Novel Indirect Touch Input Techniques Applied to Finger-Forming 3D Models

Henri Palleis
University of Munich (LMU)
Munich, Germany
henri.palleis@ifi.lmu.de

Julie Wagner
University of Munich (LMU)
Munich, Germany
julie.wagner@ifi.lmu.de

Heinrich Hussmann
University of Munich (LMU)
Munich, Germany
hussmann@ifi.lmu.de

ABSTRACT

We address novel two-handed interaction techniques in dual display interactive workspaces combining direct and indirect touch input. In particular, we introduce the notion of a horizontal *tool space* with task-dependent graphical input areas. These input areas are designed as single purpose control elements for specific functions and allow users to manipulate objects displayed on a vertical screen using simple one- and two-finger touch gestures and both hands. For demonstrating this concept, we use 3D modeling tasks as a specific application area. Initial feedback of six expert users indicates that our techniques are easy to use and stimulate exploration rather than precise modeling. Further, we gathered qualitative feedback during a multi-session observational study with five novices who learned to use our tool and were interviewed several times. Preliminary results indicate that working with our setup is easy to learn and remember. Participants liked the partitioning character of the dual-surface setup and agreed on the benefiting quality of touch input, giving them a 'hands-on feeling'.

CCS Concepts

•Human-centered computing → Interaction techniques; Gestural input;

Keywords

indirect touch; polygon modeling; perspective-dependent gestures; two-handed input; qualitative data;

1. INTRODUCTION

Interactive workspace scenarios combining a vertical with a horizontal touch display have been proposed both in research and as commercial products (e.g., [5, 13], HP Sprout [1]). Apart from occlusion and precision, especially ergonomic considerations suggest the use of indirect touch input in such settings: the hands operate predominantly on the horizontal and the vision centers on the vertical surface (e.g., [22]). In contrast to established indirect touch input devices like touch pads or graphic tablets, the use of touch screens as input devices involves display capability and opens up novel design opportunities for touch interaction techniques.

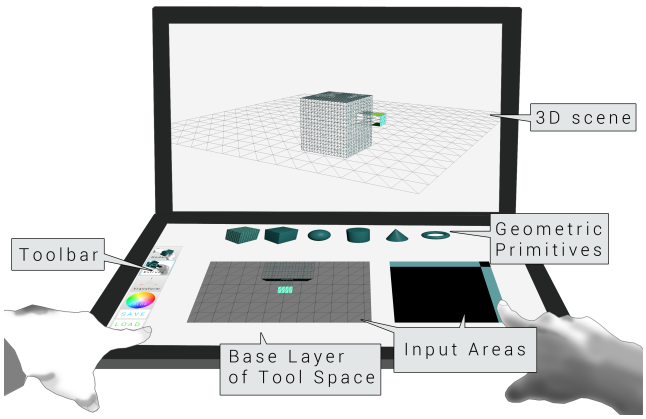

Figure 1: The basic setup consists of the visual display of the scene (top) and the tool space (bottom).

By employing multiple fingers and potentially both hands, touch input can lead to a higher control bandwidth compared to single pointer input. Existing touch pads such as Apple's Magic Trackpad already support *indirect* multi-finger touch gestures, which allow direct (e.g., two-finger scrolling) and high-bandwidth (e.g., two-finger zooming) control of functions depending on application context and system preferences. While currently the design opportunities of custom input techniques based on indirect multi-touch gestures are confined by the form factor and purpose of touch pads as *general* input devices, larger horizontal touch screens offer ample possibilities to explore form factors, arrangements, spatial structure, visualizations and mappings of touch input areas in order to serve *specific* functions. In this paper, we present our concept of a *tool space* - a horizontal touch display that allows the direct and spatial activation of commands through compositions of virtual touch pads with task-specific layout and functionality.

For this purpose we identified 3D modeling as an interesting use case, because of its functional complexity and the challenges arising from the dimensional mismatch of 2D touch input and virtual 3D environments. Three-dimensional modeling based on single pointer input decomposes the task complexity by introducing a variety of modes and virtual

[1] https://sprout.hp.com/us/en/

AVI '16, June 07 - 10, 2016, Bari, Italy

ISBN 978-1-4503-4131-8/16/06...$15.00

DOI: http://dx.doi.org/10.1145/2909132.2909257

tools, often resulting in a complex UI and lengthy interaction sequences, bearing little resemblance to hand crafting in the real world and presenting a high entry barrier for novice users. While the use of multi-touch input increases the input bandwidth, the transition from a 2D input surface to 3D content does not come natural and has inspired different approaches to navigation and object manipulation in virtual 3D environments (e.g., gesture sets [25], graphical handles [6] etc.).

In this paper, we present two indirect touch interaction techniques targeted at distinct 3D modeling tasks: *edge-loop scaling* and *extrusion*. While the techniques evolved around the scope of dual-display interactive workspaces and qualitative results show that users like this setup for its partitioning character, our techniques are not limited to them: they are designed to explore the potential of indirect touch input for complex tasks and may also be applied in slightly altered settings (e.g., on a large tilted display). Further, we acknowledge that our tools do not form a feature-complete 3D modeling suite and thus their expressiveness is limited to certain kinds of shapes. The contribution lies primarily in the conceptual work and the qualitative findings we gained in first user studies. However, our implementation also features novel technical concepts, like perspective-dependent gestures, which may have potential beyond the particular hardware setting.

2. RELATED WORK

In this section, we discuss related work on 3D touch interaction and indirect touch interaction. With our concept of the *tool space*, we try to address challenges of direct touch input for 3D manipulation tasks (a) without introducing a novel modeling paradigm and (b) by exploiting the design opportunities of a touchscreen used as indirect input device.

2.1 3D Touch Interaction

Despite precision and occlusion problems, direct multi-touch gesture input is in frequent use for basic 2D *RTS* object manipulation (rotate, scale, translate) [26] and view manipulation through pan-and-zoom navigation [12]. Several techniques in the literature address basic 3D manipulation tasks, such as positioning objects in 3D [20, 25], 3D object rotation [2, 11, 25], and 3D navigation tasks [10, 17]. Relatively little research has been conducted on multi-touch techniques for object-shape manipulation [16, 23] – to our knowledge none on *edge-loop manipulation* or *extrusion* for polygon modeling.

Essentially, all 3D touch interaction techniques face the problem of mismatching *degree of integration* [4] between the required 3D control of the object of interest and the 2D gestural input captured by the physical input device. There are two solutions to this problem: (1) by constraining the control-bandwidth of simultaneously modifiable dimensions to match the limited *degrees of freedom* (DoFs) of the 2D input surface or (2) by increasing the input-bandwidth, e.g. trough enabling simultaneous control of multiple parameters in the application with bimanual and gestural interaction.

The first approach often exploits the type of data and the context of the application to *constrain* the manipulation of an object '*in a meaningful way*' [17]: using *z-technique* [20], users can position furniture in a virtual 3D room constrained to the room's ground with a 1-finger gesture on the 2D input surface. Another example is the adaption of established 3D manipulation widgets from conventional graphical UIs to touch input, as presented by Cohé et al. [6] in their concept of an axis-constraining object transformation tool.

The second approach requires the design of larger gestural input vocabularies, e.g. through multi-digit or two-handed input [19, 23, 25]. In the *three finger rotation* technique [11], two fingers of one hand define an axis and the index finger of the second hand rotates the object around the defined axis. While such gesture sets allow an integral control of both positioning and orientation, there is evidence that a strict separation of control is preferred by users and outperforms the integral control of these parameters [21].

A completely different approach to designing new interaction techniques for 3D modeling is to change the modeling paradigm itself. Paper3D [23] adapts a paper craft paradigm using folding and cutting to create 3D models, taking advantage of real world analogies for gesture design. ILoveSketch [3] and Teddy [14] create 3D geometries from 2D strokes. Whereas this provides a quick *sense of achievement* for first-time users, it is less predictable how a 2D sketch is converted to a 3D shape. There seems to be no best paradigm: some modeling tasks are easier, others harder to perform using each paradigm.

2.2 Indirect Touch Input

Schmidt et al. [27] compared direct and indirect multi-touch input on a large horizontal/vertical dual surface setup. In particular, they explored an absolute mapping for pointing and dragging tasks. The indirect condition – hand contours where displayed on the vertical screen to provide visual feedback – caused fatigue and decreased pointing performance. Based on this, Voelker et al. [30] compared different interaction techniques to enable a comfortable tracking state for indirect touch input in such setups and found lift-and-tap to be the most promising. In the context of interactive workspace ergonomics (e.g.,[31]), both Voelker et al. [29] and Pfeuffer et al. [24] introduced gaze-based mode switching between direct and indirect touch input. Further, Gilliot et al. [8] explored the influence of input surface form factors on indirect target selection tasks with an absolute mapping and found that decreasing the input surface size improves target selection accuracy and that diverging aspect ratios between input and display areas decreases it.

More recently, indirect touch input has been explored as input modality for stereoscopic 3D systems. Simeone and Gellersen compared direct and indirect touch input techniques and found that indirect input results in less errors due to reduced occlusion [28]. Giesler et al. showed precision benefits for indirect touch input based on shadows cast by virtual objects onto a touch-screen surface compared to in-air interaction techniques [7].

3. DESIGN RATIONALE

To exemplify the concept of the *tool space*, we designed two interaction techniques for 3D polygon modeling. We acknowledge the wide variety of existing polygonal modeling techniques, but focus on the following: (1) *edge-loop scaling*, i.e. the scaling of selected edge loops and (2) *polygon extrusion*. Further, we do not intend to challenge existing input devices and user interfaces for 3D modeling, but hope to demonstrate the potential of our concept beyond the techniques proposed in this paper.

With our techniques, we address the sequential workflow

of 3D object-shape manipulation tasks in today's desktop environments. They are designed with three properties of interaction *instruments* in mind [4]: degree of *indirection*, *integration*, and *compatibility*. Degree of indirection refers to a measure of spatial and temporal offsets generated by an instrument describing a continuum between direct and indirect manipulation. An optimal, low degree of indirection could be achieved by, e.g. performing the modeling tasks directly on the scene object; however, due to fatigue effects, touch imprecision and occlusion issues [1], we chose to use indirect gestural interaction – which has been shown to be more precise [18] – and direct interaction on dedicated input areas and visual representations of the scene object in a separate *tool space*.

Degree of integration refers to the ratio between the number of DOFs that users can control simultaneously in the application and and number of DOFs captured by an input device. For integral tasks [15], e.g. 3D-positioning, we inherently have a degree of integration below one due to the mismatch between 2D surface input to control 3D position values; the ratio can be raised through the design of multi-digit or bimanual interaction techniques. The degree of compatibility refers to a measure for the similarity between the physical actions of users when performing the interaction technique and the response of the virtual object. We aim for a high degree of compatibility by using simple and well-known one- and two-finger touch gestures (i.e. one-finger tapping/dragging, two-finger rotation, pinch-to-zoom) in combination with a dynamic spatial multiplexing of the touch input area in contrast to complex gesture sets.

4. PROTOTYPE SETUP

Our prototype consists of two conventional touch screens, arranged as a pair of connected horizontal and vertical surfaces (see figure 1). We chose this setup as we foresee that touch displays will be integrated into augmented desktop environments. In particular, we conceive the horizontal touch display as an extension rather than a replacement for keyboard and mouse input, as proposed by [5]. Still, our techniques can also be transferred to PC/tablet combinations or collaborative settings with multiple tablets and a central large display.

The horizontal screen constantly displays a toolbar and the available geometric primitives (see figure 1) and leaves space required for our interaction techniques introduced below. Geometric primitives can be added to the scene using a swipe-up gesture, adhering to the conceptual model of both displays forming a unified interaction surface similar to curved displays as described in the literature [13]. The prototype is developed based on JavaFX 8 and currently runs on a HP desktop PC with a 2.8 GHz Intel Core i7 CPU, an ATI Radeon HD5670 graphics card and two Dell S2340T touch displays with a resolution of 1920 x 1080 pixels each. Implementation details of the setup as well as a documentation and source code of the algorithms underlying our techniques are available from the first author's website[2].

4.1 Scene Navigation and Object Positioning

Our prototype enables *panning*, *zooming*, and *orbiting* scene navigation: two-finger gestures are mapped to panning, *pinch* gestures to zooming, and one-finger touch gestures orbit the virtual scene camera on a sphere around the object continuously directed towards the object in the scene. These gestures can be performed anywhere on the visual base layer of the *tool space* and within the 3D scene.

Figure 2 outlines the basic object manipulation in our prototype – translating, rotating and scaling objects are supported as well. Users touch-select objects on the vertical screen, invoke an object manipulation widget, and transform objects using a pair of interactive orthographic views (front view, top view) and a dedicated rotation/scaling widget displayed within the *tool space*.

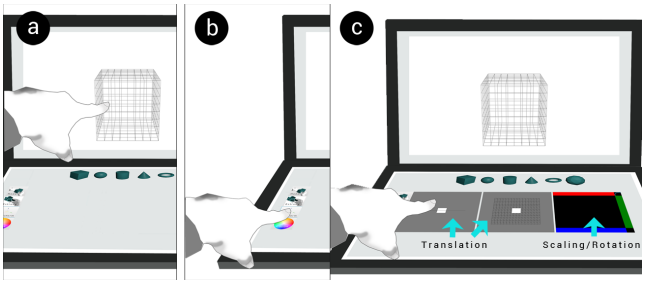

Figure 2: The basic workflow: (a) selecting an object with direct touch, (b) selecting a tool from the toolbar, and (c) using touch widgets to control the object indirectly.

4.2 Edge-Loop Scaling

With *edge-loop scaling*, users select a sequence of connected edges that reaches around the surface of the mesh, commonly referred to as *edge-loop*, and scale it (figure 3). By selecting one or multiple edge-loops and repeating this procedure, various shapes can be approximated from geometric primitives such as boxes or cylinders. In our prototype, we provided box and cylinder primitives with a fixed amount of edge loops, prepared beforehand using the mesh subdivision feature of Blender[3]. We designed a two-handed indirect touch interaction technique for scaling edge loops with the goal of increasing the degree of both *integration* and *compatibility* [4].

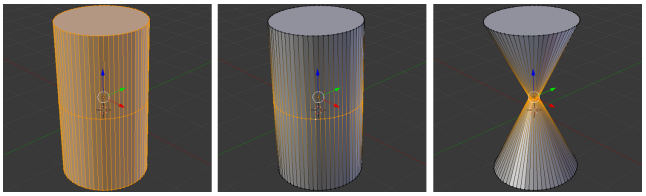

Figure 3: Edge loop scaling in Blender: a subdivided cylinder mesh in Blender (left), the selected edge loop (middle), the scaled edge loop and the resulting shape (right).

Our approach to edge loop scaling follows Guiard's [9] principles of asymmetric bimanual actions in the physical world. Figure 4 outlines the workflow: The non-dominant hand (1) sets the frame-of-reference, starts the interaction sequence and performs coarse actions by touching either the vertical or horizontal bar, which triggers the appearance of

[2]http://www.medien.ifi.lmu.de/indirect3D

[3]https://www.blender.org/

a mesh-selection volume that can be translated along the y-axis (vertical bar) or x-axis (horizontal bar) with one-finger and resized using a two-finger pinch gesture (2). At any time, all edge loops contained in the selection volume are selected. Performing a pinch gesture with the dominant hand (3) in the square area while maintaining the current selection with the non-dominant hand controls the scaling of the selected edge-loops. Refining the selection with the non-dominant hand and scaling with the dominant hand can occur in quick succession or simultaneously.

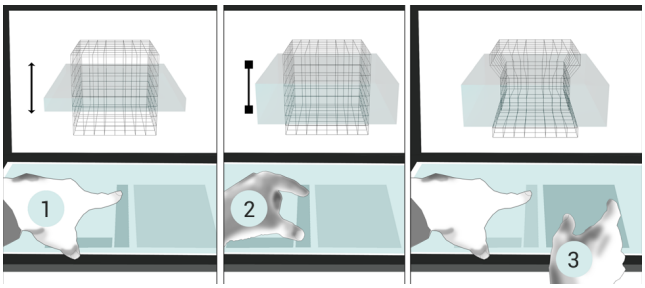

Figure 4: The bimanual edge loop scaling workflow: (1) the non-dominant hand translates and (2) scales the selection volumen, the dominant hand scales the contained edge loops (3).

Both, the dragging and pinch gestures used to translate and resize the selection volume are mapped relatively to the last state of the selection volume. Translating and resizing can be interrupted and resumed without causing the selection volume to jump. Further, position and size of the selection volume is constrained by the scene objects' dimensions allowing quick movements towards the respective ends. In order to minimize the need for clutching and yet enable precise control with small movements, the mapping between the finger movement on the bars and the translation and scaling of the selection volume is based on a discrete gain change function:

$$\triangle trans = \begin{cases} \triangle pos \div bar\text{-}length \cdot size & \text{if } \triangle p \leq 10 \\ \triangle pos \cdot 2 \div bar\text{-}length \cdot size & \text{if } 10 < \triangle p \leq 20 \\ \triangle pos \cdot 3 \div bar\text{-}length \cdot size & \text{if } \triangle p > 20 \end{cases}$$

$\triangle trans$ describes the translation delta of the selection volume along the respective axis of the 3D object over a time interval of 15 ms relative to the respective *size* of the 3D objects. $\triangle p$ describes the relative pixel distance traveled within the length p of the control bar (here $500px$). The angle of the pinch gestures of the dominant hand is not taken into account, in order to enable a comfortable hand positioning.

4.3 Extrusion

Performing extrusion in conventional 3D authoring tools (e.g., Blender) requires users to frequently switch modes resulting in a sequential workflow of alternating selection, navigation and transformation commands. We try to simplify this process by providing extrusion tools that allow the spatial activation of commands such as camera control, polygon selection and the actual extrusion. As seen in figure 5, our extrusion technique includes (a) the scene object, (b) the polygon selection tool, (c) the extrusion touchpad, and (d) the base layer for scene camera control.

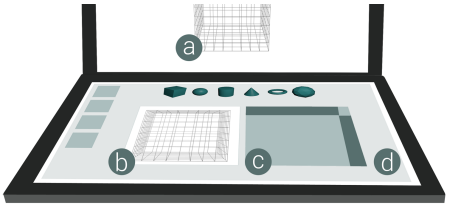

Figure 5: Overview of the extrusion tool: (a) the scene object, (b) the polygon selection tool, (c) the extrusion touchpad and (d) the base layer for scene camera control.

(1) Selecting Surfaces and Polygons. We decided to add the additional polygon selection tool in the horizontal tool space to reduce arm fatigue when making precise sub-mesh selections. Moreover, the two separated views of the 3D object help to keep an overview of the scene while selecting polygons. Users select a surface of the scene object on the vertical screen via direct touch, which causes the *polygon selection tool* (figure 5 (b)) to display an additional visual representation of the scene object. In particular, it shows an animated virtual camera motion towards the selected surface (point-of-interest, similar to Navidget [10]). The camera takes a viewpoint position perpendicular to the selected surface allowing the user to see onto the entire surface represented within the polygon selection tool area. The animated camera motion is intended to support the users' spatial awareness — note that the view of the scene object on the vertical screen is not changed. The distance between virtual camera and 3D object can further be adjusted by using pinch gestures within the polygon selection tool.

Inside the polygon selection tool, users can select and deselect single polygons of the object by tapping and also by dragging similar to finger painting. The selected polygons are visually highlighted both in the polygon selection tool and in the scene, displayed on the vertical screen. Figure 6 shows a schematic overview of these steps. The polygon selection is based on a standard ray casting algorithm that casts rays from the virtual camera of the polygon selection tool onto the selected surface. A swipe-up gesture performed on the polygon selection tool deselects all polygons.

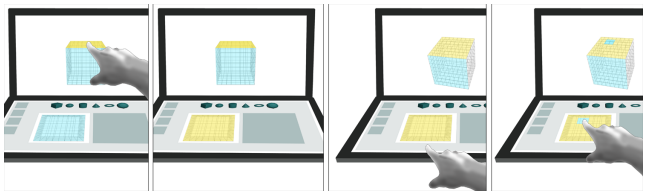

Figure 6: The workflow for selecting surfaces and polygons for extrusion. From left to right: a user selects a surface-of-interest, the polygon selection tool's view is updated, the user orbits the scene camera, the user selects single polygons.

(2) Perspective-dependent Extrusion. Once a set of polygons of an object is selected, the selection can be extruded via dragging gestures on the extrusion touch pad (figure 5

(c)) with the dominant hand. The interpretation of the gestures depends on the orientation of the scene controlled with the non-dominant hand on the base layer (figure 5 (d)).

During extrusion, changing the scene's camera viewpoint will cause continuously updated polygon selections, reassigning polygons to the possible extrusion directions *up*, *down*, *left*, and *right* at all times. Selecting a polygon and then rotating the camera will make the selection flip when a certain threshold angle is reached. Performing one-finger dragging gestures within the *extrusion pad* will cause the extrusion of the currently selected polygons. This allows to extrude along 3D paths without intermediary explicit de-/selections.

The extrusion workflow we propose is still inherently sequential, however it reduces activation costs [4] by spatial tool multiplexing: it encourages bimanual operation, during which the non-dominant hand sets the camera viewpoint using the tool space's base layer and therefore frames the scope of potential extrusions performed with the dominant hand using the extrusion pad.

When a touch gesture is initiated on the extrusion touch pad, the 2D touch coordinates are used to determine whether the touch occurred on one of the constraining bars or not. Further, a time stamp is set. The actual extrusion logic is handled by the event handler *On Touch Moved* that is executed on every touch movement update. The following pseudocode (Algorithm 1) illustrates the procedure that runs on every touch movement update.

Algorithm 1 The Pseudocode for the *OnTouchMoved* event handler of the extrusion pad

```
if timeDiff ≤ 100ms then
    sDeltaX += deltaX
    sDeltaY += deltaY
else
    if !extruded then
        determineSelectedFaces(sDeltaX, sDeltaY)
        extrude(selectedFaces)
        extruded = true
    else
        translateExtrudedFaces(deltaX, deltaY)
    end if
end if
```

The variable *timeDiff* is the difference between the current system time and the time stamp set at the *onTouchStart* event handler. As long as it is smaller or equal to 100 ms, the touch movement deltas are summed up in the variables *sDeltaX* and *sDeltaY*. After the first 100 ms of the touch movement, these sums of the two deltas are used in combination with the current camera viewpoint to determine which polygons to extrude. Then, the insertion of the new vertexes is triggered with the *extrude()*-function.

After the insertion, the current deltas of the touch movement are used to translate the new polygon. It can be constrained to the surface normal (the polygon's local z-axis) by starting the dragging movement in one of the constraining bars of the extrusion pad. In this case, a vector that is the result of a multiplication of the surface normal and a factor depending on the touch movement delta is added to each vertex. Whether the horizontal or the vertical touch movement delta is used also depends on the viewing orientation of the selected polygons. If the dragging is not initiated within one of the constraining bars, then the new vertexes are additionally translated by either the polygon's local x- or y-axis, also depending on the current viewing orientation of the polygon.

5. INITIAL EXPERT USER FEEDBACK

Six participants (two female, aged between 23 and 59, all right-handed) with various backgrounds and all with prior 3D modeling experience in either teaching or agency work performed three modeling tasks: (1) modeling a simple rectangular table with a single centered leg (i.e. the *edge-loop scaling task*) and (2) re-modeling a given shape available as print-out during the task (i.e. *extrusion task*) and (3) modeling a small scene with a combination of both techniques. After task (1) and (2), they rated *ease-of-use*, *perceived performance time* and *precision* using Likert-scales (1 = "strongly disagree", 5 = "strongly agree"). Finally, we conducted a semi-structured interview.

In general, the participants experienced the control of our tools as easy ((median (md) = 5 for task 1, = 4 for task 2)) and fast (md = 5 for task 1, = 4.5 for task 2). Regarding precision, the users' ratings are less positive (md = 2 for task 1, = 3 for task 2), which is not surprising due to finger input and the tool's lack of precision features (e.g., numerical information, snapping etc.). Three participants stated that modeling using our bimanual edge-loop scaling tool almost felt like working with *'modeling clay'* or *'pottery'* and after the *extrusion task*, one participant said that extruding felt like *'painting in space'*, indicating a high degree of compatibility [4]. The more experienced modelers did not see an instant benefit of the tools, but were still surprised about their own capabilities in creating shapes with our prototype. P3 commented positively on the conveyed *sense of achievement*: shapes created with the edge-loop scaling tool exhibit similar stylistic properties, yet resulted in 'beautiful shapes', which might stimulate exploration of various shapes.

6. EVALUATION WITH NOVICE USERS

Following the implications of the initial user feedback that our techniques rather stimulate exploration than allow for precise modeling, we conducted an observational user study with five 3D modeling novices (four female), aged between 17 and 25, all students from varying educational backgrounds (business, sociology, cultural science and computer science). Our goal was to gain qualitative data on the application's general learnability and usability.

6.1 Procedure

The study was organized in 3 sessions (60-90 minutes) per participant, with breaks of 2-5 days between sessions. The goal of the first two sessions was to familiarize the participants with the functions of the prototype and give them first hands-on experience. In the first session, they got an introduction to the prototype's features and had the possibility to explore them freely. The second session started with a recap of the prototype's functions, followed by a free modeling task. In the last session, the users were asked to apply their knowledge from the prior sessions and model an imaginary room with FAD (figure 8 and 7).

In the end of each session, participants were asked to answer 5-Point Likert scale questions concerning the learnability of the system and to classify the prototype on a 7-point

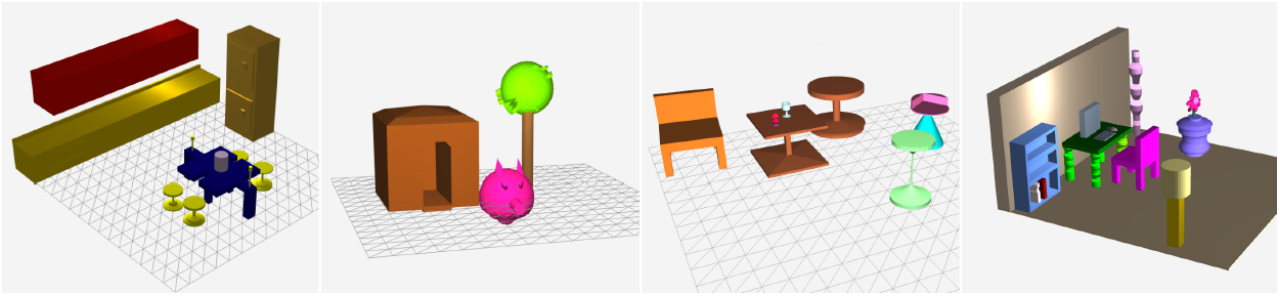

Figure 7: Exemplary modeling results from our observational study with five 3D modeling novices.

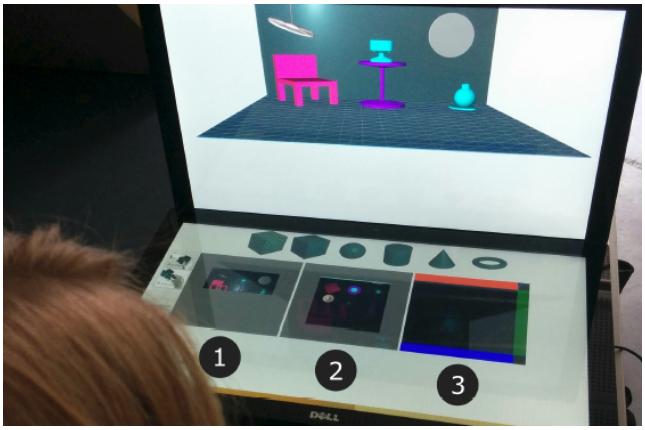

Figure 8: A participant during one of the sessions. (1) Side view to translate selected objects along the scene's *y-axis*, (2) top view to translate them in the scene's *xz-plane* and (3) scaling widget with axis-constraining bars (red, green, blue) at the borders.

semantic differential. The answers were further discussed in concluding semi-structured interviews. After the final session, participants reflected on the three sessions and their progress in a more comprehensive interview.

6.2 Observations and Feedback

All participants perceived the interaction with the prototype as easy across all sessions (Table 1). After familiarizing with the tool it became '*quite intuitively usable*' (P1) even if it seemed rather complex when it was first introduced by the experimenter. The early doubts some users had regarding the learnability of the tool are reflected in the semantic differential ratings: after the first session, three of five users tended towards '*hard to learn*', whereas after the second session four users rated on the '*easy to learn*' side (Table 1). The interviews of the second sessions further revealed that some were surprised (P1,P3,P4) by how well they remembered how to work with the tools from the first session (md = 2 for the statement '*It requires a lot of time to learn how to use the prototype*' across all participants and sessions).

6.2.1 Reproducibility

According to the questionnaires, our techniques facilitate the reproducibility of the workflow: The ratings of the two statements (a) '*The prototype requires me to remember a lot of details*' (md = 2 across all participants and sessions) and (b) '*The prototype is designed in such a way that learned steps are easily remembered*' (md = 4 across all participants and sessions) support this assumption.

6.2.2 Display Setup

All participants commented positively on the display setup because of its partitioning character. The vertical screen was mostly used as overview and the horizontal screen as the main input area. Interestingly, all participants except P4 used the vertical screen for navigating the scene, although the same could have been achieved using the horizontal screen. When we addressed this, the most common response was that the vertical screen was used for having an overview over the entire scene, hence it felt natural to navigate the whole scene on there. However, three of the participants also mentioned that in the current version of the tool a non-touch screen would have been sufficient for an overview. Further, some wished for more functionality on the vertical screen, such as direct manipulation of objects.

6.2.3 Role of Hands and Touch

Mostly, all participants operated the tool exclusively with their dominant hand, no matter which display they used. Only when required by the tool, two-handed input was used (edge-loop scaling). Approached upon this, users said that as long as the tool does not require two-handed input, controlling it with the strong hand feels quicker and more accurate, even though some noticed that distances between input elements would be shorter when handled with their weak hand. P3 commented that the tool could be designed differently with a clear segregation of tasks for the strong hand and for the weak hand. P5 mentioned that the edge-loop scaling tool layout (selection volume control on the left, scaling on the right) can lead to the necessity of either switching or crossing hands, because '*you would start selecting the area with your right hand in the left field and then realize you need to operate the field on the right as well.*'

The participants agreed on the benefiting quality of using touch interaction instead of mouse and keyboard input for the tool, referring to the '*hands-on feeling*' and comparing it to working with clay. However, selecting polygons for extrusion led to occasional fat finger problems. P3 suggested introducing brushes with diameters for selection and a controllable polygon selection view (manual zoom and navigation). Prompted by these issues, participants felt that the tool is generally not suitable when details are important. Accurate placing or scaling is not possible without some sort of numerical input or constraints (e.g., only allowing gradual scaling or a grid-alignment) for distances or positions.

6.2.4 Problems

Even though perceived as easy-to-use, some felt that the insert-mechanism for objects was sumptuous: '*swiping it in is a nice metaphor, but in the computer context I still have clicking in mind*' (P2). We could also observe this during the study when participants would tap on the object several times to insert it into the scene and then after a few tries remembering the swipe-move. It also occurred that participants would swipe towards the middle of the horizontal screen, which is not recognized by the system.

Finally, when asked for suitable usage contexts, participants recommended it for schools due to it's capability to facilitate visual thinking. Some saw the direct interaction as a good way to teach 3D modeling to non technically-minded users. Further, they saw potential in the setup for purpose-built games or creative tasks such as video or photo editing.

7. DISCUSSION AND FUTURE WORK

Both our initial feedback session with expert users and the observational study with novice users were formative studies with few participants, during which we only gathered qualitative data on our proposed setup and interaction techniques. Thus, we cannot provide conclusive results that quantify characteristics of our interaction techniques compared to state-of-the-art techniques and allow a generalization to a wider context. Nonetheless, our results provide a number of interesting perspectives on novel indirect touch interaction techniques.

We see potential for our concept beyond 3D modeling, especially in application domains where existing user interfaces require multi-dimensional input and exhibit a low *degree of compatibility* [4], i.e. a divergence between the physical actions of the user and the response of the manipulated object. That may for instance be the case with creative applications, such as timeline animation or sound arrangement. In such cases, task-specific spatial activation areas in a separate tool space can visually guide high-bandwidth and potentially two-handed input that is easy to learn and remember and can deliver a quick sense of achievement.

The concept of spatial tool activation within *tool space* also encourages to think about novel high-degree-of-freedom interaction techniques, resulting from combinations of commands. E.g., a combination of extrusion and camera rotation might be used to easily create bent or twisted shapes.

Further, our observations suggest that, in the particular case of 3D modeling, our techniques may present rather an extension to than a replacement of the existing ecology of input devices and user interfaces. We see potential especially in early modeling phases characterized by form finding and exploration. We envision a scenario, where 3D artifacts are first approximated using touch input techniques like ours and subsequently elaborated with established tools that allow for precision. In order to gain a better understanding of such a scenario, we plan an in situ evaluation of our prototype with high school students who currently are being trained in Google SketchUp [4]. Further, we intend to run formal experiments to learn about performance differences between our indirect touch tools and established input techniques, e.g. by comparing our extrusion tool to input based on mouse and keyboard shortcuts.

[4]https://www.sketchup.com

8. CONCLUSION

In the context of indirect touch input techniques, we presented our concept of a *tool space* - a dedicated partition of a touch screen separated from the main visual display area of a computer setup used to render task-specific virtual touch pads with specialized input mappings. To exemplify this concept, we implemented two indirect touch techniques for 3D polygon modeling based on multi-finger and two-handed gestural input on a horizontal/vertical dual display setup. First user feedback indicates that our techniques are enjoyable, easy to use and stimulate exploration. Participants of a multi-session observational study liked the partitioning character of the dual-display setup and could learn as well as remember the tools more easily than they expected in the beginning. Some reported to experience edge-loop scaling as *modeling clay*, which suggests that a 'hands-on feeling' can also be achieved with indirect touch input techniques. However, participants expressed criticism regarding the control precision. Also, two-handed input was only used when enforced, indicating that the adoption of bimanual operation may need longer learning phases.

9. REFERENCES

[1] P.-A. Albinsson and S. Zhai. High precision touch screen interaction. In *Proc. CHI '03*, pages 105–112. ACM, 2003.

[2] O. K.-C. Au, C.-L. Tai, and H. Fu. Multitouch gestures for constrained transformation of 3d objects. *Comp. Graph. Forum*, 31:651–660, May 2012.

[3] S.-H. Bae, R. Balakrishnan, and K. Singh. Ilovesketch: As-natural-as-possible sketching system for creating 3d curve models. In *Proc. UIST '08*, pages 151–160. ACM, 2008.

[4] M. Beaudouin-Lafon. Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proc. CHI '00*, pages 446–453. ACM, 2000.

[5] X. Bi, T. Grossman, J. Matejka, and G. Fitzmaurice. Magic desk: Bringing multi-touch surfaces into desktop work. In *Proc. CHI '11*, pages 2511–2520. ACM, 2011.

[6] A. Cohé, F. Dècle, and M. Hachet. tbox: A 3d transformation widget designed for touch-screens. In *Proc. CHI '11*, pages 3005–3008. ACM, 2011.

[7] A. Giesler, D. Valkov, and K. Hinrichs. Void shadows: Multi-touch interaction with stereoscopic objects on the tabletop. In *Proc. SUI '14*, pages 104–112. ACM, 2014.

[8] J. Gilliot, G. Casiez, and N. Roussel. Impact of form factors and input conditions on absolute indirect-touch pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 723–732, New York, NY, USA, 2014. ACM.

[9] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior*, 19(4):486–517, 1987.

[10] M. Hachet, F. Decle, S. Knödel, and P. Guitton. Navidget for 3d interaction: Camera positioning and further uses. *Int. J. Hum.-Comput. Stud.*, 67(3):225–236, Mar. 2009.

		Score								
		3	2	1	0	1	2	3		
incomprehensible	Session 1	-	-	-	-	3	2	-	Session 1	*comprehensible*
	Session 2	-	-	-	-	3	-	2	Session 2	
	Session 3	-	-	-	-	1	3	1	Session 3	
easy to learn	Session 1	1	1	-	-	1	2	-	Session 1	*hard to learn*
	Session 2	2	1	1	-	-	1	-	Session 2	
	Session 3	1	3	-	-	-	1	-	Session 3	
unpredicatble	Session 1	-	1	-	1	2	1	-	Session 1	*predictable*
	Session 2	-	-	1	2	-	2	-	Session 2	
	Session 3	-	-	-	1	3	1	-	Session 3	
quick	Session 1	1	-	-	1	2	2	-	Session 1	*slow*
	Session 2	1	2	1	1	-	-	-	Session 2	
	Session 3	-	3	-	1	-	1	-	Session 3	
complicated	Session 1	-	-	-	-	3	1	1	Session 1	*easy*
	Session 2	-	-	-	-	4	-	1	Session 2	
	Session 3	-	-	-	-	3	1	1	Session 3	
clear	Session 1	1	1	3	-	-	-	-	Session 1	*confusing*
	Session 2	2	1	1	1	-	-	-	Session 2	
	Session 3	1	3	1	-	-	-	-	Session 3	

Table 1: Distribution of ratings for the semantic differential. The numeric values represent the number of participants.

[11] M. Hancock, S. Carpendale, and A. Cockburn. Shallow-depth 3d interaction: Design and evaluation of one-, two- and three-touch techniques. In *Proc. CHI '07*, pages 1147–1156. ACM, 2007.

[12] M. S. Hancock, S. Carpendale, F. D. Vernier, D. Wigdor, and C. Shen. Rotation and translation mechanisms for tabletop interaction. In *Proc. Tabletop '06*, pages 79–88. IEEE Computer Society, 2006.

[13] F. Hennecke, W. Matzke, and A. Butz. How screen transitions influence touch and pointer interaction across angled display arrangements. In *Proc. CHI '12*, pages 209–212. ACM, 2012.

[14] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *ACM SIGGRAPH 2007 courses*, page 21. ACM, 2007.

[15] R. J. Jacob, L. E. Sibert, D. C. McFarlane, and M. P. Mullen Jr. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(1):3–26, 1994.

[16] Y. Kang, H. Kim, H. Suzuki, and S. Han. Editing 3d models on smart devices. *Computer-Aided Design*, 59(0):229 – 238, 2015.

[17] T. Klein, F. Guéniat, L. Pastur, F. Vernier, and T. Isenberg. A design study of direct-touch interaction for exploratory 3d scientific visualization. *Comp. Graph. Forum*, 31(3pt3):1225–1234, June 2012.

[18] S. Knoedel and M. Hachet. Multi-touch rst in 2d and 3d spaces: Studying the impact of directness on user performance. In *Proc. 3DUI '11*, pages 75–78. IEEE Computer Society, 2011.

[19] J. Liu, O. K.-C. Au, H. Fu, and C.-L. Tai. Two-finger gestures for 6dof manipulation of 3d objects. *Comput. Graph. Forum*, 31(7pt1):2047–2055, Sept. 2012.

[20] A. Martinet, G. Casiez, and L. Grisoni. The design and evaluation of 3d positioning techniques for multi-touch displays. In *Proc. 3DUI '10*, pages 115–118, March 2010.

[21] A. Martinet, G. Casiez, and L. Grisoni. The effect of dof separation in 3d manipulation tasks with multi-touch displays. In *Proc. VRST '10*, pages 111–118. ACM, 2010.

[22] S. Meyer, O. Cohen, and E. Nilsen. Device comparisons for goal-directed drawing tasks. In *Proc. CHI '94*, pages 251–252. ACM, 1994.

[23] P. Paczkowski, J. Dorsey, H. Rushmeier, and M. H. Kim. Paper3d: Bringing casual 3d modeling to a multi-touch interface. In *Proc. UIST '14*, pages 23–32. ACM, 2014.

[24] K. Pfeuffer, J. Alexander, M. K. Chong, Y. Zhang, and H. Gellersen. Gaze-shifting: Direct-indirect input with pen and touch modulated by gaze. In *Proc. UIST '15*, pages 373–383. ACM, 2015.

[25] J. L. Reisman, P. L. Davidson, and J. Y. Han. A screen-space formulation for 2d and 3d direct manipulation. In *Proc. UIST '09*, pages 69–78. ACM, 2009.

[26] J. Rekimoto. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *Proc. CHI '02*, pages 113–120. ACM, 2002.

[27] D. Schmidt, F. Block, and H. Gellersen. A comparison of direct and indirect multi-touch input for large surfaces. In *Proc. INTERACT '09*, pages 582–594. Springer-Verlag, 2009.

[28] A. L. Simeone and H. Gellersen. Comparing indirect and direct touch in a stereoscopic interaction task. In *Proc. 3DUI '15*, pages 105–108, March 2015.

[29] S. Voelker, A. Matviienko, J. Schöning, and J. Borchers. Combining direct and indirect touch input for interactive workspaces using gaze input. In *Proc. SUI '15*, pages 79–88. ACM, 2015.

[30] S. Voelker, C. Wacharamanotham, and J. Borchers. An evaluation of state switching methods for indirect touch systems. In *Proc. CHI '13*, pages 745–754. ACM, 2013.

[31] R. Wimmer, F. Hennecke, F. Schulz, S. Boring, A. Butz, and H. Hu. Curve: Revisiting the digital desk. In *Proc. NordiCHI '10*, pages 561–570. ACM, 2010.