

# MixFab: A Mixed-Reality Environment for Personal Fabrication

Christian Weichel<sup>1</sup> Manfred Lau<sup>1</sup> David Kim<sup>2</sup> Nicolas Villar<sup>2</sup> Hans Gellersen<sup>1</sup>

<sup>1</sup>Lancaster University  
Lancaster, United Kingdom

{c.weichel, m.lau, h.gellersen}@lancaster.ac.uk

<sup>2</sup>Microsoft Research  
Cambridge, United Kingdom

{b-davidk, nvillar}@microsoft.com

## ABSTRACT

Personal fabrication machines, such as 3D printers and laser cutters, are becoming increasingly ubiquitous. However, designing objects for fabrication still requires 3D modeling skills, thereby rendering such technologies inaccessible to a wide user-group. In this paper, we introduce *MixFab*, a mixed-reality environment for personal fabrication that lowers the barrier for users to engage in personal fabrication. Users design objects in an immersive augmented reality environment, interact with virtual objects in a direct gestural manner and can introduce existing physical objects effortlessly into their designs. We describe the design and implementation of *MixFab*, a user-defined gesture study that informed this design, show artifacts designed with the system and describe a user study evaluating the system's prototype.

## Author Keywords

personal fabrication; direct manipulation; 3D printing; mixed-reality; 3D modeling

## ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: User Interfaces - Interaction styles

## General Terms

Design; Human Factors

## INTRODUCTION

Personal fabrication of 3D objects has been spurred by rapid advances in printing tools and techniques. The quality, speed, capability and ease of use of 3D printers is rapidly increasing while cost is going down, enabling wider audiences as “makers” of physical objects. The outcome of the process is distinguished by the immediate presence it has in the real world - physical, tangible, and usable in a real world sense. The front-end of the process is in stark contrast; the tools that help users design things they would like to make still reside on flat computer screens.

In this paper, we introduce a new system, *MixFab*, to support design for personal fabrication with a mixed reality approach.

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](mailto:permissions@acm.org).  
*CHI 2014*, April 26 – May 1, 2014, Toronto, Ontario, Canada.  
Copyright © 2014 ACM 978-1-4503-2473-1/14/04..\$15.00.  
<http://dx.doi.org/10.1145/2556288.2557090>

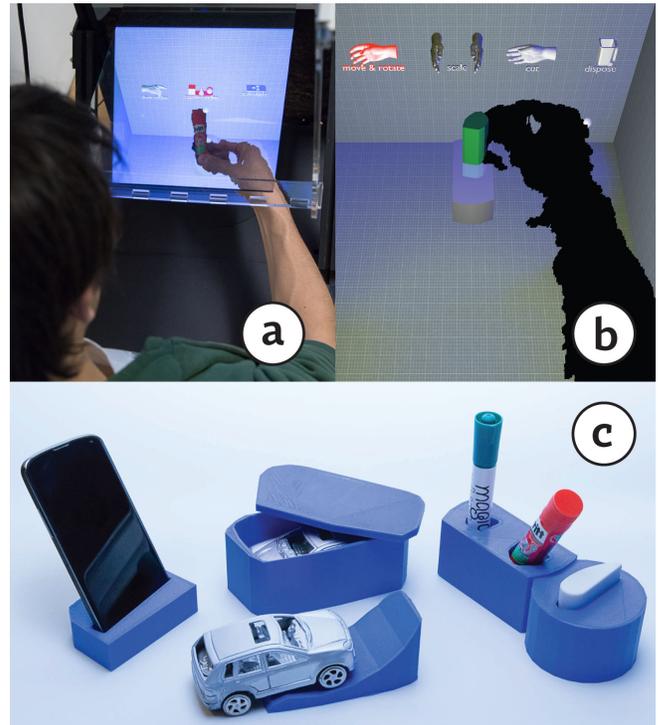


Figure 1. *MixFab*: mixed-reality environment for personal fabrication. (a) a user positioning a physical object in the *MixFab* prototype system, (b) screenshot of a user manipulating a virtual object, (c) 3D printed objects created with the system.

The system is aimed to lower the barrier for casual design of 3D content, and to enable a design experience where the digital model of an artifact-to-be-made is created in a physical work space that affords direct manipulation and interaction with artifacts that are real. To this end, the system integrates three core concepts: (1) use of immersive augmented reality to provide a 3D visualization of the artifact under construction projected in the real world; (2) support for users to shape artifacts directly with their hands, replacing the need for advanced modeling skills with intuitive gestures; (3) enabling use of real artifacts in the design process such that new artifacts can be shaped to fit existing ones.

*MixFab*'s mixed-reality environment is by virtue of a Holodesk-like structure [11] where the user sees virtual content merged with the real world. Users can introduce physical artifacts as size-reference or to capture their shape – Figure 1a shows a user placing a glue-stick inside a virtual object to create the glue-stick's virtual replica in place. In Figure 1b

the just created glue-stick replica (green object) is manipulated as if it was still a physical entity; by virtue of gesture recognition, users can directly manipulate virtual and physical objects alike. Hands and other physical artifacts properly occlude other objects and face-tracking provides a parallax-corrected image, creating important depth-cues.

We make four contributions in this paper. First we propose and implement an immersive mixed-reality environment by combining an augmented reality setup, gesture recognition and 3D scanning capabilities. Our second contribution is a set of user-defined gestures for 3D modeling obtained through a study in which we observe how users would perform basic tasks unconstrained by any system or augmentation. We then present MixFab's design environment, which is based on these gestures. It is centered around direct and natural interaction with virtual artifacts, effortless integration of physical objects into the design process and a self-explanatory interface. Fourth, we perform a user study evaluating the design decisions of MixFab and providing evidence that especially the effortless integration of existing objects is of value.

## RELATED WORK

MixFab is related to personal fabrication, mixed-reality modeling environments and gestural interaction.

### Non-expert fabrication systems

Several semi-commercial products seek to enable non-engineers to design objects for personal fabrication machines, typically by providing a feature-reduced version of traditional CAD systems [1, 2]. Others reduce the complexity of the modeling task by specializing on a specific purpose. For example, SketchChair [23] lets users draw chairs and produces the patterns required to fabricate them, while Plushie [18] follows a similar idea for stuffed toys.

### Personal fabrication with existing objects

Using existing objects in the design process has been explored in a variety of ways. Follmer et al. focus on mixing objects to create new ones. KidCAD [10] lets children combine the 2.5D shape of their toys, while CopyCAD [9] enables the reuse of 2D shapes in a CNC milling setting.

Constructable [19] extends the 2D contour reuse by copying textures to workpieces. A camera takes a photo of the texture and transfers it on the part using a laser-cutter. Enclosed [27] uses electronic components as handles and size references during enclosure design. It automatically adds cutouts to the enclosure patterns so that the components can be mounted.

MixFab lets users remix existing objects, use them to construct new shapes and as a physical size reference, combining the features above in one system.

### Mixed-reality modeling environments

Anderson et al. use tangible proxies to create and shape virtual objects [3]. Tangible building blocks that can sense their spatial configuration are used to create the models; a 3D scanner for scanning clay figures is also proposed. The clay based modeling is extended by Sheng et al. [25] who use clay for direct, fine-grained manipulation. Arisandi et al. introduce tangible tools such as a hammer or a knife, for modeling in

an augmented reality (AR) environment [4]. MirageTable is a general-purpose mixed-reality environment targeting collaborative scenarios, that supports physics based interaction with existing physical objects [5].

Designing fabricable objects in an AR setting has been explored. Modeling-in-Context uses a single photo as reference for size, perspective and ratio [16]. Situated Modeling takes this a step further by stamping primitives into an augmented 3D space, designing artifacts in their designated place [15].

MixFab provides a mixed-reality environment, in which users can interact naturally without the need to wear props [4, 15] or being constrained by physicality [25].

### Gesture-based systems

Shape creation has been transported into virtual space in a variety of ways. Some let users deform models using both hands [20, 17]; others use the motion of the hand [24] or its curvature [12] to define the shape.

Gestures for 3D manipulation and navigation can be distinguished between symbolic and natural gestures. FingerARtips [7] introduces finger-based augmented reality interaction with haptic feedback. The Perceptive Workbench uses pointing in combination with 3D objects [26]. Kim et al. propose a set of symbolic gestures for 3D modeling [14] by studying the interaction with real world objects. Data Miming follows a natural gesture approach, in that they do not prescribe a set of gestures but record users unconstrained hand movement [12].

Whereas related work often has author-defined gestures, we inform our gesture-set by a user defined gesture study [29].

### Hand pose recognition

Hand pose recovery often relies on user augmentation. Surface Drawing utilizes gloves [24], others use reflective markers [14]. Oka et al. [21] require only a camera in the environment and no user augmentation. For an overview of hand pose recognition techniques, see [22]. We implemented an appearance based approach using a single depth-camera, requiring no user-worn equipment or prior calibration.

### MIXFAB

At MixFab's core is an immersive mixed-reality system creating a high permeability between the virtual and physical world. It enables new and exciting interactions that were not possible with each component taken by itself.

We implemented MixFab's physical configuration by building upon the Holodesk frame [11], although other hardware implementations may also be used. The setup superimposes virtual content with the real world by using a beam-splitter and a display mounted at a 45 degree angle. It provides an interaction volume roughly the size of modern 3D printers. A depth camera placed at the top of the frame provides data for interaction within the system. We further add a motorized turntable, to the bottom of the frame, for 3D scanning.

Our processing pipeline is designed to specifically support seamless interaction between virtual and physical objects, blurring the border between the two. The main components of this pipeline are:

**Gesture Recognition** which is solely based on the depth-data provided by the Kinect serves as input modality for the interaction with virtual objects. It does not require any user-augmentation or prior calibration.

**3D Shape Acquisition** is supported at a trade-off between time and precision. One can capture the rough shape of an object in real-time or acquire a more precise scan in about a minute. Physical objects can be captured anywhere in the frame, allowing their placement relative to virtual objects.

**Sketch Recognition** enables users to describe objects they want to create without having to be very precise.

**Mesh data manipulation** serves as back-end for object creation, acquisition and manipulation. We support complex operations (e.g. constrained Delaunay triangulation or plane segmentation) required for object acquisition, and *constructive-solid geometry* operations for shape manipulation which produce 3D printable models.

### USER STUDY: USER-DEFINED 3D MODELING GESTURES

In previous work, hand gestures are typically defined by the respective authors, rather than users [14, 20]. We are interested in what gestures users would intuitively perform to create and manipulate objects, also to inform our subsequent system design. To this end we conduct a user-defined gesture study with a methodology similar to Wobbrock et al. [29].

#### Tasks and Procedure

Each participant was subsequently given a set of tasks (order determined using a balanced latin square). For each task, they were shown one or two images of the desired outcome and asked to perform a gesture to create that desired outcome. Participants were instructed to imagine the objects depicted on the images as being displayed in front of them.

After each gesture, users were asked to rate the gestures suitability and how easy it was to perform, both on a rating scale from 1 (very unsuitable / very hard) to 5 (very suitable / very easy). Once all ten tasks were completed, all users completed a survey querying their age and gender. We further asked for prior CAD experience and how much that experience influenced the proposed gestures (Rating scale, 1 no experience / no influence to 5 a lot of experience / strong influence). Users were seated at a table with a camera placed a meter above the surface, resulting in an interaction area of about  $60 \times 50$  cm.

Upon completion of the study, the recorded video material was transcribed and coded to extract the suggested gestures. Quantitative data collected through the questionnaires, as well as user agreement [29] is used to judge the quality and confidence of users in the proposed gestures.

We invited twelve participants from various departments at our university. Half of the participants were female with age ranging from 19 to 42 ( $M=30.83$  years,  $SD=7.47$ ). Eight users reported no experience with CAD (overall  $M=2.08$ ,  $SD=1.31$ ) and thus little influence of prior experience ( $M=2$ ,  $SD=1.53$ ).

#### Results and Observations

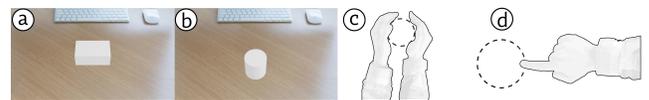
All participants were able to propose a gesture for every task, sometimes more than one in which case they were asked to

report the one they preferred. In the following we report gestures that were proposed by at least two or more participants.

	Task / Gesture	<i>c</i>	<i>s</i>	<i>e / A</i>	
Create Primitives	<b>create box</b>			0.42	
	<i>draw outline</i>	7	3.86	4.57	
	hand boundaries	3	4.00	5.00	
Create Primitives	<b>create cylinder</b>			0.37	
	hand boundaries	6	4.33	4.83	
	<i>draw outline</i>	4	4.00	4.00	
Rotating and Translating	<b>move box</b>			0.33	
	<i>1h grab &amp; move</i>	4	5.00	5.00	
	2h grab & move	4	4.50	5.00	
Rotating and Translating	1h push	4	5.00	5.00	
	<b>rotate box</b>			0.40	
	<i>1h grab &amp; rotate</i>	7	4.57	4.86	
Rotating and Translating	2h grab & rotate	2	5.00	3.50	
	2h rotation	2	4.50	5.00	
	Scale	<b>scale 1 axis</b>			0.71
<i>compression</i>		10	4.80	5.00	
<b>scale 3 axis</b>				0.26	
Scale	<i>compression</i>	5	3.80	4.80	
	corner pinch & resize	3	3.67	4.00	
	Plane Cut	<b>plane cut</b>			0.35
<i>karate</i>		5	4.00	4.20	
knife		4	2.25	4.25	
Plane Cut	points	3	4.33	4.33	
	Scan	<b>object scan</b>			0.17
		<i>dwelt time</i>	4	3.50	4.50
Add Material	<b>add material</b>			0.31	
	<i>stuff</i>	4	4.00	4.50	
	gap trace	4	3.50	4.50	
Add Material	filling tool	2	2.50	3.00	
	Remove	<b>remove object</b>			0.22
		<i>wipe</i>	4	5.00	5.00
move out		3	4.33	5.00	
Remove	smash	2	4.50	4.50	

**Table 1.** List of tasks and corresponding gestures (described by more than one user) during the study. *c* is the count of how many users suggested the gesture, *s* is the reported suitability and *e* is the reported easiness. *A* the agreement among the users as defined by Wobbrock. Gestures recommended for each task group are written in *italics*.

#### Creating primitives



**Figure 2.** (a, b) the pictures shown to study participants (create box/cylinder), (c) *hand boundaries* gesture, (d) *drawing outline* gesture

While creating primitives has no equivalent in the physical world, describing 3D shapes is a common task. When creating boxes, three users choose a method similar to *Data Miming* [12] and define the shape by describing it using their hands (hand boundaries gesture). In case of the cylinder, a

majority preferred to describe the shape using the curvature of their hands. Drawing the outline of the shape with the index finger and extruding it into 3D space was proposed by a majority when creating boxes, and by a third of the participants suggested it for cylinders.

#### Rotating and translating



Figure 3. (a, b) the pictures shown to study participants (Rotate/move box from A to B), (c) two-handed grab and rotate gesture, (d) one-handed grab gesture

Rotating and moving objects are everyday tasks performed using two variants of the same gestures: one-handed vs. two-handed. The pictures shown to the participants contained a keyboard and a mouse next to the virtual object, as a size reference. It seems that the size of virtual object was interpreted differently by each participant, leading to the use of one hand if the object is perceived to be small or both hands if the object is perceived to be large.

#### Scaling



Figure 4. (a, b) the pictures shown to study participants (scale 1DOF/3DOF from A to B), (c) compression gesture, (d) corner pinch & resize gesture

Ten out of twelve users suggested the same gesture: compression. Both hands are placed around the object and changing their distance, changes the objects size.

#### Plane cut

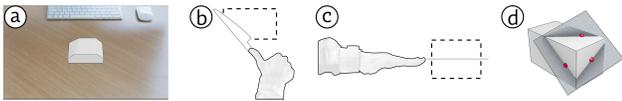


Figure 5. (a) the picture shown to study participants (plane cut of shapes from box), (b) knife gesture, (c) Karate gesture, (d) three points defining a plane

Cutting a slice of an object is a daily task (e.g. cutting a slice of bread or cheese). Users without CAD experience proposed gestures resembling such actions. Four users moved their flat hand or thumb where they wanted to cut, miming a knife. Five users indicated how they wanted to cut by performing a “Shuto” (Knife Hand) motion from Karate. Those experienced in CAD, suggested that one might select three points on the object to define a plane used for cutting.

#### Adding material

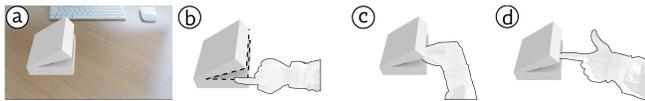


Figure 6. (a) the picture shown to study participants (add material inside gap), (b) gap trace gesture, (c) stuffing gesture, (d) filling tool gesture

Most users suggested a “stuffing motion” as if they were to take a handful of material and put it onto the object. Others

traced the gap they wanted closed with the index finger and confirmed again using a stuffing gesture or by pressing a button. We believe that the “gap-trace” gesture is an artifact of the picture that demonstrated the task: the gap to close was of regular and linear nature. Suggestions would most likely be different when manipulating more organic shapes.

#### Scanning objects



Figure 7. (a, b) the picture shown to study participants (positive scan of existing cup, negative imprint of cup in box), (c) the dwell time action

Transforming physical objects to virtual ones proved to be the most challenging task; to some extent because it is difficult to convey the need of the operation without a system being present. Participants proposed a variety of actions (agreement score: 0.17), but only one was mentioned multiple times: dwell time. Users place the object in the desired position, move their hands away and wait for a certain amount of time.

#### Removing objects

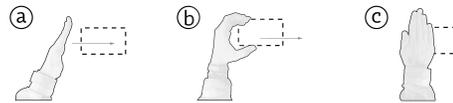


Figure 8. (a) wipe gesture, (b) move out gesture, (c) smash gesture

Removing objects is a daily task. We often throw things away, place them elsewhere or deform them prior to disposal. The gestures suggested for removing objects tend to resemble such actions. *Wipe* and *move out* – the two most prominent gestures – have the same intent: move the object out of the workspace. Wiping objects (moving a hand fast from one side to the other, hitting the object) was suggested more than *move out* and rated easier/more suitable.

#### Discussion

We observe a similar pattern as Wobbrock *et al.* in that user agreement is inversely proportional to the task complexity [29]. More complex tasks (such as object scan) have low user agreement scores, whereas more simple ones (such as rotation or translation) yield higher agreement amongst users. Despite low agreement rates, suitability and easiness remain at high levels, suggesting confidence in the proposed gestures.

To choose an appropriate gestures for each task group, we use the count of how often a gesture was suggested as main metric. In cases where the suggestion count is not distinctive, we decide based on suitability and easiness rating. The gestures recommended for each task are marked in *italic* in table 1.

For most task groups suggestion count, easiness and suitability are sufficient criteria, except for the creation of primitives. When looking at box and cylinder creation separately, we’d be required to choose different gestures for each of them which is undesirable as it would be likely to cause confusion with users. Adding the suggestion counts within the task group however, yields a slight preference for the *draw outline* gesture (11 suggestions vs. 9 for *hand boundaries*).

## USER INTERFACE

MixFab's user interface is centered around the gestural creation, modification and assembly of objects. The user mainly interacts with "gestural icons" and the virtual objects being created and assembled (Figure 9).

Gestural icons depict a certain hand pose, showing the user what gesture to perform to trigger a certain action or change a certain property. All hand icons translate to gestural input (e.g. draw outline), and all non-hand icons translate to automation (e.g. scan object). To perform an action, the user first selects the appropriate icon and then either performs the gesture (e.g. scales the object) or waits for the system to complete its task. The icons are context-sensitive, hence inform the user about the possible operations of the system.

Virtual objects displayed in MixFab can have three different states. Inert objects that cannot be modified without selecting a gestural icon, is colored in a slightly transparent gray (inert state). Once an object becomes modifiable using a gesture, it turns yellow (inactive state). Objects that are currently being modified, are colored green (active state). This color-coding provides feedback about the current system state, especially the grasping of objects. It allows users to determine whether the system recognizes them as engaged in a gesture and what influence their movement will have on the scene.

### Creating objects

There are four ways users can create objects: drawing an outline, having the system capture the outline of a physical artifact, 3D scan an existing object or load an existing 3D model.

#### *Draw outline*

Users can draw the outline of primitive shapes on the floor of MixFab, using only their index finger. The system then recognizes the sketch as either a circle or rectangle and extrudes it to 3D space. The height of the object is set by the height of the hand above the systems ground. Once the height is as desired, the other hand taps the floor to fix the height.

#### *Capture outline*

Capturing the 2D outline and extruding it to 3D space is a simple but fast method to capture an object's shape. Users can place existing objects anywhere in the frame and after a fixed dwell time, the system captures the outline and automatically extrudes it to the existing object's height. Users can manipulate the object's height, by indicating the height with one hand and confirming it with the other.

#### *3D scan*

The MixFab frame has a turntable built in which serves to rotate objects so that objects of more complex shapes can be scanned. To scan an object, the user selects the "scan object" icon, and places the physical object together with the scanning rig on the turntable (Figure 11). The system then waits until all hands are out of the frame before it starts rotating the object to capture it from all sides. Once scanning is complete, the virtual object appears where the existing one was placed.

#### *Load model*

CAD drawn models can incorporate functional aspects or higher-resolution details than what the built-in 3D scanner is

able to capture. When the user selects the "load model" icon, a grid of scaled-down models is provided from which the user can select the desired model (by selecting icons). The new object is then placed in the center of the frame.

### Manipulating objects

Once an object has been created, it can be manipulated in three ways: translation & rotation, scaling and cutting off parts of the object.

#### *Translation & Rotation*

Translation and rotation is performed using a one-handed grabbing gesture, much like one would grab a cup. If the object is grabbed so that the hand intersects the object, it attaches to the user's hand so that the object can be moved freely within the interaction volume. Grabbing any point away from the object lets users change the object's orientation. A lever is formed between the base of the object and the hand-tip, which is then used to rotate the object. Translation and rotation both snap to common values (e.g. the floor for translation and 0/90 degrees for rotation).

#### *Scaling*

Objects can be uniformly scaled using a two-handed compression gesture. Users place their hands on either side and the scaling factor is a function of their distance. We implemented relative, yet direct scaling using a fixed control-display gain. When users first assume the compression posture, their hand distance is identified as 100% scale. Changing the distance between both hands then scales the object.

#### *Cutting*

Cutting objects removes material, rather than splitting objects. To perform a cut, the user indicates the desired position of the cut using their flat hand ("Shuto"/Karate gesture) along the X-axis. Tapping on the ground with the other hand confirms the cut. If the user indicates the cutting position with the right hand, the right side of the cutting plane is discarded; indicating with the left hand removes the left side.

### Assembling objects

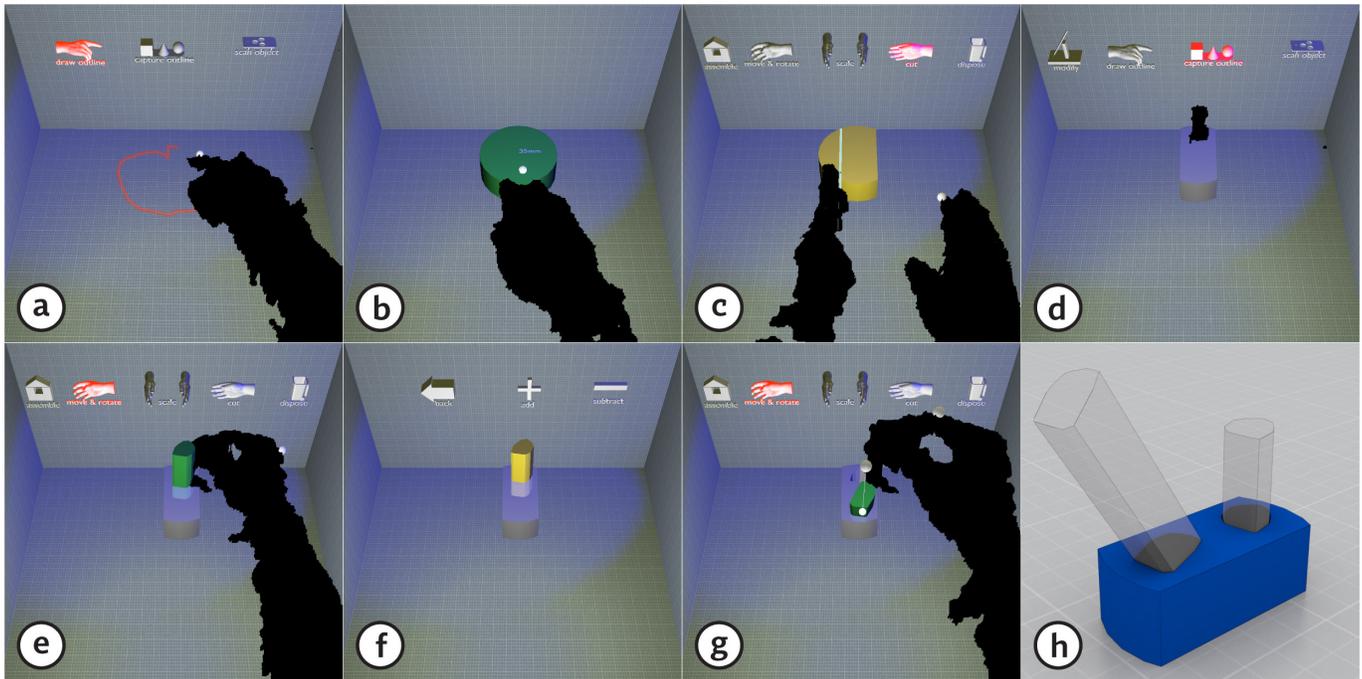
Object assembly combines two objects, either by adding them together or by subtracting one from the other. Fusing two objects can be used to add material or refine the shape of an object. Subtracting one object from the other is commonly used to create holes or cavities to hold other objects.

There is no specific gesture for assembly. Object assembly is simply a matter of selecting the way the two objects are to be combined. Union or difference of meshes are symbolized with a plus or a minus sign respectively (Figure 9).

## WALKTHROUGH: CONSTRUCTING A DESK ORGANIZER

We illustrate the systems use by constructing a desk organizer (Figure 9h) that will hold a pen and a glue-stick.

We start with creating the base shape by drawing a circular outline (Figure 9a). The system recognizes the drawing as a circle, beautifies it and offers the outline for extrusion. We set the height using one hand; the height snaps to 5mm increments and is displayed just above the object. To confirm the height, we tap with the other hand (Figure 9b).



**Figure 9.** MixFab’s user interface: (a) a user drawing an object’s outline, (b) setting the height of the cylinder, (c) plane cuts of the cylinder, (d) capturing the shape of a physical object (glue-stick) positioned in the virtual one, (e) moving an object (glue-stick) upwards by grabbing it with one hands, (f) object assembly (difference of glue-stick with existing virtual shape), (g) rotating an object (pen), (h) the desktop organizer (blue)

To create the semi-rectangular shape of the organizer, we cut off both sides. First we cut off the right side of the object by indicating the cut position with the right hand and confirming with the left. To cut the left side, we repeat the procedure, this time holding the left hand where we want to cut and confirming with the right one (Figure 9c).

Next, we create the first hole which will hold the glue-stick. We position the real glue-stick where we want the hole to be within the virtual object. Once in position, we select “capture outline” and move our hands out of the frame (Figure 9d). The system then captures the outline of the glue-stick and extrudes its height. Confirming that initial height with the left hand, turns it into a virtual glue-stick replica.

As the glue-stick was standing on the ground of the frame, the virtual glue-stick replica is on the ground as well. If we were to assemble the object as it is, we would create a hole through the whole shape. To have some material at the bottom of the hole, we grab the virtual glue-stick, move it a few millimeters up and release it to fix it in that position (Figure 9e).

Eventually we assemble the virtual glue-stick and the previously created base to create the hole for the stick. After selecting assembly, we are asked to choose the method of assembly (Figure 9f). Choosing subtract removes material where the glue-stick was, leaving a hole of correct size and position.

Lastly we repeat the steps above for the pen, placing it in its desired position, capturing its outline, extruding it and moving it up a few millimeters. To make the pen easier to access, we tilt it forward by grabbing at a point in space, forming a lever with which the object is re-oriented (Figure 9g). Once in

correct position and orientation, we assemble the virtual pen replica resulting in the final desktop organizer (Figure 9h).

## SYSTEM IMPLEMENTATION

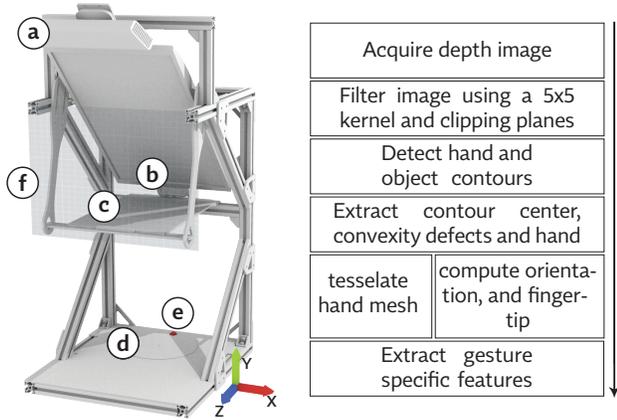
We built a prototype to implement the MixFab system by using *Holodesk’s* hardware frame [11]. Holodesk provides an immersive environment with an interaction volume roughly the size of modern 3D printers. Our hardware differs in that we use a Kinect for Windows rather than a Kinect360 and mirror setup, and have a turntable built into the frame for 3D scanning. Most importantly, on the software side, we employ a different processing pipeline and provide a gesture-based interface rather than a physics-based one.

### System Hardware

The hardware consists of a display mounted at a 45 degree angle, being reflected through a 50/50 half-mirror into the interaction space. A Microsoft Kinect depth sensor mounted at the top of the frame is used for capturing the interaction with the system, while a second camera placed between the display and half-mirror is used to implement perspective correction through face tracking. A motorized turntable for 3D scanning is built into the floor of the frame (Figure 10).

### Calibration

Two cameras need to be calibrated once (not per user): the facetracker and the Kinect. The facetracking camera is calibrated to a plane perpendicular to the half-mirror (Figure 10f). As the dimensions of the frame are known, the exact position and orientation of the facetracking camera (and thus faces tracked) can be mapped to real-world coordinates. The



**Figure 10.** [left] The MixFab hardware (a) Microsoft Kinect depth sensor, (b) webcam used for face tracking, (c) 50T/50R half-mirror, (d) motorized turntable, (e) common world origin (virtual), (f) face-tracker calibration plane (virtual). [right] MixFab’s general processing pipeline

Kinect is calibrated using its RGB camera by placing a laser-cut calibration pattern at a fixed location on the bottom of the frame. Using an approximation of the depth-to-RGB transformation, we can map depth data to the real world coordinate system with respect to a common origin (Figure 10e).

### General Processing and Gesture Recognition

MixFab follows an appearance based approach to hand posture and gesture recognition that requires no prior calibration or user augmentation. Depth data from the Kinect is processed to extract a set of features which is later used by specialized gesture recognizers. We rely solely on the depth image, as the half-mirror occludes the hands in the color image.

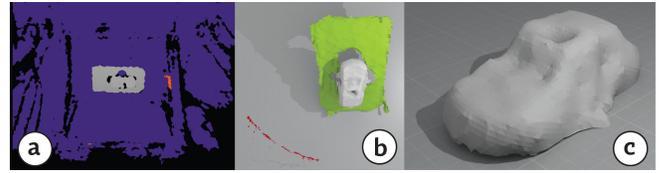
The general processing pipeline is as follows: first we acquire a depth frame from the Kinect, filter it using a 5x5 kernel, remove points using previously defined clipping planes and tessellate the remaining points to provide an occlusion mesh; all of which is implemented in OpenCL. Then we find all connected components touching the image border, with an area greater than  $A_{hand}$ . For each such contour, we compute its center, orientation via the Hu moments and finger-tip which is the convexity defect farthest along the principal axis.

#### Touching the floor

To implement touch input on the floor of the frame, we threshold the Y component of the finger-tip [28]. As the floor surface is flat and we map the Kinect depth-data to real-world coordinates with an origin in the floor plane, we set  $d_{surface} = d_{max} = 0$  and  $d_{min} = 20\text{mm}$ .

#### Sketch recognition

Sketches drawn on the floor are represented as 2D polygons  $p_1, \dots, p_N$  which we simplify using the Douglas-Peucker algorithm [8] and an edge-angle based filter (all adjacent edges with an enclosing angle less than  $\alpha_{join}$  are joined). We consider the sketch to be a rectangle if its average segment length  $\frac{1}{N} \sum_{i=1}^{N-1} |p_i - p_{i+1}|$  is less than  $L_{rect}$ . A circle has its center at the center of the polygon and its radius is the average distance of each point to that center. Rectangles are the bounding rectangle of the polygon.



**Figure 11.** The object scanning process. (a) The Kinect depth data (white) is filtered using clipping planes (purple), leaving some noise (red), (b) the scanned object (white) contains that noise (red). Removing noise and scanning table (green) and closing remaining holes results in the scanned object (c).

#### Open hands: plane cut and scaling

Open hands (Figure 5c) pointing along the Z-axis, result in a local minimum of the contour arc length. We exploit those observations and threshold the arc-length  $L$  of the hand contour going back  $d_{hand}$ mm. An “open hand” posture is recognized when the arc-length is less than  $L_{max}$ .

#### Grabbing

A not-grabbing hand forms several convexity defects [6]. If a defect with an angle greater than  $\alpha_{min}$  and depth greater than  $d_{open}$  is found, the hand is considered to be in an open state. If no such defect is found, the hand is considered to be in a grabbing state. While maintaining a grabbing pose, the finger-tip detection heuristic does not work reliably. Part of the grabbing pose is to bend the hand compared to the arm, causing the finger-tip to move away too far from the orientation axis previously computed. The real tip of the hand and origin of the hand contour form another convexity defect, which is approximately stable when the correct posture is maintained. Kalman filtering yields a usable hand-tip estimation.

#### Wiping

When performing the wiping gesture, users move their flat hand from one side of the frame to other in a speedy fashion. To detect that gesture, we continuously sample the contour centers  $X$  component with a fixed window size  $w$  (corresponding to the time in which the gesture has to be performed). If all points in that window are equidistant, their distances monotonously inc-/decreasing and the start/end points are at least  $d_{width}$  apart, a wipe gesture was performed.

#### Contour capturing

MixFab’s processing pipeline distinguishes between hand and object contours, if they are not connected (see previous section). All contours are subject to perspective distortion, which we correct using the previously acquired calibration. To capture the outline of an object, we build the convex hull of all object contours in the frame. Thus, objects can be grouped and produce a smooth shape from the noisy Kinect data, but we also slightly reduce precision. We further find the highest point within the hull, making that the initial extrusion height.

#### Object scanning

3D scanning in MixFab uses Kinect Fusion and a custom built turntable/scanning rig. Kinect Fusion estimates the camera to world coordinates using the *iterative closest point* (ICP) algorithm. ICP implicitly requires geometric features to converge, resulting in a poor scanning performance on “uninteresting” scenes. Normally Kinect Fusion is used on a room scale, with enough clutter so that the ICP based camera tracking works

well. In MixFab however, we scan single objects only, resulting in severe alignment errors without our scanning rig.

After filtering the depth image using clipping planes, there is still a some degree of noise left (Figure 11a). Due to the sparse nature of the images produced by the clipping, noise has a drastic impact on the camera tracking and thus scanning performance. To alleviate those issues, we designed a scanning table with an extreme width to height ratio, that aids Kinect Fusion in producing its camera alignment.

Once an object has been scanned, the resulting mesh data has to be cleaned. The noise integrated during scanning has to be removed, the scanning table has to be removed, holes have to be closed and the mesh has to be made watertight.

The scanning process begins with integrating plane-clipped point clouds into the *truncated signed distance function* (TSDF) representation maintained by Kinect Fusion. Once the object has been captured from all sides, the TSDF is transformed to a tri-mesh. All unconnected components with less than  $c_{noise}$  faces are removed. The scanning table surface is found by computing the largest connected component using a threshold of the discrete RMS curvature as connectivity condition. We then fit a plane to the scanning table vertices and rotate the mesh, so that the scanning table is in the  $XZ$  plane. All vertices and faces closer than  $d_{cut}$  to the scanning table plane are removed. We remove unconnected components with less than  $\frac{1}{2}c_{noise}$  faces, as well as non-manifold vertices/faces and fill gaps less than  $c_{hole}$  units in arc length.

The resulting mesh is likely to contain holes – at least one from cutting away the scanning table. We smooth the mesh (thus hole boundaries) using Laplacian smoothing. For each remaining hole, we fit a plane to the boundary vertices, project those vertices to that plane and compute a Constrained Delaunay triangulation (CDT), hence closing the hole. In a last step, we remove non-manifold vertices/edges created by the CDT and fill the resulting gaps (Figure 11c).

The mesh processing pipeline is implemented using the *Visualization and Computer Graphics Library*, *sgCore* and *qHull*. It takes 30 seconds to complete one revolution of the turntable and less than 10 seconds to perform the mesh processing.

## USER STUDY: SYSTEM EVALUATION

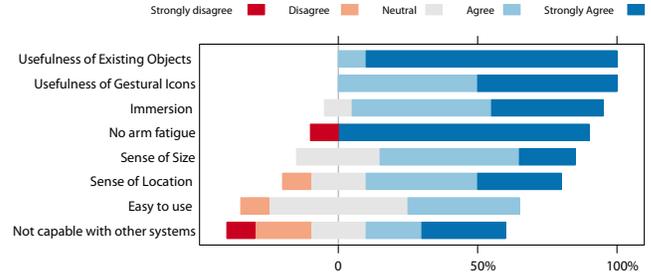
In this study, we evaluated the interaction cycle, design decisions and prototype implementation of MixFab. We were interested in how well non-engineering users could use the system. During the study, we collected primarily qualitative feedback to gain insight to the experience of using the system.

### Tasks and Procedure

Participants were first asked to sign a consent form and given an introduction to the system. We started by introducing the idea of designing objects for 3D printing by showing example objects created with MixFab (Figure 1c). They were then shown the “desktop organizer” walk-through (Figure 9) and given five minutes to familiarize themselves with the system.

#### Building things

To guide users during their exploration of the system, we asked them to replicate the desktop organizer example. The



**Figure 12.** Agreement distribution of the post-task questionnaire. The further bars extend to the right (relative to zero), the more users agree.

glue-stick and pen were provided and we guided users when necessary. Once the example was completed, we asked users to design a phone dock and provided our phone dock example (Figure 1c). Participants could use their own phone or an LG Nexus4 we provided. Users were encouraged to design the object on their own, but were assisted when necessary.

#### Semi-structured interview

Upon completion of all design tasks, we presented users with a set of statements and asked them to rate how much they agreed with each of them, on a 5-point Likert scale. We then went into a semi-structured interview asking about their experience, trying to gain insight into the usability of the system.

#### Participants

We invited 10 participants (5 female) from various departments on our campus. Participants were between 19 to 31 years old ( $M=24.3$  years,  $SD=4.52$ ), and all were right-handed. All except one participant had no experience with CAD systems or an engineering background (Rating scale, 1 a lot of experience to 5 no experience,  $M=4.56$ ,  $SD=1.01$ ).

## Results and Discussion

### Creation, Preparation, Assembly

MixFab’s construction mechanism was quickly understood by all users. Knowing when to create a new object, modifying it and assembling two objects seemed to pose no problem for the participants. User 2 reported that “when I was told to create the phone-dock I had a strategy in my head, thus knowing when to use a physical object.” The method of assembling objects to create new ones, was particularly well received – some users said that they “[...] very much liked this way of putting things together, to compose objects” (U6).

#### Using existing objects

Using existing objects during the design process was deemed useful by all users (100% agreement, Figure 12). Not having to measure objects and being able to place them in their desired position was highlighted by users “I very much liked [...] the thing that you can bring real physical objects in there.” (U6). Being able to use an existing object as starting point or base for designing new ones was mentioned as one of the benefits of the system: “I like the idea of being able to put my phone in there and design something around it.” (U8). The effortless integration of existing objects was even considered fun: “[...] it’s fun because you know there is no sort of effort required to replicate existing objects.” (U1)

### Natural object manipulation

For interaction to be natural users to feel immersed and have a sense of object size and location. 90% of the participants agreed that they were immersed into the system. A majority of users (70%) agreed that they had a sense of size and location of objects as well as their hands (U10: *“I liked that the objects were as big as they are in reality.”*). Users had no issues with selecting the gestural icons, further indicating that they had a sense of where things were in the frame. Manipulating objects was reported to be easy (U3: *“it felt easy to create and manipulate 3D objects compared to other systems which I image would take quite a bit of competency”*) and interacting with the system felt natural (U8: *“I liked how natural [...] the way I interacted with it [felt]”*).

### Usability

Several usability aspects are subsumed under “ease of use”: navigating within the system, ergonomic aspects and implementation specific artifacts. Finding their way around the system posed no greater challenge to users (U2: *“it [the system] is easy to comprehend; it’s self-explanatory.”*); partly because of the gestural icons. While interacting with the system, users at times asked what to do next but shortly afterwards selected the appropriate icon and continued on their own; all users reported that the gestural icons were useful (Figure 12).

Mid-air gestural interaction runs the risk of inducing arm-fatigue when used for an extended period of time. During our study, 9 of 10 users reported no arm fatigue. When using MixFab, many of the gestures are performed on the floor of the system and are often interleaved with short pauses of rest. As users sit close to the system, they do not have to extend their arms very far, further reducing the risk of arm fatigue.

The accuracy of the gesture recognition had the biggest impact on usability. Some users found it hard to execute precise movements (U5: *“sometimes I wished that it was more accurate”*, U6: *“the system is very sensitive, it was hard to really make accurate movements.”*) or had trouble with disengaging from a gesture. Most of the issues revolved around moving objects (U8: *“when you were moving and let go it was jumping a bit”*). Others however, found the precision to be sufficient. When asked if precision was a problem, user 7 answered: *“no, that was easy”*. Overall, users agreed that the system was easy to use (Figure 12).

### Using other systems

Our study participants had no experience with CAD and modeling tools. When asked if they would be able to design the items they designed during the study with other systems, 40% answered that they would be capable of doing so, despite no prior experience. Some users expected our system to be the way items are commonly designed: *“I have never used any of the CAD tools, but I think it’s kind of like this one”* (U7).

## DISCUSSION AND FUTURE WORK

The results of our study provide evidence that MixFab can be used by non-expert users to design meaningful objects for fabrication. Integrating existing objects was found particularly useful by all participants. Interaction with the system

is natural, by virtue of the gestures proposed during the user-defined gesture study and the mixed reality setup. Users have a sense of size and location of the object they’re designing.

In MixFab existing objects first have to be digitized before they can be used which is beneficial in that it allows us to e.g. scale and alter the object. Using the physical object as tangible proxy however, would likely increase immersion. We could introduce recursion by designing an object, fabricating it and introducing its physical manifestation back into the design process, making it semi-interactive fabrication.

Capturing existing objects comes at a detail vs cost trade-off. Our prototype can capture a crude form of objects in real-time, a more detailed one can be had at a small time cost. This trade-off is likely to shift towards an increased level of detail at decreasing costs. Other material properties, such as color and texture will likely be capturable in the near future. With recent advances in appearance fabrication and 3D printing such features could also be physically reproduced.

The mixed-reality environment of MixFab helps users to get a sense of size of the objects they’re designing, bringing both closer together. In the MixFabs prototype implementation objects look artificial however. Immersion could likely be increased by providing a more realistic object representation taking environmental lighting, proper material appearance and texture into account. Stereoscopy, in combination with the head-tracking, would further improve realism.

Not having to wear special equipment increases immersion thus naturalness of the interaction; not having to go through a calibration procedure prior to using the system increases the users readiness to engage with the system. Being free of user-augmentation and calibration comes at a cost, however: precision and accuracy. To some extent this is caused by the coarse spatial resolution of consumer depth cameras – something that is likely to change in the near future. A model-based hand tracking approach or specialized sensors hand-tracking sensors [13] are bound improve precision.

Gestural modelling is less precise than traditional CAD environments. First, the RGBd sensor limits attainable precision, compared to i.e., a mouse; something that will get better as such sensors improve. Second, gestures themselves can limit precision. It is hard, for example, to accurately place an object in mid-air without haptic feedback. MixFab currently implements snapping to the ground when moving objects, or snapping to 45 degree increments when rotating. Extending this approach to tool-specific constraints (as in Interactive construction [19]) will improve gestural modeling precision, and enable users to design symmetrical, reflected and parallel features - something that is not yet possible in MixFab.

## CONCLUSION

In this paper we have presented MixFab – a mixed-reality environment for personal fabrication. We described the system rationale, user-defined gesture study that informed its gestures and its implementation. The system was evaluated in a user study showing that users were successfully able to use the system – even considered it fun. We have presented a

set of objects designed with the system, demonstrating its expressive enough to create meaningful and useful objects. We found the effortless integration of existing objects and mixed-reality environment creates an engaging and immersive environment to create content for personal fabrication.

## ACKNOWLEDGMENTS

We would like to thank John Hardy for the helpful implementation tips and support. This work was supported by the EU Marie Curie Network iCareNet under grant number 264738.

## REFERENCES

1. Autodesk 123D Design.  
<http://www.123dapp.com/design>.
2. TinkerCAD: Mind to design in minutes.  
<https://tinkercad.com/>.
3. Anderson, D., Frankel, J. L., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., Sullivan, E., and Yedidia, J. S. Tangible interaction + graphical interpretation: a new approach to 3d modeling. In *Proc. SIGGRAPH* (2000), 393–402.
4. Arisandi, R., Takami, Y., Otsuki, M., Kimura, A., Shibata, F., and Tamura, H. Enjoying virtual handcrafting with tooldevice. In *Adjunct Proc. UIST* (2012), 17–18.
5. Benko, H., Jota, R., and Wilson, A. Miragetable: Freehand interaction on a projected augmented reality tabletop. In *Proc. CHI* (2012), 199–208.
6. Bradski, G., and Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly Media, Incorporated, 2008.
7. Buchmann, V., Violich, S., Billinghamurst, M., and Cockburn, A. Fingertips: Gesture based direct manipulation in augmented reality. In *Proc. GRAPHITE* (2004), 212–221.
8. Douglas, D. H., and Peucker, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10 (1973), 112–122.
9. Follmer, S., Carr, D., Lovell, E., and Ishii, H. Copycad: remixing physical objects with copy and paste from the real world. In *Adjunct Proc., UIST* (2010), 381–382.
10. Follmer, S., and Ishii, H. Kidcad: digitally remixing toys through tangible tools. In *Proc. CHI* (2012), 2401–2410.
11. Hilliges, O., Kim, D., Izadi, S., Weiss, M., and Wilson, A. Holodesk: direct 3d interactions with a situated see-through display. In *Proc. CHI* (2012), 2421–2430.
12. Holz, C., and Wilson, A. Data miming: inferring spatial object descriptions from human gesture. In *Proc. CHI* (2011), 811–820.
13. Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I., and Olivier, P. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proc. UIST* (2012), 167–176.
14. Kim, H., Albuquerque, G., Havemann, S., and Fellner, D. W. Tangible 3d: hand gesture interaction for immersive 3d modeling. In *Proc. EGVE* (2005), 191–199.
15. Lau, M., Hirose, M., Ohgawara, A., Mitani, J., and Igarashi, T. Situated modeling: a shape-stamping interface with tangible primitives. In *Proc. TEI* (2012), 275–282.
16. Lau, M., Saul, G., Mitani, J., and Igarashi, T. Modeling-in-context: user design of complementary objects with a single photo. In *Proc. SBIM* (2010), 17–24.
17. Llamas, I., Kim, B., Gargus, J., Rossignac, J., and Shaw, C. D. Twister: a space-warp operator for the two-handed editing of 3d shapes. In *ACM SIGGRAPH* (2003), 663–668.
18. Mori, Y., and Igarashi, T. Plushie: an interactive design system for plush toys. In *ACM SIGGRAPH* (2007).
19. Mueller, S., Lopes, P., and Baudisch, P. Interactive construction: interactive fabrication of functional mechanical devices. In *Proc. UIST* (2012), 599–606.
20. Nishino, H., Utsumiya, K., and Korida, K. 3d object modeling using spatial and pictographic gestures. In *Proc. VRST* (1998), 51–58.
21. Oka, K., Sato, Y., and Koike, H. Real-time fingertip tracking and gesture recognition. *Computer Graphics and Applications, IEEE* 22, 6 (2002), 64–71.
22. Pavlovic, V. I., Sharma, R., and Huang, T. S. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 7 (July 1997), 677–695.
23. Saul, G., Lau, M., Mitani, J., and Igarashi, T. Sketchchair: an all-in-one chair design system for end users. In *Proc. TEI* (2011), 73–80.
24. Schkolne, S., Pruett, M., and Schröder, P. Surface drawing: creating organic 3d shapes with the hand and tangible tools. In *Proc. CHI* (2001), 261–268.
25. Sheng, J., Balakrishnan, R., and Singh, K. An interface for virtual 3d sculpting via physical proxy. In *Proc. GRAPHITE* (2006), 213–220.
26. Starner, T., Leibe, B., Minnen, D., Westyn, T., Hurst, A., and Weeks, J. The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3d reconstruction for augmented desks. *Machine Vision and Applications* 14, 1 (2003), 59–71.
27. Weichel, C., Lau, M., and Gellersen, H. Enclosed: A component-centric interface for designing prototype enclosures. In *Proc. TEI* (2013), 215–218.
28. Wilson, A. D. Using a depth camera as a touch sensor. In *Proc. ITS* (2010), 69–72.
29. Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *Proc. CHI* (2009), 1083–1092.