

DualCAD: Integrating Augmented Reality with a Desktop GUI and Smartphone Interaction

Alexandre Millette*
École de Technologie Supérieure
Montreal, Canada

Michael J. McGuffin*
École de Technologie Supérieure
Montreal, Canada

ABSTRACT

Head-Mounted Displays (HMDs) combined with 3-or-more Degree-of-Freedom (DoF) input enable rapid manipulation of stereoscopic 3D content. However, such input is typically performed with hands in midair and therefore lacks precision and stability. Also, recent consumer-grade HMDs suffer from limited angular resolution and/or limited field-of-view as compared to a desktop monitor. We present the DualCAD system that implements two solutions to these problems. First, the user may freely switch at runtime between an augmented reality HMD mode, and a traditional desktop mode with precise 2D mouse input and an external desktop monitor. Second, while in the augmented reality HMD mode, the user holds a smartphone in their non-dominant hand that is tracked with 6 DoF, allowing it to be used as a complementary high-resolution display as well as an alternative input device for stylus or multitouch input. Two novel bimanual interaction techniques that leverage the properties of the smartphone are presented. We also report initial user feedback.

KEYWORDS

3D modeling; CAD; Augmented Reality; multimodal; real-world interface prop; HMD; smartphone

INDEX TERMS

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems---Artificial, augmented, and virtual realities; H.5.2 [Information Interfaces and Presentation]: User Interfaces---Input devices and strategies, Interaction styles; J.6 [Computer-Aided Engineering]: Computer-aided design

1 INTRODUCTION

The falling price of head-mounted displays (HMDs) creates new opportunities for virtual reality (VR) and augmented reality (AR). 3D environments will be accessible to more users, providing them with stereoscopic, head-coupled perspective depth cues. Such depth cues enable faster understanding of 3D scenes [52]. AR and VR platforms will also allow a user to point directly in 3D with a device or with their fingers, which is preferable to a mouse for certain tasks [1, 16, 25, 39, 47], such as simultaneous 6 DoF docking tasks [36]. However, this paradigm of HMD with 3D input also has problems. Recent consumer HMDs suffer from limited field-of-view and/or limited pixel density. Also, users quickly tire of holding their arms up [22], and pointing in midair is less precise and less stable when there is no physical support for the limbs involved. We propose two solutions to these problems for CAD and 3D modeling.

First, we allow a user to switch between an augmented reality mode (called **ARCAD**) and a desktop mode (**DesktopCAD**). In ARCAD, the user benefits from the depth cues provided by an HMD and also

from increased DoF input enabled by pointing in midair or with a hand-held device. In DesktopCAD, the user benefits from the precise, stable input of a mouse, and from a bright, high resolution external LCD monitor. The user can switch between rough, high-DoF tasks in ARCAD, and precise work in DesktopCAD. Our prototype comprising these two modes is called **DualCAD**.

Second, to improve the ARCAD mode, we leverage a smartphone enabling multitouch and stylus input. The smartphone's display is bright and high-resolution, complementing the relatively low resolution of the HMD. Also, input performed with multiple fingers or with the stylus on the phone benefit from the stability provided by the phone's physical surface, and also from the user's natural ability to use their hands asymmetrically (one to hold the smartphone, the other to interact with the phone's screen). Also, the phone's position and orientation are tracked in 3D, allowing it to be used as a direct or indirect input device, such as a menu, a wand, or a prop. Previous work has also used a handheld mobile device to interact with an immersive 3D environment [8, 13, 18, 39, 43, 50, 54], however our work extends this by introducing two new asymmetrical bimanual interaction techniques, Draw-and-Drop and Touch-and-Draw, that enable extrusion or painting before positioning geometry in the 3D scene.

We report user feedback from five professionals having years of experience in 3D content creation. Our literature survey also contributes a taxonomy for classifying previous work.

2 BACKGROUND

Because our prototype is multimodal, we surveyed CAD and similar systems involving a variety of interaction devices. Table 1 classifies such systems according to input and output. Although the table is not exhaustive of all previous work, whenever possible, each cell in Table 1 contains at least one example of previous work.

Consider first the options for input. Mice (top row of Table 1) have the advantages of filtering out hand tremor, retaining their position when released, and having buttons that move perpendicular to the plane of motion for increased stability during button events [2]. Because mice use *indirect* input (i.e., with separate input and output spaces), the cursor's motion can also be enhanced with acceleration or with target-aware techniques like semantic pointing [14]. A mouse is thus excellent for precise work when only 2 DoF are needed, and can even be faster than a 3D input device in simple 3D positioning tasks [5, 49]. However, users sometimes need to quickly sketch geometry in 3D, or reposition/rotate virtual objects with the aid of snapping, in which cases high precision is not required, and having a 3-or-higher DoF device is faster than a mouse [16, 25, 47] (lower rows of Table 1). Physical devices tracked in 3D can be used as props for easier repositioning of virtual objects of a similar shape [24]. These tradeoffs between mouse and 3D input motivate a system that can switch between the two.

Next, consider output devices. Several desktop systems support stereo output, often with some kind of 3D input. Some of these have the user's hands *in front* of the 3D scene (e.g., volumetric displays [19], HybridSpace [6]) and others require hands behind the 3D rendering (e.g., Toucheo [21], SpaceTop [34]). However, none of these examples allow hands to be placed inside the 3D scene with fully correct occlusions (i.e., virtual objects hiding hands *and vice versa*, as appropriate). A see-through HMD (right-most column of Table 1) has the advantage that, when combined with a depth camera, the captured depth image can be written into the Z-buffer to eliminate parts of the 3D scene that should be occluded by the hands. Our prototype uses the Meta 1 Developer Kit [37], which has a depth camera (however, limitations in the currently available API prevent writing the captured depth image into a Z-buffer).

Next, consider previous efforts toward multimodal 3D systems. Fuge et al. [16] describe a modeling interface that uses hands for "rapid generation and manipulation of ideas", stating "surface representations are [later] exported [...] to a commercial CAD system for further" operations. This suggests that a more integrated solution supporting two-way switching between modes would be valuable. Zollmann and Langlotz [57] propose a bridge between virtual 2D drawings and physical 3D clay models. HybridSpace [6] and a system combining HoloLens and Autodesk Maya [26] can both be described as multimodal systems, allowing a user to switch between one mode with mouse and 2D GUI, and another mode based on 3D input, however neither of these systems makes use of a hand-held device for input as in our work. Kijima and Ojika [30] also proposed a system with the ability to switch between Desktop

and HMD modes, however the virtual objects in their system could not be edited in the HMD mode.

Our DualCAD system can switch at runtime between different device configurations, and hence covers multiple cells in Table 1. One previous system that also covers multiple cells in Table 1 is Hyve-3D, with which our work differs by using an HMD with stereo display, allowing the user to place their hands *in* the 3D scene, and enabling runtime switching between modalities.

Much of Table 1 concerns handheld devices. Previous work has used smartphones as portals into a non-immersive 3D augmented reality [1, 27, 33], while other work has used handheld surfaces (tablets, smartphones, or PIPs (Personal Interaction Panels)) as alternative viewports, slices, or lenses within immersive or semi-immersive 3D environments [7, 8, 13, 44, 45, 50]. Other work has used 3D tracking of a handheld surface to select a cutting plane [43, 45] or used the 3D tracking of the device to perform geometric modeling [13, 39], assembly [28] and sketching of 2D geometry [13, 18]. Our system also uses tracking of the device for geometric modeling and enables sketching of 2D geometry, but ours is unique in allowing this 2D geometry to be extruded or projected onto a surface (via our Draw-and-Drop and Touch-and-Draw interaction techniques) before being positioned in the 3D scene.

DualCAD is distinguished from previous work primarily by covering multiple cells in Table 1, and also by its *simultaneous* use of a handheld touchscreen and HMD. The (Handheld touchscreen, Head-Mounted Display) cell in the table is also occupied by HVE Level Editor, but this latter did not allow for *simultaneous* use of the handheld device and HMD, unlike our system.

Table 1: Examples of systems covering various input/output combinations. Highlighted cells are occupied by the DualCAD prototype (in bold).

		OUTPUT			
		Desktop & tabletop monitor	Immersive screen (wall-sized or surrounding)	Handheld active display	Head-Mounted Display
INPUT	2D mouse, keyboard	(traditional CAD) HybridSpace [6] Kijima's Fused Environment [30] SpaceTop [34] DesktopCAD			HoloLens+Maya [26]
	2D multitouch on a desktop screen	Mockup builder [12]	N/A		
	Handheld PIP: Personal Interaction Panel	Tangible displays [44]	Studierstube [46]	Tangible displays [44]	Personal Interaction Panel [45] Studierstube [46] Construct3D [29] Virtual Gorilla Exhibit [7]
	Handheld device tracked in 3D	Hyve-3D [13] Place-casting [28] ARCAD	CAVE [11] Hyve-3D [13] CAVE VR Interface [39]	Tangible displays [44]	3DM [9] Personal Interaction Panel [45] Studierstube [46] Construct3D [29] Virtual Gorilla Exhibit [7] Kijima's Fused Environment [30] HVE Level Editor [50] ARCAD
	Handheld touchscreen (e.g. smartphone) for fingers & stylus	Hyve-3D [13] Plane-casting [28] ARCAD	Hyve-3D [13] CAVE VR Interface [39]	(various mobile apps) Hyve-3D [13] CAVE VR Interface [39] HVE Level Editor [50] ARCAD	HVE Level Editor [50] ARCAD
	Hand(s) tracked in 3D	HybridSpace [6] Mockup builder [12] MixFab [53] SpaceTop [34] 6D Hands [51]	CAVE [11]	Midair finger selection [27]	HoloLens [38] Virtual Gorilla Exhibit [7] (and many others) ARCAD

* alex_millette@hotmail.com
* michael.mcguiffin@etsmtl.ca

3 DESIGN OF DUALCAD

Motivated by [16, 30], we envision users alternating between phases of precise and rough work. Midair 3D input could be used to lay out some rough initial forms in 3D [16], that are then refined using a mouse. Or, a user may switch back to using 3D input to quickly and approximately rearrange the components of an assembly or redefine curves in 3D. DualCAD's two modes of DesktopCAD (for precise work) and ARCAD (for rough work, with optional snap-to-object) support such scenarios, allowing a user to switch, at runtime, between the upper-left cell in Table 1, and the cells in the lower-right.

3.1 DESKTOPCAD MODE FOR 2D INPUT

In the DesktopCAD mode, a traditional GUI allows the user to create, select, annotate with a text label, change the color of, group, ungroup, and transform virtual objects, using 2D and 3D widgets [10], as shown in Figure 1.

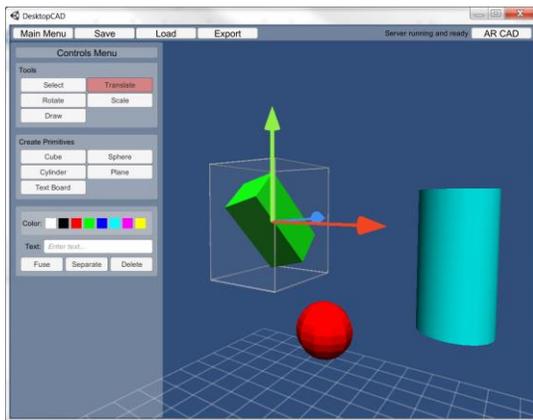


Figure 1: Traditional GUI interface in DesktopCAD

3.2 ARCAD MODE FOR 3D INPUT

We opted for a *see-through* HMD to allow the user to see their hands while they are working, as well as see the screen on the smartphone, and also to allow them to switch back to DesktopCAD and use the GUI on the external screen without removing the HMD.

In the ARCAD mode, the depth camera of the Meta HMD tracks finger tips in 3D, however the tracking does not capture fingers that are occluded when hands cross, and also does not capture finger orientation. Thus, we also wanted to use a physical hand-held device that could be tracked with 6 DoF for certain operations. In our prototype, this device is held in the non-dominant hand (NDH) while the dominant hand's (DH's) index finger is tracked in 3D. We chose a smartphone to serve as this NDH device, for a few reasons:

1. Most users already own a smartphone, and new sensors [17] allow phones to track their own position in 3D.
2. The bright, high resolution screen on the phone can display menus, options, or other feedback, complementing the HMD which typically has a more modest angular resolution and brightness.
3. Phones can detect multitouch gestures. So in addition to using the phone's position in space to point in 3D, the user can also use finger motions on the phone's screen to move in 3D [28] (e.g., a 2-finger pinch gesture for rotation or scaling) or perform gestures on the phone to launch commands.
4. Stylus input on the phone affords sketching and handwriting, and the phone's surface makes such input much more stable, precise, and comfortable for the user than with midair pointing [35]. The user can also feel the feedback of stylus contact with the screen.

3.2.1 Bimanual Interaction

In many two-handed tasks, the hands are used asymmetrically [20] with the NDH moving less frequently and more coarsely than the DH, and the NDH often providing the context or frame-of-reference for fine work done by the DH. Such asymmetrical hand use is leveraged in user interfaces like T3 [32], a 2D drawing program where the NDH positions a "toolglass" palette which the DH clicks through. The same T3 prototype also supports symmetrical bimanual interaction: the two hands can be moved apart to position diagonally opposite corners of a rectangle, or they can grab points on the canvas and then move at the same time to zoom or pan the workspace. Such symmetrical interaction is now widely familiar to users thanks to pinch gestures on smartphones.

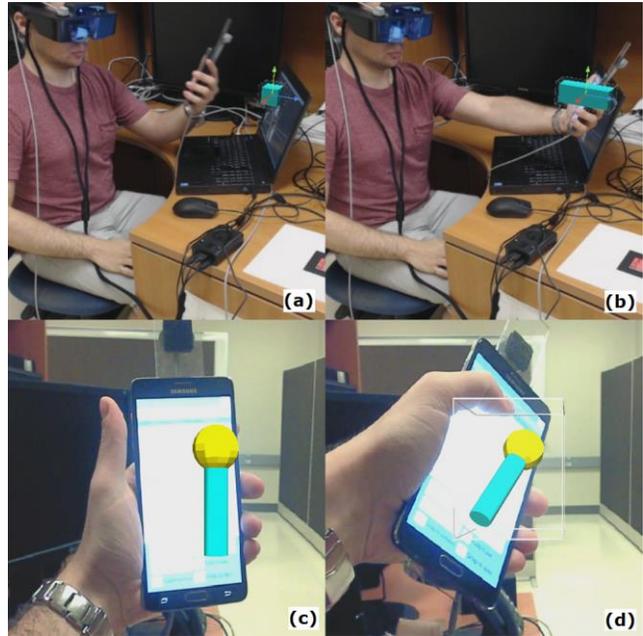


Figure 2: The user uses the scaling tool to rescale a cube (a) along a single axis (b) (virtual object composited from external perspective for illustrative purposes). The user selects a model with a direct transformation (c) and uses the smartphone to translate and rotate (6 DoF) the model (d).

In 3D user interfaces, there has been less investigation of bimanual interaction techniques, with the notable exceptions of [3], [40] and [56]. In our work, because the user's smartphone (held in the NDH) and the DH's finger are both tracked in 3D, we can implement symmetrical bimanual interaction techniques for controlling the camera (moving hands together or apart to zoom, or around each other to rotate the scene) or for object creation (moving the hands apart to position the diagonally opposite corners of a box). Furthermore, because the smartphone's screen enables multitouch and stylus input, we identified two novel techniques for asymmetrical interaction:

- **Draw-and-Drop:** the DH uses a stylus to draw a polygon on the phone (held in the NDH). This polygon is then extruded into 3D and remains attached to the phone. The NDH then positions the newly extruded object in the 3D scene and drops it there.
- **Touch-and-Draw:** the phone in the NDH serves as a prop for a virtual 3D object that has been grabbed. Dragging a finger across the phone's screen rotates the object, allowing the user to make any part of the object's surface face toward the front. The DH may also use a stylus to draw on the phone's screen, causing ink to be

projected onto the object's texture map. Again, the NDH may position the object in the scene and drop it there.

4 IMPLEMENTATION

Our HMD is a Meta 1 Developer Kit [37], with 23 degree FOV, 960×540 pixels (480×540 per eye), and front-facing depth and RGB cameras. A Samsung Galaxy Note 4 with a 5.7 inch screen was selected as the phone, because of its high quality "S Pen" stylus. An Android app on the phone uses Bluetooth to communicate with the main program, which runs on a laptop and uses Unity 3D to render the 3D scene. To track the position and orientation of the phone and of the HMD, we used Polhemus Patriot sensors. Early tests showed there was electromagnetic interference between the Polhemus and the phone, so we attached the phone and its Polhemus sensor to opposite ends of a long plastic rectangle, to separate their antennas. In future, optical tracking could replace the Polhemus.

5 INTERACTION TECHNIQUES

The smartphone displays a Marking Menu [31] that can be activated by drawing a stroke with the same hand that holds the phone (normally NDH). Users familiar with the stroke set can gesture without looking, for eyes-free operation. Items in the Marking Menu allow the user to create geometric primitives, change their color, add a handwritten text label with the stylus, or enter selection or transformation modes for translating, rotating, and scaling.

For selecting objects, empirically-based guidelines [41] recommend either raycasting, using a wand, or direct selection in 3D, depending on the circumstances. Our system supports all three. Raycast selection is done with the DH's index finger (from the point of view of the right eye), and wand selection is done by projecting a ray out from the phone, which can be comfortably held at the user's side.

To transform objects, we implemented both direct and indirect methods. With the *indirect* methods, the phone is used to control the transformation (either to translate, rotate, scale (Figure 2a, 2b), or to translate+rotate the selected object with 6 DoF), and the offset between the virtual object and the phone allows the phone to be held more comfortably near the user's body to reduce arm fatigue. In addition, indirect translation is done with a virtual wand, allowing small hand motions to translate the object a large distance.

With the *direct* (i.e., no offset) transformation methods, the phone is first positioned in 3D on the virtual object to select and a button on the phone's screen is pressed to select the object, then the phone is used to rotate, scale, or to perform 6 DoF translation+rotation (Figure 2c, 2d). Another button press releases the object. An additional method enables the DH finger to perform direct translation, which is less tiring than using the NDH phone. There is also an optional snap-to-object mode to facilitate object placement in ARCAD mode.

We tried implementing bimanual interaction techniques for camera manipulation and for object creation (e.g., simultaneously positioning opposite corners of a box), but found the HMD's FoV too narrow for these to be feasible. We therefore added a special mode to the phone, allowing the user to control the entire 3D scene with 1- or 2-finger "pinch-to-zoom/rotate" gestures to horizontally translate, rotate around a vertical axis, or scale the 3D scene.

The stylus can be used to draw an annotation displayed on the side of an object. It also serves in two novel interaction techniques. The first is **Draw-and-Drop**, a technique for sketching prismatic objects. Figure 3 shows a user adding a new building to a 3D scene with this technique. The user first draws a closed polygon on the phone (Figure 3a) causing a prism to be extruded (3b). Next, the user slides along the phone to adjust the extrusion height (3c),

positions and orients the phone in 3D to place the new object (3d), and presses a button on the phone to release the object (3e). This workflow allows the user to draw and extrude the object with arms held close to the body for comfort, and only position it in the scene at the end.

The second novel technique that uses the stylus is **Touch-and-Draw** (Figure 4). In this technique, the user grabs an existing piece of geometry with their NDH, causing a part of its surface to coincide with the phone, as if the phone were a physical proxy for the object. The user may *touch* the phone's screen with their finger to rotate the object, causing different parts of the object's surface to align with the phone's screen, as if the user were scrolling over the surface. The user may also *draw* on the phone with the stylus, causing the pen's ink trail to be projected onto the virtual object's texture map. In this manner, the user may draw on all sides of the object, while always "holding" the object in their NDH, respecting Guiard's model [20] of bimanual hand use. Once the user has finished drawing, they use their NDH to reposition the object and deposit it back in the 3D scene. This workflow also allows the user to draw and manipulate the object in a more comfortable position, with the arms held close to the body.

6 INFORMAL USER FEEDBACK

Five users (three professors and two students), not at our institution, performed simple tasks with the prototype. All five are members of an educational program in digital media and 3D content creation, each having 3 to 10+ years of experience with 3D modeling and animation software such as Maya, Mudbox, 3DS Max, and ZBrush. All five also had previous experience with Oculus VR headsets.

The users were asked to perform simple tasks in ARCAD mode for ≈25 minutes each: creating an object (like a cylinder), scaling it, then rotating and positioning it with 6 DoF manipulation with the phone, and repeating this 2 more times with other objects; using Draw-and-Drop to extrude and create 3 prisms; using Touch-and-Draw to draw on the textures of 3 cylinders.

All users were able to successfully complete the tasks, and all agreed that the ability to switch between a desktop mode and an AR mode was valuable. During the tasks, 4 out of 5 users described the interaction techniques as "cool" or "super" without prompting. When asked to compare with their previous experience with VR headsets, users disagreed on their preference between VR and AR: one user liked the way that VR cuts them off from the surrounding world to help them focus on their task, another found that AR is less disorienting than VR because they can still see the physical world and see their own nose.

3 users sometimes disliked having to switch their attention between the AR 3D scene and the visual feedback on the phone's screen, however this may have been due to the HMD's small FOV. As a solution, 2 users suggested showing visual feedback such as menus in the 3D space surrounding the user.

Users sometimes performed incorrect gestures in midair. To select an object, a user can point at it with a single finger extended in a vertical direction, making it visible to the Meta HMD's cameras. However, 3 of the users, when asked to select an object by pointing with their finger, extended their finger horizontally toward the object, making it difficult for the Meta to detect. Two users also tried grabbing an object with their bare (right) hand and tried rotating it, and one user even thought that the system was successfully interpreting his hand-rotation gestures when in fact it was misinterpreting the input.

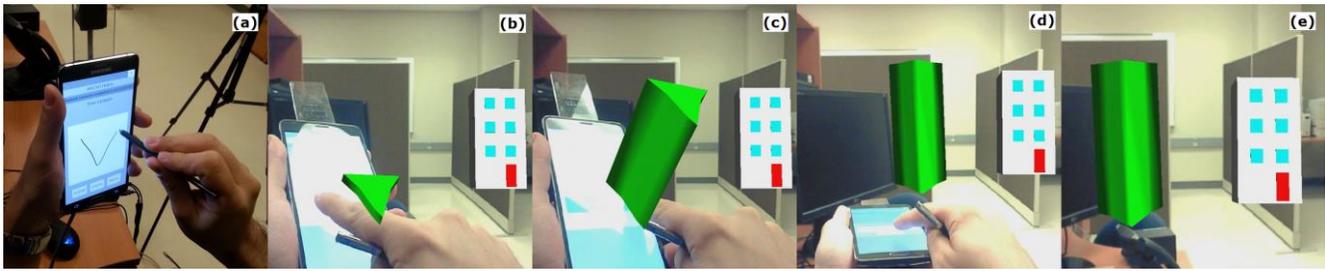


Figure 3: The Draw-and-Drop interaction technique. The user draws a polygon using the stylus (a), the polygon becomes a prism (b) which is then extruded by sliding a finger (c), moved into position on the scene (d) and finally detached (e).

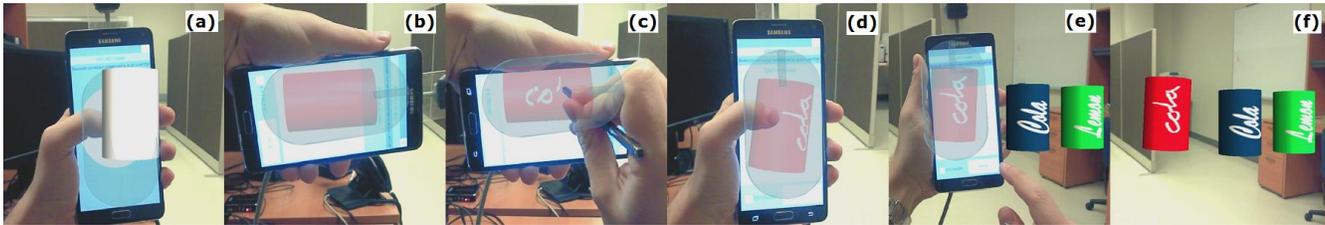


Figure 4: The Touch-and-Draw interaction technique. The user selects a model with the phone (a) which then snaps to the device (b). The user can draw with the pen (c), rotate the model with his finger (d), and finally place (e) and detach the model (f).

7 DISCUSSION

Having a larger FOV would improve the user experience, and would likely reduce the inconvenience expressed by some users when switching attention between the 3D scene and the phone's screen. Nevertheless, based on the user feedback, we anticipate that even with an immersive FOV $> 90^\circ$, users may still prefer to use the phone as an optional device, and not be forced to attend to the visual feedback on its screen. Our recommendations are to:

1. Display menus, status, and guides to gesture shortcuts, and other visual feedback on *both* the phone screen and in the 3D AR scene. This would, for example, allow users to perform radial menu gestures with their fingers on the phone screen without looking at the phone, even while they are still learning the gesture shortcuts.
2. Make the phone optional for as many functions as possible. Some users prefer using bare hands for certain functions. However, current limitations on hand-tracking algorithms, and the nature of bimanual techniques like Draw-and-Drop and Touch-and-Draw, mean that the phone will still be necessary for certain functions.
3. Recognize and respond to gestures that users find "intuitive": pointing directly at an object, grabbing an object with a single bare hand to rotate or move it, and moving hands apart or together to zoom. If, for technical reasons, these gestures cannot be reliably interpreted, at least detect their attempted use and guide the user to an alternative gesture using the phone.
4. Detect if the user appears "stuck" or is incorrectly performing a known gesture (such as extending a pointing finger with an orientation that is difficult to track), and guide the user toward correct usage of the gesture, and eventually toward expert usage.

REFERENCES

- [1] Bai, H., Lee, G.A., Ramakrishnan, M. and Billingham, M. 2014. 3D gesture interaction for handheld augmented reality. In *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*, ACM, 7.
- [2] Balakrishnan, R., Baudel, T., Kurtenbach, G. and Fitzmaurice, G. 1997. The Rockin'Mouse: integral 3D manipulation on a plane. In *Proc. CHI*, ACM, 311-318.

To implement recommendations 3 & 4, designers might take inspiration from previous systems in 2D [4, 15, 48] that guide users in learning gestures. These recommendations will also become easier to implement with advances in hand recognition algorithms [42].

By giving a choice to either use the phone or two bare hands, users would be free to vary their modality based on task, arm fatigue, and personal taste, with the phone enabling indirect input with arms close to the user's body and stable stylus input for drawing.

8 CONCLUSION

We have contributed (1) the DualCAD prototype implementation, in which a user performs 3D modeling by switching between a mouse-based monoscopic DesktopCAD mode that offers greater precision, and an augmented reality ARCAD mode with 3D input when more DoF are desired, and (2) the novel interaction techniques of Draw-and-Drop and Touch-and-Draw for use with a smartphone and HMD.

9 FUTURE DIRECTIONS

The HMD could also be used to display 3D virtual icons floating above physical mouse or keyboard buttons, to help a user learn keyboard shortcuts. Our system's hardware could be augmented with a tabletop display (e.g., combining our interaction techniques with those of Mockup Builder [12]), or with sensors to track the stylus in 3D or sense the user's grip of devices [23, 55].

- [3] Balakrishnan, R. and Kurtenbach, G. 1999. Exploring bimanual camera control and object manipulation in 3D graphics interfaces. In *Proc. CHI*, ACM, 56-62.
- [4] Bau, O. and Mackay, W.E. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proc. UIST*, ACM, 37-46.
- [5] Bérard, F., Ip, J., Benovoy, M., et al. 2009. Did "Minority Report" get it wrong? Superiority of the mouse over 3D input devices in a 3D placement task. In *INTERACT 2009*, Springer, 400-414.

- [6] Bogdan, N., Grossman, T. and Fitzmaurice, G. 2014. HybridSpace: Integrating 3D freehand input and stereo viewing into traditional desktop applications. In *Proc. 3DUI*, IEEE, 51-58.
- [7] Bowman, D.A., Wineman, J., Hodges, L.F. and Allison, D. 1998. Designing animal habitats within an immersive VE. *IEEE CG&A* (5), 9-13.
- [8] Budhiraja, R., Lee, G. and Billinghurst, M. 2013. Interaction techniques for HMD-HHD hybrid AR systems. In *ISMAR*, IEEE, 243-244.
- [9] Butterworth, J., Davidson, A., Hench, S. and Olano, M.T. 1992. 3DM: A three dimensional modeler using a head-mounted display. In *Proc. I3D*, ACM, 135-138.
- [10] Conner, B.D., Snibbe, S.S., Herndon, K.P., et al. 1992. Three-dimensional widgets. In *Proc. I3D*, ACM, 183-188.
- [11] Cruz-Neira, C., Sandin, D.J. and DeFanti, T.A. 1993. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proc. SIGGRAPH*, ACM, 135-142.
- [12] De Araújo, B.R., Casiez, G. and Jorge, J.A. 2012. Mockup builder: direct 3D modeling on and above the surface in a continuous interaction space. In *Proc. GI 2012*, Canadian Information Processing Society, 173-180.
- [13] Dorta, T., Kinayoglu, G. and Hoffmann, M. 2015. Hyve-3D and rethinking the 3D cursor: unfolding a natural interaction model for remote and local co-design in VR. In *SIGGRAPH 2015*.
- [14] Elmqvist, N. and Fekete, J.-D. 2008. Semantic pointing for object picking in complex 3D environments. In *GI 2008*, Canadian Information Processing Society, 243-250.
- [15] Freeman, D., Benko, H., Morris, M.R. and Wigdor, D. 2009. ShadowGuides: visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proc. ITS*, ACM, 165-172.
- [16] Fuge, M., Yumer, M.E., Orbay, G. and Kara, L.B. 2012. Conceptual design and modification of freeform surfaces using dual shape representations in augmented reality environments. *Computer-Aided Design*, 44 (10), 1020-1032.
- [17] Google. Project Tango. <https://www.google.com/atap/project-tango/>
- [18] 2014. Gravity - 3D Sketching. <https://vimeo.com/89649904>
- [19] Grossman, T., Wigdor, D. and Balakrishnan, R. 2004. Multi-finger gestural interaction with 3D volumetric displays. In *Proc. UIST*.
- [20] Guiard, Y. 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior*, 19 (4), 486-517.
- [21] Hachet, M., Bossavit, B., Cohé, A. and de la Rivière, J.-B. 2011. Toucheo: multitouch and stereo combined in a seamless workspace. In *Proc. UIST*, ACM, 587-592.
- [22] Hincapié-Ramos, J.D., Guo, X., Moghadasian, P. and Irani, P. 2014. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proc. CHI*, ACM, 1063-1072.
- [23] Hinckley, K., Pahud, M., Benko, H., et al. 2014. Sensing techniques for tablet+stylus interaction. In *Proc. UIST*, ACM, 605-614.
- [24] Hinckley, K., Pausch, R., Goble, J.C. and Kassell, N.F. 1994. Passive real-world interface props for neurosurgical visualization. In *Proc. CHI*, ACM, 452-458.
- [25] Hinckley, K., Tullio, J., Pausch, R., et al. 1997. Usability analysis of 3D rotation techniques. In *Proc. UIST*, ACM, 1-10.
- [26] Microsoft HoloLens. 2015. Microsoft HoloLens: Partner Demo with Maya by Autodesk. <https://www.youtube.com/watch?v=q0K3n0Gf8mA>
- [27] Hürst, W. and Van Wezel, C. 2013. Gesture-based interaction via finger tracking for mobile augmented reality. *Multimedia Tools and Applications*, 62 (1), 233-258.
- [28] Katzakis, N., Kiyokawa, K. and Takemura, H. 2013. Plane-casting: 3d cursor control with a smartphone. In *Proc. APCHI*, ACM, 199-200.
- [29] Kaufmann, H. and Schmalstieg, D. 2003. Mathematics and geometry education with collaborative augmented reality. *Computers & Graphics*, 27 (3), 339-345.
- [30] Kijima, R. and Ojika, T. 1997. Transition between virtual environment and workstation environment with projective head mounted display. In *Virtual Reality Annual International Symposium, 1997*.
- [31] Kurtenbach, G. and Buxton, W. 1994. User learning and performance with marking menus. In *Proc. CHI*, ACM, 258-264.
- [32] Kurtenbach, G., Fitzmaurice, G., Baudel, T. and Buxton, B. 1997. The design of a GUI paradigm based on tablets, two-hands, and transparency. In *Proc. CHI*, ACM, 35-42.
- [33] Lakatos, D., Blackshaw, M., Olwal, A., et al. 2014. T(ether): Spatially-aware handhelds, gestures and proprioception for multi-user 3D modeling and animation. In *Proc. SUI*, ACM, 90-93.
- [34] Lee, J., Olwal, A., Ishii, H. and Boulanger, C. 2013. SpaceTop: integrating 2D and spatial 3D interactions in a see-through desktop environment. In *Proc. CHI*, ACM, 189-192.
- [35] Lindeman, R.W., Sibert, J.L. and Hahn, J.K. 1999. Towards usable VR: an empirical study of user interfaces for immersive virtual environments. In *Proc. CHI*, ACM, 64-71.
- [36] McMahan, R.P., Gorton, D., Gresock, J., et al. 2006. Separating the effects of level of immersion and 3D interaction techniques. In *Proc. VRST*, ACM, 108-111.
- [37] Products: META Company. <https://www.getameta.com/products>
- [38] Microsoft HoloLens. <http://www.microsoft.com/microsoft-hololens/en-us>
- [39] Mine, M., Yoganandan, A. and Coffey, D. 2014. Making VR work: building a real-world immersive modeling application in the virtual world. In *Proc. SUI*, ACM, 80-89.
- [40] Piumsomboon, T., Altimira, D., Kim, H., et al. 2014. Grasp-Shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In *ISMAR*, IEEE, 73-82.
- [41] Poupirev, I., Ichikawa, T., Weghorst, S. and Billinghurst, M. 1998. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *CGF*, 41-52.
- [42] Sharp, T., Keskin, C., Robertson, D., et al. 2015. Accurate, Robust, and Flexible Real-time Hand Tracking. In *Proc. CHI*.
- [43] Song, P., Goh, W.B., Fu, C.-W., et al. 2011. WYSIWYF: exploring and annotating volume data with a tangible handheld device. In *Proc. CHI*, ACM, 1333-1342.
- [44] Spindler, M., Büschel, W., Winkler, C. and Dachsel, R. 2014. Tangible displays for the masses: spatial interaction with handheld displays by using consumer depth cameras. *Personal and Ubiquitous Computing*, 18 (5), 1213-1225.
- [45] Szalavári, Z. and Gervautz, M. 1997. The Personal Interaction Panel—a Two-Handed Interface for Augmented Reality. In *Computer graphics forum*, Wiley Online Library, C335-C346.
- [46] Szalavári, Z., Schmalstieg, D., Fuhrmann, A. and Gervautz, M. 1998. “Studierstube”: An environment for collaboration in augmented reality. *Virtual Reality*, 3 (1), 37-48.
- [47] Toma, M.I., Gîrbacia, F. and Antonya, C. 2012. A comparative evaluation of human interaction for design and assembly of 3D CAD models in desktop and immersive environments. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 6 (3), 179-193.
- [48] Vogel, D. and Balakrishnan, R. 2004. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proc. UIST*, ACM, 137-146.
- [49] Wang, G., McGuffin, M.J., Bérard, F. and Cooperstock, J.R. 2011. Pop-up depth views for improving 3D target acquisition. In *Proc. GI*, Canadian Human-Computer Communications Society, 41-48.
- [50] Wang, J. and Lindeman, R. 2014. Coordinated 3D interaction in tablet-and HMD-based hybrid virtual environments. In *Proc. SUI*, ACM, 70-79.
- [51] Wang, R., Paris, S. and Popović, J. 2011. 6D hands: markerless hand-tracking for computer aided design. In *Proc. UIST*, ACM, 549-558.
- [52] Ware, C. and Franck, G. 1996. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM TOG*, 15 (2).
- [53] Weichel, C., Lau, M., Kim, D., et al. 2014. MixFab: a mixed-reality environment for personal fabrication. In *Proc. CHI*, ACM, 3855-3864.
- [54] Wilkes, C.B., Tilden, D. and Bowman, D.A. 2012. 3D user interfaces using tracked multi-touch mobile devices.
- [55] Yoon, D., Hinckley, K., Benko, H., et al. 2015. Sensing Tablet Grasp+Micro-mobility for Active Reading. In *Proc. UIST*, 477-487.
- [56] Zeleznik, R.C., Forsberg, A.S. and Strauss, P.S. 1997. Two pointer input for 3D interaction. In *Proc. I3D*, ACM, 115-120.
- [57] Zollmann, S. and Langlotz, T. 2009. Poster: spatially augmented tape drawing. In *Proc. 3DUI*, IEEE, 147-148.