widgets gets less adequate. We believe that in a future version of the recognition software, this could be solved by automatically correcting for the observed angle of the marker.

Because the widget markers are inside a sealed case, we use the phone's LED light to provide illumination for computer vision. While this consistent source provides ideal lighting, it does incur some additional battery drain.

Another limitation of the widgets is the need for a physical movement to detect interaction. This requirement prevents us from developing widgets that are, for example, touch sensitive or able to

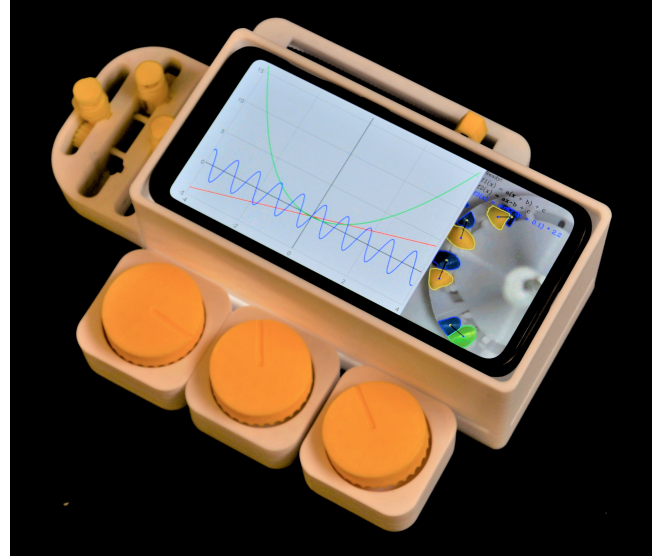


Figure 10: Interactive graph, the buttons are used to select the current function. The dials control the a , b and c parameters of the functions in the graph. The slider is used to zoom in and out on the x -axis.

sense sounds. While many of these kinds of events can be sensed by the phone itself, the necessity for widget movement prevents us from providing physical controls based on them.

While the 3D-printed nature of ClipWidgets gives it versatility and flexibility, it also brings the corresponding limitations of hobby-grade 3D printers. The widget components are too small to be printed in one piece, are not perfectly smooth, and do not perfectly connect to each other, causing friction in movement and wiggle between parts. These imperfections lead to a feeling of lower quality than with non-printed objects, and in the case of the game controller, caused computer vision errors during particularly exciting portions of the game due to shaking of the case and markers.

The mechanical nature of ClipWidgets imposes fundamental constraints on widget design. We cannot make the widgets smaller than their interior mechanisms, and they must be able to generate enough force to drive the rotary motion of the markers.

Finally, while the widgets could avoid occlusion, they reduced the directness of manipulation comparing to onscreen conductive interfaces. We believe this problem can be address by having content on the screen that correspond to the position and type of the widgets.

6 FUTURE WORK

Our demonstrator widgets illustrate that ClipWidgets can sense the traditional interactions of pressing, sliding and rotating, but the technique of detecting marker rotation could be used for other kinds of interaction as well. For example, squeezing a shape printed with flexible filament could cause airflow across vanes, which would cause a marker to rotate. Non-rotational changes to markers might be detectable as well; for example, using thermochromic materials to print the markers could indicate temperature change.

In this paper we explored further modularity of ClipWidgets through widget stacking. However, we only explored horizontal stacking of widgets. In principle, we could stack widgets at any angle or orientation, as long as we are able to print some kind of “transmission” to allow interaction force to rotate the markers.

Our focus in this work was to test and illustrate the main concept of ClipWidgets: modular physical widgets with interaction detected by the device camera. Our system does not currently include a 3D design tool or a phone API, limiting its accessibility to non-expert users. Although the shafts and markers are standard between widgets, and the gear systems versatile enough to re-use in different designs, entirely new widgets still require familiarity with 3D design. In the future, we hope to develop a semi-automated design assistant to create printable gear trains (perhaps extending work by Coros *et al.* [13]). Likewise, a visual programming tool similar to Pineal’s [24] would enable people without programming ability to also use ClipWidgets.

We would also like to improve our computer vision system such that we can reduce the size of the widget markers enough to allow them to fit through the widget slots, making ClipWidgets truly plug-and-play. Embedding magnets into the phone case and widgets could also simplify the process of adding and removing widgets, at the cost of slightly increasing the complexity of assembly.

Lastly, we want to test ClipWidgets with designers and other end users. We plan to run tests where users develop their own prototypes. From these tests we hope to find out how much ClipWidgets assists users in prototyping and learn what aspects to further improve.

7 CONCLUSION

This paper introduces ClipWidgets, a system that allows users to easily create tangible user interfaces around a smartphone. ClipWidgets use the smartphone’s rear camera and computer vision to sense the state of a set of 3D printed widgets attached to it. Our implementation of ClipWidgets provides three different widgets, a button, dial and slider. We demonstrated ClipWidgets’s functionality through three demos showcasing its reconfigurability and generalizability. In future work we hope to explore widgets that can sense less traditional input like temperature or wind speed and improve the overall robustness of ClipWidgets.

ACKNOWLEDGMENTS

We would like to thank Valkyrie Savage for providing valuable feedback on the writing of this paper.

REFERENCES

- [1] Jason Alexander, Anne Roudaut, Jürgen Steimle, Kasper Hornbæk, Miguel Bruns Alonso, Sean Follmer, and Timothy Merritt. 2018. Grand Challenges in Shape-Changing Interface Research. In *CHI '18: Proceedings of the 36th Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 299–314. <https://doi.org/10.1145/3173574.3173873>
- [2] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nanya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2043–2046. <https://doi.org/10.1145/1978942.1979238>
- [3] Daniel Avrahami, Jacob O. Wobbrock, and Shahram Izadi. 2011. Portico: tangible interaction on and around a tablet. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. Association for Computing Machinery, New York, NY, USA, 347–356. <https://doi.org/10.1145/2047196.2047241>
- [4] Patrick Baudisch, Torsten Becker, and Frederik Rudeck. 2010. Luminos: Tangible 3D Interaction on Tabletop Computers Based on Blocks Made from Glass Fiber Bundles. *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, 1–10.
- [5] Andrea Bianchi and Ian Oakley. 2013. Designing tangible magnetic accessories. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. Association for Computing Machinery, New York, NY, USA, 255–258. <https://doi.org/10.1145/2460625.2460667>
- [6] Mads Bock, Martin Fisker, Kasper Fischer Topp, and Martin Kraus. 2015. Initial Exploration of the Use of Specific Tangible Widgets for Tablet Games. In *Social Informatics (Lecture Notes in Computer Science)*, Luca Maria Aiello and Daniel McFarland (Eds.). Springer International Publishing, Cham, 183–190. https://doi.org/10.1007/978-3-319-15168-7_23
- [7] Mads Bock, Martin Fisker, Kasper Fischer Topp, and Martin Kraus. 2015. Tangible Widgets for a Multiplayer Tablet Game in Comparison to Finger Touch. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '15)*. Association for Computing Machinery, New York, NY, USA, 755–758. <https://doi.org/10.1145/2793107.2810269>
- [8] Florian Born. 2021. *Modulares interface*. Retrieved Aug 2, 2021 from <https://florianborn.com/projects/modulares-interface>
- [9] Liwei Chan, Stefanie Müller, Anne Roudaut, and Patrick Baudisch. 2012. Cap-Stones and ZebraWidgets: sensing stacks of building blocks, dials and sliders on capacitive touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2189–2192. <https://doi.org/10.1145/2207676.2208371>
- [10] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D Input Using Two Magnetic Sensors. In *UIST '13 Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. ACM, 237–244. <https://doi.org/10.1145/2501988.2502035>
- [11] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. 2016. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1504–1514. <https://doi.org/10.1145/2858036.2858125>
- [12] Chin-yu Chien, Rong-Hao Liang, Long-Fei Lin, Liwei Chan, and Bing-Yu Chen. 2015. FlexiBend: Enabling Interactivity of Multi-Part, Deformable Fabrications Using Single Shape-Sensing Strip. In *UIST '15: Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New York, New York, USA, 659–663.
- [13] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational Design of Mechanical Characters. *ACM Transactions on Graphics* 32, 4 (July 2013), 83:1–83:12. <https://doi.org/10.1145/2461912.2461953>
- [14] Sebastian Günther, Martin Schmitz, Florian Müller, Jan Riemann, and Max Mühlhäuser. 2017. BYO*: Utilizing 3D Printed Tangible Tools for Interaction on Interactive Surfaces. In *Proceedings of the 2017 ACM Workshop on Interacting with Smart Objects (SmartObject '17)*. Association for Computing Machinery, New York, NY, USA, 21–26. <https://doi.org/10.1145/3038450.3038456>
- [15] Chris Harrison and Scott E Hudson. 2009. Abracadabra: Wireless, High-Precision, and Unpowered Finger Input for Very Small Mobile Devices. *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology* (Oct. 2009), 121–124. <http://portal.acm.org/citation.cfm?id=1622176.1622199>
- [16] James L Hunt, BG Nickel, and Christian Gigault. 2000. Anamorphic images. *American Journal of Physics* 68, 3 (2000), 232–237.
- [17] Sungjae Hwang, Myungwook Ahn, and Kwang-yun Wahn. 2013. MagGetz: customizable passive tangible controllers on and around conventional mobile devices. In *Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13)*. Association for Computing Machinery, New York, NY, USA, 411–416. <https://doi.org/10.1145/2501988.2501991>
- [18] Michael D Jones, Kevin Seppi, and Dan R Olsen. 2016. What You Sculpt Is What You Get: Modeling Physical Interactive Devices with Clay and 3D Printed Widgets. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 876–886. <https://doi.org/10.1145/2858036.2858493>
- [19] Hyunyoung Kim, Céline Coutrix, and Anne Roudaut. 2018. *KnobSlider: Design of a Shape-Changing UI for Parameter Control*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3173574.3173913>
- [20] Hyunyoung Kim, Patrícia Deud Guimarães, Céline Coutrix, and Anne Roudaut. 2019. ExpandDial: designing a shape-changing dial. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. 949–961.
- [21] Takuya Kitade and Wataru Yamada. 2019. Prismodule: Modular UI for Smartphones Using Internal Reflection. In *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology (New Orleans, LA, USA) (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 119–121. <https://doi.org/10.1145/3332167.3356892>

- [22] Scott R Klemmer, Björn Hartmann, and Leila Takayama. 2006. How bodies matter: five themes for interaction design. In *Proceedings of the 6th conference on Designing Interactive Systems*. 140–149.
- [23] Gierad Laput, Eric Brockmeyer, Scott E Hudson, and Chris Harrison. 2015. Acoustuments: Passive, Acoustically-Driven, Interactive Controls for Handheld Devices. In *CHI '15: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 2161–2170. <https://doi.org/10.1145/2702123.2702414>
- [24] David Ledo, Fraser Anderson, Ryan Schmidt, Lora Oehlberg, Saul Greenberg, and Tovi Grossman. 2017. Pineal: Bringing Passive Objects to Life with Embedded Mobile Devices. In *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 2583–2593. <https://doi.org/10.1145/3025453.3025652>
- [25] Dingzeyu Li, David I W Levin, Wojciech Matusik, and Changxi Zheng. 2016. Acoustic Voxels: Computational Optimization of Modular Acoustic Filters. *ACM Transactions on Graphics (TOG)* 35, 4 (July 2016), 88–12. <https://doi.org/10.1145/2897824.2925960>
- [26] Robert MacCurdy, Robert Katzschmann, Youbin Kim, and Daniela Rus. 2016. Printable Hydraulics: A Method for Fabricating Robots by 3D Co-Printing Solids and Liquids. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3878–3885. <https://doi.org/10.1109/ICRA.2016.7487576>
- [27] Sebastián Marichal, Anadrea Rosales, Fernando Gonzalez Perilli, Ana Cristina Pires, Ewelina Bakala, Gustavo Sansone, and Josep Blat. 2017. CETA: Designing Mixed-Reality Tangible Interaction to Enhance Mathematical Learning. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3098279.3098536>
- [28] Karola Marky, Martin Schmitz, Verena Zimmermann, Martin Herbers, Kai Kunze, and Max Mühlhäuser. 2020. 3D-Auth: Two-Factor Authentication with Personalized 3D-Printed Items. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3313831.3376189>
- [29] Nobutaka Matsushima, Wataru Yamada, and Hiroyuki Manabe. 2017. Attaching Objects to Smartphones Back Side for a Modular Interface. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology (Québec City, QC, Canada) (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 51–52. <https://doi.org/10.1145/3131785.3131810>
- [30] Jess McIntosh, Paul Strohmeier, Jarrod Knibbe, Sebastian Boring, and Kasper Hornbæk. 2019. Magnetips: Combining Fingertip Tracking and Haptic Feedback for Around-Device Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300638>
- [31] Monogram. 2021. *Monogram: Modular productivity tool for creative pros*. Retrieved Aug 2, 2021 from <https://monogramcc.com/>
- [32] Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. 2013. Touch & Activate: Adding Interactivity to Existing Objects Using Active Acoustic Sensing. In *UIST '13 Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, New York, USA, 31–40. <https://doi.org/10.1145/2501988.2501989>
- [33] Huaishu Peng, François Guimbretière, James McCann, and Scott Hudson. 2016. A 3D Printer for Interactive Electromagnetic Devices. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*. ACM Press, Tokyo, Japan, 553–562. <https://doi.org/10.1145/2984511.2984523>
- [34] Huaishu Peng, Jennifer Mankoff, Scott E Hudson, and James McCann. 2015. A Layered Fabric 3D Printer for Soft Interactive Objects. In *CHI '15: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 1789–1798. <https://doi.org/10.1145/2702123.2702327>
- [35] Michael L Rivera, Melissa Moukperian, Daniel Ashbrook, Jennifer Mankoff, and Scott E Hudson. 2017. Stretching the Bounds of 3D Printing with Embedded Textiles. In *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 497–508. <https://doi.org/10.1145/3025453.3025460>
- [36] Valkyrie Savage, Colin Chang, and Björn Hartmann. 2013. Sauron: embedded single-camera sensing of printed physical user interfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, St. Andrews Scotland, United Kingdom, 447–456. <https://doi.org/10.1145/2501988.2501992>
- [37] Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. 2015. Makers' Marks: Physical Markup for Designing and Fabricating Functional Objects. In *UIST '15: Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New York, New York, USA, 103–108.
- [38] Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B. Goldman, Gautham Mysore, and Wilmot Li. 2015. Lamello: Passive Acoustic Sensing for Tangible Input Components. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1277–1280. <https://doi.org/10.1145/2702123.2702207>
- [39] Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. Capricate: A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 253–258. <https://doi.org/10.1145/2807442.2807503>
- [40] Martin Schmitz, Andreas Leister, Niloofar Dezfali, Jan Riemann, Florian Müller, and Max Mühlhäuser. 2016. Liquido: Embedding Liquids into 3D Printed Objects to Sense Tilting and Motion. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 2688–2696. <https://doi.org/10.1145/2851581.2892275>
- [41] Martin Schmitz, Florian Müller, Max Mühlhäuser, Jan Riemann, and Huy Viet Viet Le. 2021. Itsy-Bits: Fabrication and Recognition of 3D-Printed Tangibles with Small Footprints on Capacitive Touchscreens. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 419, 12 pages. <https://doi.org/10.1145/3411764.3445502>
- [42] Martin Schmitz, Jürgen Steimle, Jochen Huber, Niloofar Dezfali, and Max Mühlhäuser. 2017. Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1001–1014. <https://doi.org/10.1145/3025453.3025663>
- [43] Martin Schmitz, Martin Stitz, Florian Müller, Markus Funk, and Max Mühlhäuser. 2019. Trilaterate: A Fabrication Pipeline to Design and 3D Print Hover-, Touch-, and Force-Sensitive Objects. In *CHI '19: Proceedings of the 37th Annual SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 454–13. <https://doi.org/10.1145/3290605.3300684>
- [44] Huiying Shen, Owen Edwards, Joshua Miele, and James M. Coughlan. 2013. CamIO: A 3D Computer Vision System Enabling Audio/Haptic Interaction with Physical Objects by Blind Users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (Bellevue, Washington) (ASSETS '13)*. Association for Computing Machinery, New York, NY, USA, Article 41, 2 pages. <https://doi.org/10.1145/2513383.2513423>
- [45] Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. 2016. Tickers and Talker: An Accessible Labeling Toolkit for 3D Printed Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 4896–4907. <https://doi.org/10.1145/2858036.2858507>
- [46] Lei Shi, Yuhang Zhao, and Shiri Azenkot. 2017. Markit and Talkit: A Low-Barrier Toolkit to Augment 3D Printed Models with Audio Annotations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, Québec City, QC, Canada, 493–506. <https://doi.org/10.1145/3126594.3126650>
- [47] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Human-Computer Interaction - INTERACT 2005 (Lecture Notes in Computer Science)*, Maria Francesca Costabile and Fabio Paternò (Eds.). Springer, Berlin, Heidelberg, 267–280. https://doi.org/10.1007/11555261_24
- [48] Ronit Slyper and Jessica Hodgins. 2012. Prototyping Robot Appearance, Movement, and Interactions Using Flexible 3D Printing and Air Pressure Sensors. In *2012 RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 6–11. <https://doi.org/10.1109/ROMAN.2012.6343723>
- [49] Evan Strasnich, Jackie Yang, Kesler Tanner, Alex Olwal, and Sean Follmer. 2017. ShiftIO: Reconfigurable Tactile Elements for Dynamic Affordances and Mobile Interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 5075–5086. <https://doi.org/10.1145/3025453.3025988>
- [50] Carlos Tejada, Osamu Fujimoto, Zhiyuan Li, and Daniel Ashbrook. 2018. Blow-hole: Blowing-Activated Tags for Interactive 3D-Printed Models. In *Proceedings of the 44th Graphics Interface Conference GI Proceedings of the Nd Graphics Interface Conference*. 122–128. <http://graphicsinterface.org/wp-content/uploads/gi2018-18.pdf>
- [51] Carlos E. Tejada, Raf Ramakers, Sebastian Boring, and Daniel Ashbrook. 2020. AirTouch: 3D-printed Touch-Sensitive Objects Using Pneumatic Sensing. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. <https://doi.org/10.1145/3313831.3376136>
- [52] Anke van Oosterhout, Eve Hoggan, Majken Kirkegaard Rasmussen, and Miguel Bruns. 2019. DynaKnob: combining haptic force feedback and shape change. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. 963–974.
- [53] Christian Weichel, Manfred Lau, and Hans Gellersen. 2013. Enclosed: A Component-Centric Interface for Designing Prototype Enclosures. In *TEI '13: Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. <https://doi.org/10.1145/2460625.2460659>
- [54] Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D Hollan, and Jan Borchers. 2009. SLAP Widgets: Bridging the Gap Between Virtual and Physical Controls on Tabletops. In *CHI '09: Proceeding of the Twenty-Seventh Annual SIGCHI Conference on Human Factors in Computing Systems*. 1–10.
- [55] Karl Willis, Eric Brockmeyer, Scott E Hudson, and Ivan Poupyrev. 2012. Printed Optics: 3D Printing of Embedded Optical Elements for Interactive Devices. In *UIST*

- '12: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New York, New York, USA, 589–598. <https://doi.org/10.1145/2380116.2380190>
- [56] Karl D. D. Willis, Takaaki Shiratori, and Moshe Mahler. 2013. HideOut: Mobile Projector Interaction with Tangible Objects and Surfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (Barcelona, Spain) (TEI '13). Association for Computing Machinery, New York, NY, USA, 331–338. <https://doi.org/10.1145/2460625.2460682>
- [57] Pui Chung Wong, Hongbo Fu, and Kening Zhu. 2016. Back-Mirror: Back-of-Device One-Handed Interaction on Smartphones. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications* (Macau) (SA '16). Association for Computing Machinery, New York, NY, USA, Article 10, 5 pages. <https://doi.org/10.1145/2999508.2999522>
- [58] Chang Xiao, Karl Bayer, Changxi Zheng, and Shree K. Nayar. 2019. Widgets: Modular Mechanical Widgets for Mobile Devices. *ACM Trans. Graph.* 38, 4, Article 100 (July 2019), 12 pages. <https://doi.org/10.1145/3306346.3322943>
- [59] Neng-Hao Yu, Sung-Sheng Tsai, I-Chun Hsiao, Dian-Je Tsai, Meng-Han Lee, Mike Y. Chen, and Yi-Ping Hung. 2011. Clip-on gadgets: expanding multi-touch interaction area with unpowered tactile controls. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. Association for Computing Machinery, New York, NY, USA, 367–372. <https://doi.org/10.1145/2047196.2047243>
- [60] Clement Zheng, Jeeun Kim, Daniel Leithinger, Mark D. Gross, and Ellen Yi-Luen Do. 2019. Mechamagnets: Designing and Fabricating Haptic and Functional Physical Inputs with Embedded Magnets. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '19). Association for Computing Machinery, New York, NY, USA, 325–334. <https://doi.org/10.1145/3294109.3295622>