Mockup Builder: Direct 3D Modeling On and Above the Surface in a Continuous Interaction Space

Bruno R. De Araùjo*, Géry Casiez† and Joaquim A. Jorge‡
*,‡INESC-ID, DEI IST, Technical University of Lisbon, Portugal
†LIFL, INRIA Lille, University of Lille, Villeneuve d'Ascq, France

Abstract

Our work introduces a semi-immersive environment for conceptual design where virtual mockups are obtained from gestures we aim to get closer to the way people conceive, create and manipulate three-dimensional shapes. We present on-and-above-the-surface interaction techniques following Guiard's asymmetric bimanual model to take advantage of the continuous interaction space for creating and editing 3D models in a stereoscopic environment. To allow for more expressive interactions, our approach continuously combines hand and finger tracking in the space above the table with multi-touch on its surface. This combination brings forth an alternative design environment where users can seamlessly switch between interacting on the surface or in the space above it depending on the task. Our approach integrates continuous space usage with bimanual interaction to provide an expressive set of 3D modeling operations. Preliminary trials with our experimental setup show this as a very promising avenue for further work.

Index Terms: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)—Input devices and strategies (e.g., mouse, touchscreen), Interaction styles (e.g. commands, menus, forms, direct manipulation);

1 Introduction

Despite the growing popularity of Virtual Environments, they have yet to replace desktop CAD systems when it comes to modeling 3D scenes. Traditional VR idioms are still umbilically connected to the desktop metaphor they aim to replace, by leveraging on the familiar Windows+Icons+Menus+Pointing (WIMP) metaphors. Worse, the command languages underlying many of these systems also do not map well to the way people learn to conceive, reason about and manipulate three-dimensional shapes. Another important obstacle, lies in that powerful modeling systems resort to constructive geometry and parametric formulations of a handful of primitives which run contrary to human perceptions and intuitions of space and physical models. As a result, users indirectly interact with models through widgets to control their parameters. However, new and affordable technologies such as depth cameras, multi-touch surfaces and multi-sensor devices motivate a fresh look at immersive interfaces. By providing more degrees of freedom, the new devices bear the promise of breaking from this mold by helping to develop interfaces that better support direct interaction. Indeed, these devices have the potential to support human modes of communication, such as sketching, gesturing and manipulating images and physical object as real-world proxies. Furthermore, many of these devices support a deeper use of human expression, such as two-handed manipulation, body posture, gaze and attention to name a few.

*e-mail: brar@vimmi.inesc-id.pt
†e-mail: gery.casiez@lifl.fr
‡e-mail: jorgej@acm.org

According to [9], immersive modeling brings three key advantages to the conceptual design process. First, they allow direct and real-time interaction. Second, users can work at full scale both in representation and interaction while being immersed. Finally in contrast to desktop systems, these attributes allow designers to get subjectively closer to their design ideas and work intuitively on their representation. Our strategy is to take full advantage of different interaction spaces and leverage their benefits for the tasks they are best designed for (e.g. using a flat surface for 2D sketching and 3D space for extruding an object). With these aims in mind, our goal is to develop a simple yet expressive system closer to the way people conceive, create and manipulate three-dimensional shapes. Thus, we devise a direct modeling approach taking advantage of sketching skills on 2D surface and gestures in 3D space operating seamlessly in the same immersive environment. These operations are fashioned following observations on how physical mock-ups are constructed manually and extend modeling operators from successful systems for fast prototyping such as Google Sketchup. Our direct modeling approach aims at interacting with the objects of interest without intermediate dialogues or gadgets, which promotes co-located interaction without sacrificing the expressivity power of the interface. Our immersive environment targets at supporting gestural and direct manipulation following the push and pull modeling paradigm to edit both topological and geometric representations of 3D models. By doing so, our goal is to propose plausible 3D gestures for modeling similar to physical mock-up interaction. Finally, we want to hide the underlying mathematical details associated to traditional CAD systems, thus bringing users into more intimate contact with virtual shapes without sacrificing their creativity. While we do not aim at working at full scale, the ability to control scale at will is an important feature to easily explore models. By using a god–like view, we intend to render virtual models as close as possible to physical mockup-ups without the associated physical constraints.

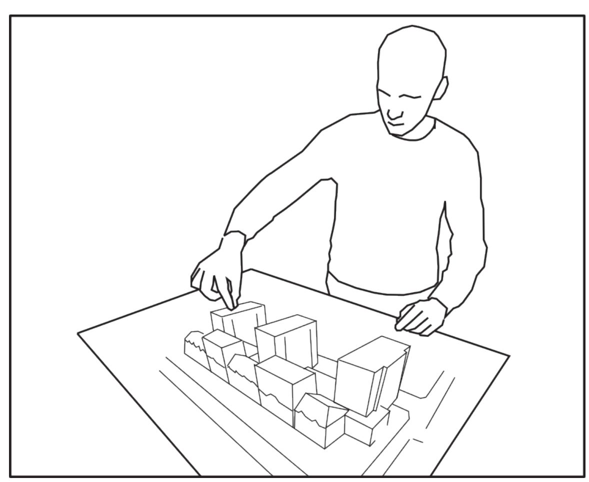

Figure 1: Mockup Builder Concept.

In this paper, we explore bi-manual and continuous interaction on and above multi-touch surfaces to bring direct modeling techniques to semi-immersive virtual environments. Our setup combines (1) stereoscopic visualization with (2) a multi-touch surface, (3) three-dimensional finger tracking and (4) a depth camera. In this way we can fuse four different but closely related human modalities to capture gaze, body posture, hand and finger position in synergistic ways. This rich sensing environment allows us to seamlessly pick and choose the sensing technique(s) most appropriate to each task. On this groundwork, we have developed an expressive set of modeling operations which build on user's abilities at creating and manipulating spatial objects. Indeed, from a small set of simple, yet powerful functions users are able to create moderately complex scenes with simple dialogues via direct manipulation of shapes. Noisy user input is continuously beautified to enable users to create smooth-looking forms by free-hand sketching. In this way, input gestures and strokes are continuously smoothed avoiding spurious artefacts and rendering shapes easier to create. Additionally, our setup affords continuous transitions between 3D (spatial) and 2D (planar surface) manipulations for modeling shapes extending the continuous space metaphor [26]. This allows users to issue gestures on and above the surface in an expected manner, e.g. extrusions of sketched shapes in contiguous, fluid gestures.

Another key feature of our approach lies in that it inherently supports bimanual asymmetric interactions. We adopt the Guiard's asymmetric model [15] for this purpose. This model proposes guidelines for designing bimanual operations based on observations of users sketching on paper. For these tasks, Guiard identifies different rules and actions for the preferred (also dominant-hand or DH) and non-preferred (also non-dominant hand, or NDH) hand. While the DH performs fine movements and manipulates tools, the NDH is used to set the spatial frame of reference and issue coarse movements. Moreover, people do not explicitly switch between defining the spatial frame of reference and manipulating tools.

We developed different interaction metaphors according to three main criteria: the location of the gesture, the participating hand(s) and the continuity of hand movements. This distinctive feature of our work combines continuous space and the Guiard asymmetric model harmoniously in a single definition of mode. Furthermore, it allows seamless and rapid mode switching in straight-forward ways, greatly contributing to the overall expressiveness of the interface using simpler dialogues.

The rest of the paper is organized as follows. After covering the related work we present our approach and describe in detail the experimental setup. We then explain the processing of input data and how the different modalities are fused. The remaining sections explain our apporaches to bimanual interaction and how we explore the continuous space to derive a simple yet very expressive set of modeling operations. Preliminary assessments and trials of our techniques show promise and encourage us to further pursue this avenue in the future.

2 Related Work

This section discusses research in three main areas relevant to our work. We first present non traditional modeling interfaces using gestures in the air, tangible objects, sketching and haptic feedback. We then cover bimanual interaction and follow with the on-and-above the surface related approaches.

Modeling System Interfaces. Schkolne *et al.* [32] introduced Surface Drawing using hand motion in the air to describe ribbon like shapes based on hand posture. Additionally, a set of tangible tracked artifacts were available, each with its own functionality. For example, kitchen tongs to pick objects, a magnet tool to deform objects or a squeezable object to delete parts of an object. While this approach allows creating free-form shapes, it appears inadequate to create rigorous manufactured shapes. FreeDrawer [34] alleviates this issue by providing a tracked stylus allowing the user to sketch networks of curves on top of a Responsive Workbench. These curves can then be used to define the boundary of free-form surfaces that can be deformed interactively. However more complex CAD editing and primitives are still out of the scope of such approach. Fleish *et al.* [11] support both freeform shape creation and regular CAD primitives by adapting traditional WIMP based interfaces to virtual immersive environment using a PIPSheet artifact. The PIPSheet is a tracked transparent glass where menus can be seen through, creating the illusion that the user interface is rendered on the glass surface. Items can be selected using a tracked stylus. Using head mounted displays, such system can be used by several users in a collaborative way to support the designing task as presented by Kaufmann [19]. Their system was used to introduce CAD modeling operations to students allowing creating simple regular primitives such as prisms, pyramids, cones and cylinders in the air. However the lack of physical support makes drawing in the air more adequate for free form modeling than to create CSG like regular objects [31]. Haptic devices can help sketching in the air although the working space is often restricted [20]. This provides an attractive solution for 3D modeling since users are able to easily learn how to use these systems and rigor improves rapidly with training as shown by recent studies [35]. Instead of only relying on gestures in the air, our approach takes advantage of both the surface and space above it, which aims at combining the benefits of both interaction spaces.

Sketching is a powerful communication tool of any real conceptual design task. However, it is still discarded by most of existing CAD modeling systems which rely primarily on single cursor based interaction and WIMP metaphor. Regarding traditional 2D environments, research on sketch based modeling interfaces has proposed several approaches to take advantage of designer drawing skills. Olsen presented a deep survey of most of the existing techniques [30]. These systems rely on gesture recognition (SKETCH), stroke beautification (Pegasus), line drawing reconstruction (SmartPaper), suggestive interfaces (Chateau), push pull sketching (Sesame [29]), freeform contour based inflation (Teddy or ShapeShop) to make sketching as a usable alternative to traditional CAD systems. Forsberg *et al.* [13] propose an adaptation of the SKETCH system to a stereoscopic ActiveDesk environment named ErgoDesk. However, they still rely exclusively on 2D gestures to create geometry using a light pen and the stereoscopic visualization is primary used for 3D exploration of shapes using a 6DoF tracker. Our approach adopts several of these techniques to go further than existing drawing-in-the-air approaches while mixing 2D sketch with 3D gestures continuously. We rely on the physical surface of a multi-touch device as a fixed drawing canvas, to free hands from holding devices used as a moving canvas in 3D space [37, 21]. We use sketch on the surface combined with gesture above the surface to define 3D trajectories while the user is experiencing a stereoscopic visualization more adequate to 3D perception. Alternatively, tangible interfaces have been used in space or on tabletop. Tangible interfaces offer natural manipulations and artifacts can correctly map tools functionality [32]. They can be as effective or even better than WIMP interfaces for 3D manipulation and edition as demonstrated by [28]. They can also be used to create 3D models such as Jota *et al.* [18] using wooden blocks of different shapes. Using a Kinect camera, the position and shape of the blocks can be captured and 3D simple scenes can be created by assembling blocks while the user is viewing the scene in stereo. Commands and plane height control are issued using an additional mobile device used as an operation console, stacking captured blocks on top of virtual content. In contrast with tangible interfaces, our approach is not limited to physical representations and provides an unconstrained designing environment regarding shape representation.

Bimanual Sketching. Bimanual interaction is a fundamental concept to our approach in that we expect to leverage the higher bandwidth provided by two–handed gestures. Hands can have an asymmetric or symmetric role [15]. Asymmetric bimanual interaction attributes different roles to each hand as presented by Balakrishnan and Kurtenbach [4] where the NDH controls a virtual camera, defining a frame of reference for the DH which manipulates objects. With symmetrical bimanual interaction, both hands have a similar role adapted to the task. While the symmetric model [3, 22, 23] has proved to be more adequate to support exclusive spatial tasks or describe shapes with hands, the asymmetric model makes it possible to take advantage of a natural task switching between hands. Initially, methods have been proposed that mimick existing asymmetric tasks such as the automotive tape drawing techniques [2, 14]. Using that approach, users create curves on a large scale display at a one to one scale using both hands. Other approaches associate different roles to each hand [4, 34]. Usually they use the NDH to manipulate objects or the view and the DH for editing as suggested by the Guiard asymmetric model [15]. To wit, our approach takes advantage of both asymmetrical and symmetrical hand operations. Most operations assign asymmetrical roles to each hand. However, for tasks such as scaling and rotating shapes on the surface or in the air, it is more natural to use symmetric assignments [33]. The IloveSketch system [1] adapts such a concept in traditional 2D sketch based modeling interfaces allowing users to control the virtual camera or 3D planes using a keyboard, while the other hand sketches on the 3D scene using a pen tablet to create curve wireframe models. While this approach is bimanual, it does not engage the hand directly – it operates two devices, the keyboard and the stylus pen. Other systems [5, 17, 24, 25] have explored the bimanual asymmetric model by combining finger- or hand- gestures with pen devices. Brandl *et al.* proposed a sketching system where the user selects options through touches using the NDH on a WIMP–based graphical interface, while the DH is used to sketch using a pen device [5]. Such a configuration allows to better explore hand gestures proposing richer interaction concepts to represent 2D editing operations such as demonstrated by Hinckley *et al.* [17]. Indeed, this makes switching between modalities easier and allows users to perform a wide range of 2D editing tasks without relying on gestures or GUI invocations. Lee combined hand gestures while sketching using a collapsible pen to define curve depth on a tabletop [24]. The NDH is tracked allowing users to seamlessly specify 3D modeling commands or modes such as the normal direction of an extrusion while specifying the displacement by interacting with the pen on the virtual scene. Contrary to their approach, we preferred to keep the surface for fast and accurate 2D drawing, while benefiting from the 3D input space for controlling depth directly. Lopes *et al.* adapted the ShapeShop sketch based free-form modeler to use both pen and multi-touch simultaneously [25]. They found out that the asymmetric bimanual model allows users to perform more manipulations in less time than conventional single interaction point interfaces, which increased the percentage of time spent on sketching and modeling tasks. By tracking the hands of the user, we adopt the asymmetric bimanual model to easily switch between sketching, model editing, navigation and spatial manipulation of objects. In addition, we do not need to rely on special input devices nor extra modalities to assign different roles to each hand.

"On" and "Above" Surface Interaction. With the widespread adoption of multi-touch devices and less expensive and intrusive tracking solutions such as the Microsoft Kinect, academic research on tabletop has refocused on "on" and "above" surface interaction techniques. Müller-Tomfelde *et al.* proposed different methods to use the space above the surface to provide ways of interacting with 2D tabletop content closer to reality [27]. While tangible devices complement the surface physically with a direct mapping to the GUI such as in the Photohelix system and StereoBlocks [18], finger gestures above the surface mimic physical interaction with real objects. Furthermore, instead of considering only finger touches, full hand posture on the surface can also be detected to provide richer interaction metaphors. Above the surface, the hand distance from the surface defines depth in 3D space giving a new dimension to the interactive region [27]. Wilson *et al.* proposed several metaphors to interact with different displays while capturing full body posture [36]. In this way, users can interact on or above the surface with 2D content or even between surfaces using the body to transfer virtual content to the hand or to another surface while moving their bodies in space. Users can also interact physically in space with projected GUI. In our system, we prefer to use the surface for GUI since it is more adequate for discrete selection and explore space gesture for modeling actions. Our approach explores the continuous space as presented by Marquardt *et al.* [26]; however we enrich their approach by combining it with the bimanual asymmetric model proposed by Guiard [15]. In addition, we rely on a stereoscopic visualization setup for architectural model visualization similar to [8]. While this system allows navigating or annotating the 3D scene mainly as if it was inside the table and use fingers as proxies over the scene, our interaction techniques focus on modeling and direct manipulation since 3D models are rendered as if they were lying atop the table. To avoid hands occlusions over the visualization, Toucheo [16] proposed a fish-tank like setup using a multi-touch surface and a stereoscopic display. However such as other setups relying on semi-transparent mirrors to create holographic illusion, it both reduces the working space and constrains the usage of the above surface space to hand gestures. Our stereoscopic visualization setup provides more freedom of movement allowing a continuous space of interaction. In addition, adopting a bimanual asymmetric model makes possible new interaction techniques which could benefit interaction with holographic display technologies when they become available.

3 Our Direct Modeling Approach

We propose a direct modeling approach to create, edit and manipulate 3D models using a small set of operations. Users interact through multi-touch gestures on a surface and gestures in space tracked by Gametrak[1] devices. Multi-touch gestures can also be used for sketching allowing to create 3D models by pushing and pulling existing content off the scene. Our models are represented using a boundary representation which decomposes the topology of objects into faces, edges and vertexes. Faces represent finite planar polygons or even surfaces delimited by edges. Edges are abstractions of segments or curves represented as 3D cubic Bézier parametric curves. These three kinds of topological features can be selected and edited by the user using direct manipulation in 3D space as explained below. Our push and pull approach proposes five operations. The simplest allows displacing topological features along a normal direction to change the geometry of the object without altering its topology. The second operation extrudes a face along the normal to extend the topology with new sided faces along the selected face. The third is a curvilinear extrusion which extends a shape by extruding a face along a path defined by a user gesture either in 3D space or on the surface. The fourth enables splitting faces by sketching linear or curvilinear strokes on them, subdividing those faces into more complex features. Finally, a snapping operation allows easily switching between surface and space editing when needed. This simple set of operations combined with modifiers (see Section 8) allows to create complex shapes through sketches and gestures using the same push and pull language as Google Sketchup or Sesame [29].

[1] See http://en.wikipedia.org/wiki/Gametrak for details.

3.1 User Inputs as Sketches or Gestures

We choose fingers tracking as our main input modality captured by the multi-touch surface when user touches it and by the Gametrak device once above. However to use such input data into sketches or gestures, we start by filtering the Gametrak data to remove the spatial jitter coming from the device and the user using the 1€ filter [6]. This data is then stored as an input gesture and updated continuously. While it is updated, input data is fitted incrementally to the best fit of lines and cubic Bézier curves. Thanks to this transformation, input gestures can be used as strokes creating shapes with sharp features or as gestures defining smooth trajectories. Our incremental fitting algorithm based on curve fitting tries to guarantee the continuity between curves and segments by adding tangency constraints during the fitting process without loosing fine details. This process also guarantees a maximal error distance of 7 millimeters between the raw and smoothed trajectories. This curve and line approximation is used for both sketches and gestures above the surface in place of the raw input data. While trajectory or 3D strokes could be defined directly using such representation, an additional beautification step is done on sketches to ease the creation of regular shapes. When a closed contour is created on the surface, further constraints are applied based on line segments to detect parallel and perpendicular line pairs and segment pairs with equal length. We use a threshold on angles between segments for parallelism and perpendicularity and a threshold ratio relationship between segments with similar length. An energy function is specified for each type of constraint and we perform an error minimization method to beautify user sketches. Regarding closed conic sections, we use a 2D shape recognizer [12] to detect circles and ellipses which are approximated by a closed piecewise curve using four cubic Bézier segments. This recognizer is also used to detect a simple erasing gesture used to delete shapes or strokes.

3.2 Selecting Modeling Parts

Selecting shapes or part of them is critical to any direct manipulation based approach. While this is done implicitly by touching a geometrical feature on the surface, we choose to use an explicit pinch gesture in space mimicking a grabbing gesture of physical objects. Visual feedback on shapes and geometrical features is provided based on their proximity with fingers.

Several selections can be performed with different granularity since any topological feature from our boundary representation can be edited. A whole shape can be selected by intersecting its bounding box with a finger. Intersecting a face, edge or vertex highlights it for selection. Since edges and vertices can be shared by more than one face or edge respectively, a continuous selection mechanism is provided to disambiguate the selection by analyzing the previously highlighted entity. For example, it is possible to highlight a particular edge of face shared by two faces by selecting it from the face the user is interested in. Empty selections, which are useful for scene manipulation, are possible both on the surface or in the space above it by simply selecting an empty area of the scene (i.e. one that does not intersect any bounding box of a shape).

3.3 Transitioning between Surface and Space

Creating 3D planar shapes in space remains an operation difficult to perform due to lack of physical constraints to guide the hand. We propose a snapping operator to easily switch between the surface and space allowing to use sketches on the surface or gestures in 3D space at convenience. Snapping is available through the contextual menu accessible on the NDH to snap on or back on any selected face. It works by computing a transformation matrix to align the 3D scene to the visible grid defined as a representation of the table surface. A simple linear animation between the two orientations is rendered to help the user understand the new orientation of the model. Furthermore, it allows sketching details on existing shapes or guaranteeing that new shapes are created on top of an existing shape. Additionally, since existing objects can occlude the selected face when snapping is performed, we give to the user the possibility to clip part of the scene using our menu. It is implemented using traditional OpenGL clipping planes defined as lying on the surface.

4 Hardware Modeling Setup

Our setup consists in a semi-immersive environment based on a stereoscopic multi-touch display 96×72 cm (42 inches) combined with a Kinect depth camera and two Gametraks used to identify and track the hands and fingers above the surface.

Head tracking is achieved in a non-intrusive way thanks to the Kinect using its skeleton detection algorithm. The skeleton is also used to track user hands allowing to locate the dominant hand according to the handedness of the user. Finger tracking is operated through multi-touch on the surface and using Gametrak devices in space (Figure 2). The visualization relies on a back-projection based system located under the table running at 120 Hz with a 1024 × 768 pixels resolution giving a pixel density of 10.6 pixels per cm (27 DPI). It is coupled with active shutter glasses from 3D Vision NVIDIA for the stereoscopic visualization.The 3D scene is rendered on top of the surface and the point of view is updated according to the position and orientation of the user's head to take into account motion parallax. The IR transmitter for the glasses uses an IR wavelength different from the multi-touch table which is based on the Diffuse Illumination technique. It is set at a position to cover the working volume around the table where the user interacts.

A camera running at 120 Hz with a 640×480 pixels resolution and positioned under the surface records finger movements on the surface, providing a maximum resolution of 6.4 dots per cm (16.25 DPI) for finger tracking. We use the iLight[2] framework version 1.6 for fingers detection and tracking. Fingers data are then sent using TUIO messages to our custom built application.

The two Gametraks are used to track the 3D position of the index and thumb of each hand when they are no longer in contact with the

[2]iliGHT Tactile Table product page: http://www.immersion.fr

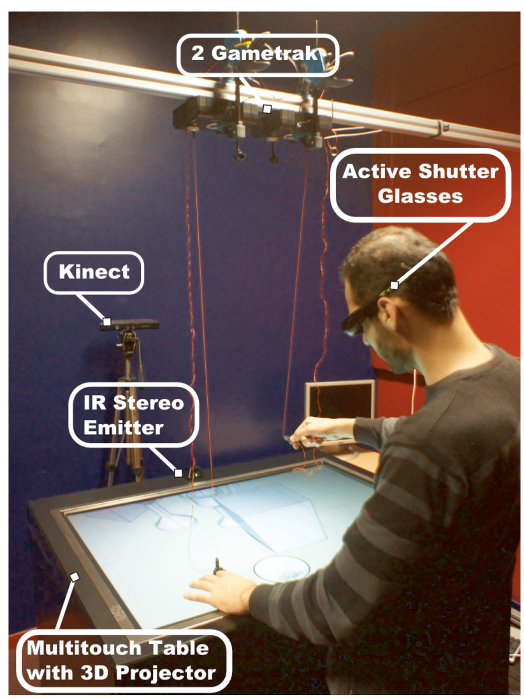

Figure 2: Overview of the setup.

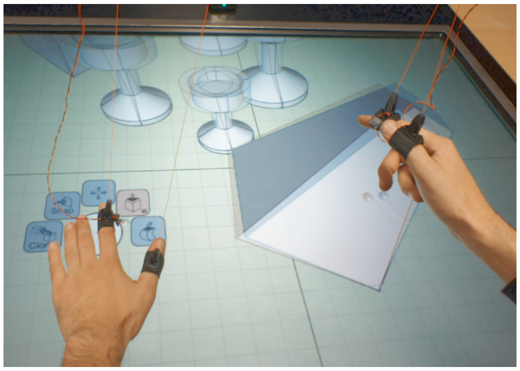

Figure 3: Detailed view of the Gametrak strings attached to the fingers with the buttons used for pinch gestures

multi-touch surface. These low cost gaming devices are placed in a reverse position centered above the table at a distance of 120 cm. The 3D position of each finger is computed from the two angles of rotation and the length of each cable, digitalized on 16 bits and reported at 125Hz to the host computer, resulting in a theoretical position resolution going from 500 dots per cm (1250 DPI) when the finger is close to the surface to 900 dots per cm (2250 DPI) when it is 50 cm above it. However the effective resolution is far lower (around 10 DPI) due to measurement noise. The retractable strings are attached to the fingers through a ring. Although strings introduce some visual clutter, they were not found to distract users from their task. The strings create a minor spring effect which reduces user hand tremor without adding fatigue. We added a 6mm diameter low profile momentary switch button on each index finger to detect pinch gestures without ambiguity (Figure 3). This simple solution provides a good trade-off regarding precision, cost and cumbersomeness compared to using a high end marker based optical tracking system or low sampling frequency (30 Hz) device such as the Kinect. The latter presents also a low tracking resolution (from 3 to 8 DPI) and is subject to finger occlusion.

The redundancy of information from the different input devices allows us to identify which finger of which hand is interacting on the surface or in the air or to choose the input source with the best tracking resolution.

5 Interpreting Input Data

Our setup relies on several input devices which should be on the same coordinate system to obtain a continuous interaction space. We chose the Kinect coordinate system as our primary coordinate system since it covers both the working and the user spaces. This section explains how we calibrate our continuous interaction space and how input data is fused into a single user model.

5.1 Calibrating Multi-touch Input Data

We provide a simple application for the user to pick the four corners of the multi-touch display in an image captured by the Kinect. These four points coupled with the 3D coordinate extracted from the Kinect depth map are used to compute the plane which minimizes the distance between them. The plane is then used to define two matrices converting touches on the surface into 3D positions and vice versa. Figure 4 presents a screenshot of our calibration application allowing the user to assess the correctness of the calibration thanks to a 3D preview of the plane and its mesh representation captured by the Kinect. The screen plane definition is used to define the frustum of the off-axis stereo perspective projection to render 3D content on top of the surface from the user point of view.

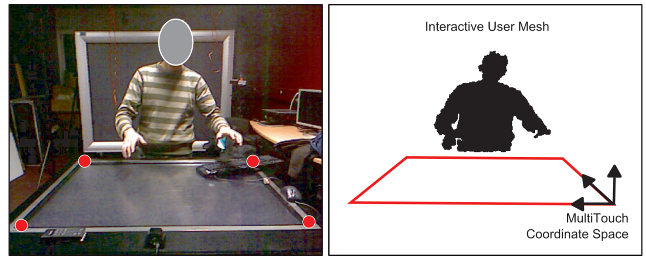

Figure 4: Calibrating 2D Touches: Kinect image camera with the four corner points selected by the user (red dots) on the left, 3D view of the user with the resulting screen plane on the right

5.2 Calibrating Gametrak Input Data

Gametrak input data is defined in a framework centered on the device base, requiring the computation of a transformation matrix into our primary coordinate system for each tracked finger. This is done using a set of one thousand matching 3D position pairs to compute the correspondence rigid transformation. The set is created by sampling the multi-touch surface screen and gathering the touch positions converted to our primary coordinate system using the matrix defined on the previous section. The rigid transformation is computed using a RANSAC algorithm [10], creating a matrix mapping Gametrak positions to our global coordinate system.

5.3 Fusing Inputs into a Single User Model

All input data that belong to the same finger are fused together as an input gesture. An input gesture might represent a stroke or gesture on or above the surface. Data coming from the multi-touch surface or the Gametraks has a unique identifier defined by the input device. After the coordinates have been converted into the same coordinate system, the fusing consists in determining when the identifiers from different sources correspond to the same finger. It also consists in adding the handedness information to each finger. A new input gesture is created when a finger touches the multi-touch surface without doing any pinch gesture, or when the finger performs the pinch and that finger was not touching the surface before. Input gestures are deleted when fingers are lifted from the surface without any pinching or when the pinch button is released above the surface. Otherwise the input gesture is updated. Multi-touch and Gametrak data are fused together based on close proximity. When a finger is on the multi-touch surface, we discard Gametrak data even if they are available as they were found to be less reliable. When a new input gesture is created, input handedness is determined by the closest hand position obtained from the Kinect skeleton.

6 Bimanual Interaction On the Surface

The multi-touch surface is primarily used as a sketching canvas where the user interacts using fingers. As previously explained, we followed the Guiard bimanual asymmetric model allowing the

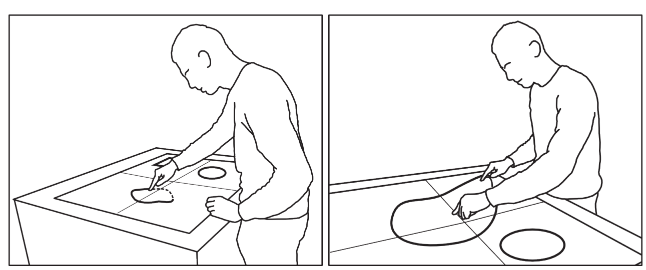

Figure 5: Bimanual Interaction on the Surface: Sketching using the DH (left) and scaling with both hands starting with the NDH (right).

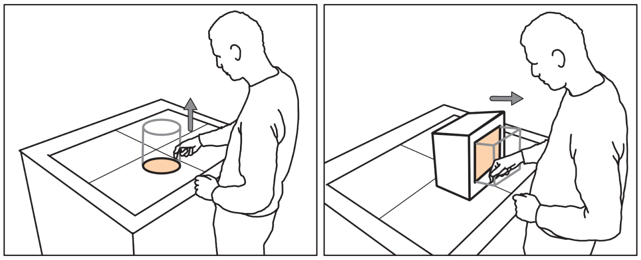

Figure 6: Face straight extrusion: along the surface normal direction (left), along a face normal direction (right).

user to implicitly switch between sketching tasks and object transformation / world manipulation (scale, rotate, translate operations on objects or on the world) depending on the hand used. Using the DH, user can sketch on the surface creating planar shapes from close contours. Contours might use lines, curves or both and can be sketched using multiple strokes. Open strokes whose extremities are close to each other are merged into a single stroke. Topological shape features are highlighted if a touch selection is performed nearby. Additionally, planar faces can be sub-divided into an arbitrary number of faces with different shapes if a face is overlapped by an open stroke starting and finishing outside that face. As explained in Section 3.1, strokes are automatically fitted into lines and curves ready to be used as sketch. However, we also use a 2D shape recognizer [12] allowing detecting simple gestures such as an erasing command by drawing a scribble. When an erasing gesture is recognized, if it overlaps open strokes, they are erased. However, if it overlaps only shapes and not open strokes, overlapped shapes are erased. This solution allows to use open strokes as construction lines while modeling.

When starting a gesture on the surface with the NDH, it is interpreted as object transformation if it is performed on an object, or world manipulation otherwise. Single touch gestures are interpreted as object or world translation. More than one finger gestures are interpreted as translation, rotation and scale operations on objects or world. 3D objects are constrained to movements along the plane parallel to the multi-touch surface. A gesture started with the NDH can be complemented by the DH allowing translation, rotation and scale with both hands (Figure 5).

Furthermore, bimanual interaction can be used to constrain drawing operations. In which case, the NDH defines constraints for the DH. For example, a user can sketch a straight line defining a plane of symmetry. First, the user selects the straight line using his NDH and sketches using the DH. As a result, the shapes sketched with the DH are mirrored by the plane of symmetry.

7 Continuous Interaction Above the Surface

Gestures with the DH above the surface are interpreted as 3D object creation or edition. Creation consists in extruding a planar shape

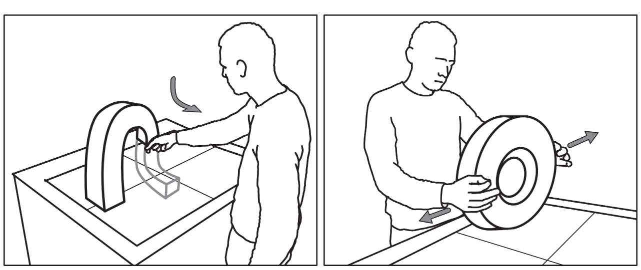

Figure 7: Extrusion along a curve gesture (left), 3D object scaling using both hands (right).

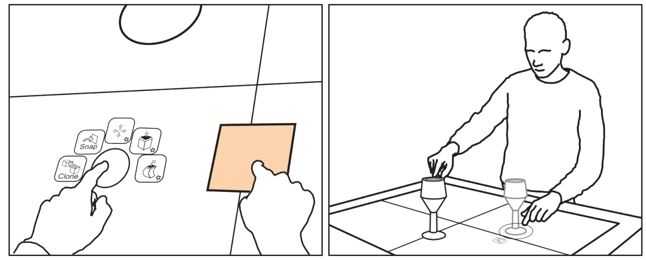

Figure 8: Example of menu presented under the NDH (left), cloning an object using both Hands (right)

previously sketched on the surface. The user first approaches the DH index finger near a shape on the surface to highlight it. He then performs a pinch gesture to extrude the shape along the normal of the surface (Figure 6). The height of the extruded object is then continuously updated and co-located with the finger position until the button is released. Planar shapes can also be extruded along the trajectory defined in the air after the user has selected this operation in a menu displayed on the NDH (Figure 7). While the user is defining the trajectory, the path is continuously re-evaluated and fitted into line segments and curve pieces to create a beautifulized freeform shape. Segments and curve pieces are created using the approach proposed by Coquillart [7] to offset the gesture from the centroid of the face to its vertexes and create a smooth free form extrusion of the profile. This method allows to extrude both polyline and curvilinear profiles along linear or curvilinear paths.

Editing follows the push and pull modeling metaphor where topological features of the shape (vertexes, edges and faces) are moved in the air along the normal direction of the face it belongs to. As described in Section 3.2, our continuous selection method allows to distinguish which face an edge or a vertex belongs to if needed. The user first highlights the geometrical feature by moving his DH index finger close to it. He then selects it with a pinch gesture. The position of the geometrical feature is then updated according to the finger position until the pinch gesture is released. Alternatively faces can be extruded along to their normal or following the trajectory defined by the user after the corresponding operation has been selected in the menu displayed on the NDH. If no geometrical feature is selected while doing the pinch gesture with the DH, the user can sketch 3D poly-lines or curves in space.

The bimanual interaction used on the surface is also valid above the surface allowing to rotate, translate and scale objects using two fingers. As on the surface, the NDH begins the interaction using a pinch gesture. The NDH defines translations only while the DH adds rotation and scale operations using the method proposed by Wang *et al.* [33]. These direct 3D object manipulations appear much more efficient compared to indirect interactions on the multi-touch surface alone (e.g. changing the depth of an object while translating it along the surface plane).

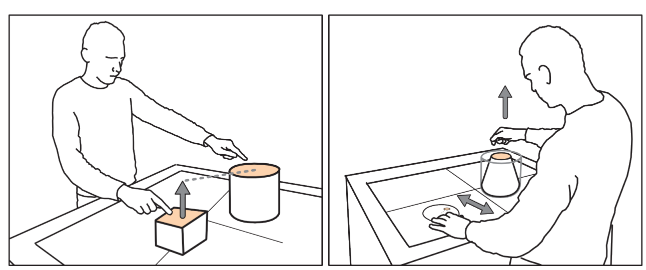

Figure 9: Defining an height constraint with the NDH (left), scaling with the NDH while extruding a shape (right).

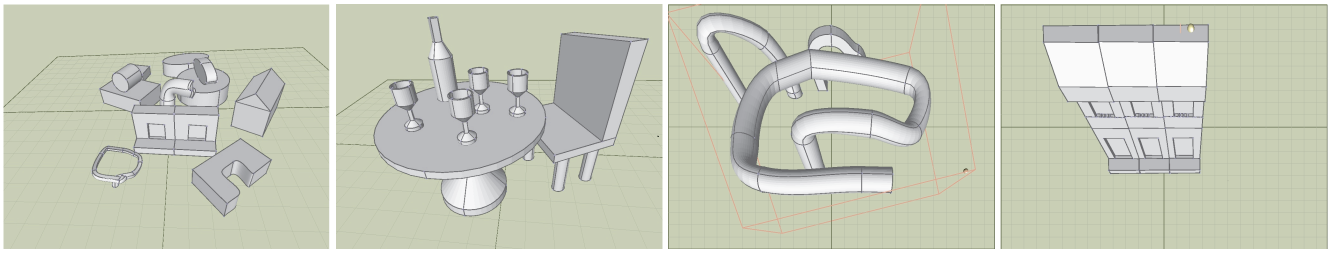

Figure 10: 3D models designed using Mockup Builder (from left to right): a set of shapes, a table with a chair, three different types of curved extruded profiles and a simple building façade. The last two images are rendered from the user point of view.

8 Exploring On and Above the Surface Interaction

We have previously used asymmetric hand operations to implicitly switch between sketching, object transformation and world manipulation. We now illustrates how the NDH can complement the operations performed by the DH with three types of operations.

First, the NDH can be used to select the mode used by the DH. Modes are presented through items shown in a contextual menu presented under the NDH. Modes presented in the contextual menu correspond to the ones available in the current mode associated to the operation performed by the DH (Figure 8). If the operation carried by the DH hand only supports a single mode, no contextual menu is shown under the NDH. To avoid visual clutter, the contextual menu transparency is adjusted based on the distance between the NDH and the surface. Above 15 cm, the menu is fully transparent and becomes progressively opaque as the NDH approaches the surface. To improve the accessibility, the contextual menu follows the NDH but its location is progressively fixed as the NDH comes closer to the surface to avoid spatial instabilities and reducing errors while selecting an item. This is simply done using the 1€ filter and adjusting its cutoff frequency based on the distance[6].

The discrete mode selection includes the type of extrusion (normal to a face or along a trajectory), the cloning operation and the snapping operation. Once in the cloning mode, discrete touches with the NDH define the location where clones appear. Snapping is available when a face is selected. It consists in rotating the world to align the face with the surface.

Instead of defining discrete operations through a contextual menu, the NDH can be used to select a geometrical feature that defines a constraint for the DH. The constraint is enabled as long as the NDH keeps his selection active. We use plane and line constraints in the extrusion and positioning operations. For example, the NDH can select a face of an object to define the maximum or minimum height for an object being extruded with the DH. Once the constraint is defined, the user continues to move his DH until the maximum or minimum height is reached. Further movements along the preceding direction do not continue to update the height of the object. This allows the user to also define that the height of an object should not be higher or lower that the height of another object. When translating an object, a plane constraint defines a limit beyond which an object cannot be moved further. While traditional modeling interfaces define constraints in a sequential way, we hypothesis that this definition of constraints on the fly allows to improve the flow of interaction.

Instead of defining discrete operations with the NDH, our last category of operations explores the usage of constrains continuously updated by the NDH. This is illustrated with the scale constraint that consists in scaling the profile while extruding a shape (Figure 7). This allows to create a cone or a frustum from a circle or a quadrilateral planar face respectively. The scaling factor can be controlled dynamically using a 2D overlay menu accessible by the NDH while extruding the shape.

9 Discussion and Conclusion

We have described an approach to model 3D scenes in a direct way using semi-immersive virtual environments through a synergistic combination of modalities afforded by novel input devices. Our system and experimental setup show that it is possible to enhance interaction by fusing data coming from different sensors. This provides a plausible environment combining benefits of multi-touch and stereo, using simple 3D operators, to model shapes using direct manipulation and simpler dialogues as compared to traditional and current systems. Combining the power of *bimanual interaction* with the flexibility of *continuous space*, we can provide effortless transition between modes and make it simple to switch between multi-touch 2D and spatial 3D gestures. This allows selecting the manipulations best suited to each task in non-obtrusive ways.

We implemented a prototype to demonstrate our modeling approach in C++ using OpenGL and OpenSG for stereoscopic visualization. Our system was deployed on an Intel I7 920 2.67 GHz processor with 3 Gb of memory RAM and an NVidia Quadro 4000 graphics card running Microsoft Windows 7 64-bit operating system. These first results are very encouraging and seemingly support further exploring our chosen avenue of work. Along the development, around 20 undergraduate and graduate students in Computer Science with variable experience with CAD applications and one Architectural researcher tested the system. They informally assessed the different design choices and iteratively improved the design of the interface. We plan to run formal evaluations with both novice and expert users to highlight and explore both the strengths and the weakness of our modeling interface. The remaining of the section discusses our achievements regarding our initial goal which was to provide a direct modeling solution.

Thanks to stereo, we provide co-location between user hands and virtual objects adapted to direct modeling methods. While the initial version used physics to detect collisions, this proved problematic while modeling. The feature was discarded instead of being activated on request. However it could be advantageous both for stacking and supporting 3D manipulations. While sketching is beneficial to surface–based interactions, beautification is a must to support creating more rigorous shapes for manufacturable objects. Figure 10 presents different models built using the interface by an expert user. As an example the second model from the left was built in 5’20” while the fourth took one of us 2’45” to complete. An expert user took 5’41” and 3’46” respectively for the same models using Rhino3D modeler. More rigorous tests should yield more exact measures, while direct editing of curves should be considered to reduce user retrials. On a positive note, the continuous interaction provides plausible gestures for extrusion and easy to define 3D trajectories leveraging the best features of the surface and space above it. While the surface invites users to sketch, the space above invites gestures and the snapping feature provides a suitable solution to transition between the two. In sum, *bimanual asymmetric interaction* provides an implicit switch between modeling and manipulation, letting the user focus on his design tasks. However it might be confusing for some users, in particular when interacting with a large multi-touch surface. That is why we allow users to scale objects using both hands if they so do wish. Still, users should heed the precedence of the non–dominant hand. As in other sketching applications, menus could not be avoided altogether and are still required in particular when selecting from several modeling operations. However, providing a scaling widget while extruding provides an efficient separation of the degrees of freedom. We are considering to further explore multiple finger tracking as an alternative using non ambiguous start and end gestures. While speech as a modality could overcome such problems or alleviate the need for menus, on an interactive tabletop, button-like activation is likely more efficient and immune to recognition errors.

The system shows clear promise and provides a good case for augmenting interactive surfaces with gesturing gaze and body posture to support interactive modeling operations. The approach can be further extended by exploring combinations of different modalities and experimenting with mode-inferencing to further enhance the fluidity of our modeling techniques.

ACKNOWLEDGEMENTS This work was partially funded by the ANR InSTInCT project (ANR-09-CORD-013) , the Interreg IV-A 2 seas SHIVA project and by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds, doctoral grant reference SFRH/BD/31020/2006 and MIVIS project PTDC/EIA-EIA/104031/2008.

REFERENCES

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

[2] R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and W. Buxton. Digital tape drawing. In *Proc. of UIST '99*, pages 161–169, 1999.

[3] R. Balakrishnan and K. Hinckley. Symmetric bimanual interaction. In *Proceedings of CHI '00*, pages 33–40, NY, USA, 2000. ACM.

[4] R. Balakrishnan and G. Kurtenbach. Exploring bimanual camera control and object manipulation in 3d graphics interfaces. In *Proceedings of CHI '99*, pages 56–62, NY, USA, 1999. ACM.

[5] P. Brandl, C. Forlines, D. Wigdor, M. Haller, and C. Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proc. of the working conference on Advanced visual interfaces*, AVI '08, pages 154–161, NY, USA, 2008.

[6] G. Casiez, N. Roussel, and D. Vogel. 1€ filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of CHI'12*. ACM, 2012.

[7] S. Coquillart. Computing offsets of b-spline curves. *Comput. Aided Des.*, 19:305–309, July 1987.

[8] J. B. De la Rivière, N. Dittlo, E. Orvain, C. Kervégant, M. Courtois, and T. Da Luz. iliGHT 3d touch: a multiview multitouch surface for 3d content visualization and viewpoint sharing. In *Proceedings of ITS '10*, pages 312–312, New York, NY, USA, 2010. ACM.

[9] J. Deisinger, R. Blach, M. Wesche, R. Breining, and A. Simon. Towards immersive modeling - challenges and recommendations: A workshop analyzing the needs of designers. In *Proc. of the 6th Eurographics Workshop on Virtual Environments*, pages 145–156, 2000.

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981.

[11] T. Fleisch, G. Brunetti, P. Santos, and A. Stork. Stroke-input methods for immersive styling environments. In *Shape Modeling and Applications*, pages 275–283, Los Alamitos, CA, USA, 2004. IEEE.

[12] M. Fonseca and J. Jorge. Using fuzzy logic to recognize geometric shapes interactively. In *The Ninth IEEE International Conference on Fuzzy Systems*, volume 1, pages 291 –296, 2000.

[13] A. S. Forsberg, J. J. L. Jr., and R. C. Zeleznik. Ergodesk: A framework for two- and three-dimensional interaction at the activedesk. In *Proc. of Immersive Projection Technology Workshop*, pages 11–12, 1998.

[14] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton. Creating principal 3d curves with digital tape drawing. In *Proc. of CHI '02*, pages 121–128, NY, USA, 2002.

[15] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19:486–517, 1987.

[16] M. Hachet, B. Bossavit, A. Cohé, and J.-B. de la Rivière. Toucheo: multitouch and stereo combined in a seamless workspace. In *Proceedings of UIST '11*, pages 587–592, NY, USA, 2011. ACM.

[17] K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton. Pen + touch = new tools. In *Proceedings of UIST '10*, pages 27–36, NY, USA, 2010.

[18] R. Jota and H. Benko. Constructing virtual 3d models with physical building blocks. In *Proceedings of CHI EA '11*, pages 2173–2178, New York, NY, USA, 2011. ACM.

[19] H. Kaufmann and D. Schmalstieg. Designing immersive virtual reality for geometry education. In *Proceedings of the IEEE conference on Virtual Reality*, VR '06, pages 51–58, DC, USA, 2006. IEEE.

[20] D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Drawing on air: Input techniques for controlled 3d line illustration. *IEEE Trans. Vis. Comput. Graph.*, 13(5):1067–1081, 2007.

[21] P. Lapides, E. Sharlin, M. C. Sousa, and L. Streit. The 3d tractus: A three-dimensional drawing board. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 169–176, DC, USA, 2006. IEEE Computer Society.

[22] C. Latulipe, C. S. Kaplan, and C. L. A. Clarke. Bimanual and unimanual image alignment: an evaluation of mouse-based techniques. In *Proceedings of UIST '05*, pages 123–131, NY, USA, 2005.

[23] A. Leal, D. Bowman, L. Schaefer, F. Quek, and C. K. Stiles. 3d sketching using interactive fabric for tangible and bimanual input. In *Proceedings of Graphics Interface 2011*, GI '11, pages 49–56, 2011.

[24] J. Lee and H. Ishii. Beyond: collapsible tools and gestures for computational design. In *Proceedings of CHI EA '10*, pages 3931–3936, New York, NY, USA, 2010. ACM.

[25] P. Lopes, D. Mendes, B. Araújo, and J. A. Jorge. Combining bimanual manipulation and pen-based input for 3d modelling. In *Proceedings of SBIM '11*, pages 15–22, New York, NY, USA, 2011. ACM.

[26] N. Marquardt, R. Jota, S. Greenberg, and J. A. Jorge. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of INTERACT'11*, pages 461–476, Berlin, Heidelberg, 2011. Springer-Verlag.

[27] C. Müller-Tomfelde, O. Hilliges, A. Butz, S. Izadi, and A. Wilson. *Tabletops - Horizontal Interactive Displays*. Human-Computer Interaction Series. Springer London, 2010.

[28] T. Novotny, I. Lindt, and W. Broll. A multi modal table-top 3d modeling tool in augmented environments. In *Proc. of the 12th Eurographics Symposium on Virtual Environments*, pages 45–52. EG, 2006.

[29] J.-Y. Oh, W. Stuerzlinger, and J. Danahy. Sesame: towards better 3d conceptual design systems. In *Proceedings of the 6th conference on Designing Interactive systems*, DIS '06, pages 80–89, NY, USA, 2006.

[30] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge. Technical section: Sketch-based modeling: A survey. *Comput. Graph.*, 33:85–103, February 2009.

[31] H. Perkunder, J. H. Israel, and M. Alexa. Shape modeling with sketched feature lines in immersive 3d environments. In *Proceedings SBIM '10*, pages 127–134, Aire-la-Ville, Switzerland, 2010. EG.

[32] S. Schkolne, M. Pruett, and P. Schröder. Surface drawing: creating organic 3d shapes with the hand and tangible tools. In *Proceedings of CHI '01*, pages 261–268, New York, NY, USA, 2001. ACM.

[33] R. Wang, S. Paris, and J. Popović. 6d hands: markerless hand-tracking for computer aided design. In *Proceedings of UIST '11*, pages 549–558, New York, NY, USA, 2011. ACM.

[34] G. Wesche and H.-P. Seidel. Freedrawer: a free-form sketching system on the responsive workbench. In *Proceedings of VRST '01*, pages 167–174, New York, NY, USA, 2001. ACM.

[35] E. Wiese, J. H. Israel, A. Meyer, and S. Bongartz. Investigating the learnability of immersive free-hand sketching. In *Proceedings of SBIM '10*, pages 135–142, Aire-la-Ville, Switzerland, 2010. EG.

[36] A. Wilson and H. Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of UIST '10*, pages 273–282, NY, USA, 2010. ACM.

[37] M. Xin, E. Sharlin, and M. C. Sousa. Napkin sketch: handheld mixed reality 3d sketching. In *Proc. of VRST '08*, pages 223–226, 2008.