Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR

Bret Jackson, *Member, IEEE*, and Daniel F. Keefe, *Senior Member, IEEE*

Fig. 1. Lift-Off is an immersive 3D modeling system for artists using a bimanual 3D user interface in virtual reality (VR). After importing hand drawn paper and pencil sketches (a) into VR and placing these in space as virtual slides (white plane in b), 3D models are created using 3D curves (red curves in b) that the artist "lifts off" of the imagery or draws freehand in space. Finally, surfaces are swept along these curves to create virtual sculptures (c).

Abstract— Three-dimensional modeling has long been regarded as an ideal application for virtual reality (VR), but current VR-based 3D modeling tools suffer from two problems that limit creativity and applicability: (1) the lack of control for freehand modeling, and (2) the difficulty of starting from scratch. To address these challenges, we present *Lift-Off*, an immersive 3D interface for creating complex models with a controlled, handcrafted style. Artists start outside of VR with 2D sketches, which are then imported and positioned in VR. Then, using a VR interface built on top of image processing algorithms, 2D curves within the sketches are selected interactively and "lifted" into space to create a 3D scaffolding for the model. Finally, artists sweep surfaces along these curves to create 3D models. Evaluations are presented for both long-term users and for novices who each created a 3D sailboat model from the same starting sketch. Qualitative results are positive, with the visual style of the resulting models of animals and other organic subjects as well as architectural models matching what is possible with traditional fine art media. In addition, quantitative data from logging features built into the software are used to characterize typical tool use and suggest areas for further refinement of the interface.

Index Terms—Immersive 3D Modeling, Virtual Reality, 3D User Interfaces, Sketch-based Modeling

1 INTRODUCTION

Three-dimensional modeling is a fundamental task in computer graphics, and its applications range from product design to art. Yet generating detailed and realistic models remains time-consuming and difficult using conventional user interfaces [31]. While skilled artists are able to create complex models with fine control using current tools such as Maya or 3DS Max, they require extensive training, and many would argue that the complexity of the tools' interfaces does not effectively support human creativity. For example, we know from creativity support literature that sketching, exemplars, and physical action are all closely associated with increasing human creativity [37], but these techniques are rarely integrated with conventional 3D modeling interfaces. This disconnect is especially evident for free-form artistic or organic modeling (e.g., Fig. 1), as opposed to more engineering-oriented modeling.

In response, the research community has contributed a variety of new user interfaces that focus on 2D sketching as a way to incrementally build a 3D model (e.g., Teddy [14], Modeling-in-Context [26], EverybodyLovesSketch [2]). While these sketch-based interfaces closely align with creative art and design processes, creating 3D models from 2D sketches is a challenging (and typically under-constrained) computational and user interface problem. To avoid modeling errors from depth ambiguity, some 2D sketch-based systems restrict themselves to specific types of models (e.g., teddy bears and other forms based on inflating silhouettes, architectural models and

---

- *Bret Jackson is with Macalester College. E-mail: bjackson@macalester.edu.*
- *Daniel F. Keefe is with the University of Minnesota. E-mail: keefe@cs.umn.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx/

other forms based on mathematically defined geometric primitives).

3D modeling systems that use sketch-based input have also been developed for virtual reality (VR). The advantages of modeling in VR include enhanced spatial perception and the potential to avoid the depth ambiguity problem. Recent studies have shown that 3D interaction in an immersive environment offers a faster and more intuitive way to work in 3D compared with a 2D desktop environment [36], and a long line of research has demonstrated that VR can be beneficial for 3D modeling tasks in general (e.g., [30]). However, what we find particularly exciting about VR for 3D modeling is the potential to use 3D trackers and stereoscopic displays to sketch 3D shapes directly in space. This is a topic that has been explored in the VR community for at least two decades. Many interfaces have been developed, but some key challenges remain regarding the level of control that can be achieved with 3D sketch-based modeling tools and the creative workflows that these tools support.

In this paper, we present an immersive 3D modeling interface, called *Lift-Off*, that explores how 2D sketches and related imagery created or collected as a first step in artists' creative process can be tightly integrated into immersive sketch-based 3D modeling tools. We hypothesize this can benefit artists by providing a scaffolding that assists with the control challenges typically associated with sketch-based 3D modeling and by connecting the preparatory work artists do outside of VR with the modeling work they do inside VR. Artists begin with traditional sketching with paper and pencil or digital drawing tools. These sketches are imported directly into the 3D modeling environment and displayed as 3D slides placed in space inside a CAVE^{TM} environment (Fig. 1b). Artists then organize the modeling process around these slides, selecting curves (e.g., contour lines) to *lift off* of the image and place in 3D space and then sweeping surfaces along the 3D curves using a sketch-based interface.

The Lift-Off system, including the modeling style, is inspired by traditional art practice. In particular, we have studied the work of the Spanish sculptor, Pablo Gargallo [10], and others who fashion stunning physical sculptures out of sheet metal. The details in the form of the sculpture, choice of what surfaces to depict and what to leave intentionally vague, and the overall quality of the lines (e.g., contour edges) in these sculptures makes it clear that they are crafted by hand, with deliberation and control over the medium. 2D sketching, including 1-to-1 scale sketches of the shapes that will eventually be cut of metal, is critical to the overall design process for artists like Gargallo [10]. Gargallo's process and results, therefore, provide strong motivation for Lift-Off. We see creating virtual sculptures in this style as a great challenge for the research community, one where the results of the work can teach us about how to build VR tools that are expressive while also being controllable and that can support human creativity.

The key contributions of Lift-Off are:

- The design of a controllable VR sketch-based modeling tool.
- A bimanual 3D user interface technique that integrates with image processing algorithms for selecting important curves from 2D images displayed in 3D space.
- A bimanual 3D user interface for bending these curves along a projection surface to define a new 3D curve.
- A technique for interactive surface sweeping based upon one, two, three, or four neighboring curves and heuristics to pick the best sweep surface from 3D sketch-based input.
- A computer graphics style that is both precise and handcrafted, as inspired by traditional sheet metal sculpture.
- Qualitative and quantitative evaluations for long-term use and a controlled drawing experiment with novices.

2 RELATED WORK

2.1 Immersive Sketch-Based 3D Modeling

Building on Clark's early 3D modeling system for a head-mounted display [7], Butterworth et al. presented the full featured 3D modeling system, 3DM [5]. 3DM supported several geometric modeling features, including a freehand extrusion operation. This made it possible for modelers to create swept 3D surfaces from a polyline cross-section based entirely upon gestural six degree-of-freedom input by moving a tracked stylus through the air. Since this time, many immersive 3D modeling systems have embraced this sweeping, freehand, gestural (often full-body) "3D sketching" style of input as a primary means of creating 3D models. Key results in this research progression include Holosketch [9], FreeDrawer [40], Surface Drawing [35], CavePainting [22], BLUI [4], Drawing on Air [24], and Drag Drawing [25].

Several artists have also utilized tools in this style professionally, both in their own art practice [19, 21, 27, 34, 43] and within interdisciplinary scientific visualization teams [20, 23]. Feedback from all of these applications is consistent. There is something very powerful about this creative use of VR, and artists praise the immediacy and novelty of creating form via 3D sketching as well as standing inside their own 3D drawings. Artists appreciate the loose, rough, organic, and handcrafted style evident in these models as a welcome contrast to the smooth and straight, more machine-like style common to most computer-aided design tools [19, 43]. However, it is also clear from both user feedback and the result images in prior literature that this type of 3D input is challenging to control. 3D sketching in VR has, in general, been a good choice for creating quick, rough, gestural sketches, but has not been particularly successful when artists wish to create more controlled models of representational subjects.

Control is not a new problem; researchers have explored several alternatives to improve control while maintaining the immediacy of freehand 3D sketching. One option is to use haptics. User studies show that haptic feedback and variations of 3D sketching, such as bimanual 3D "tape drawing", significantly improve user control of 3D sketch-based input when tracing complex 3D curves [24, 25]. Unfortunately, the haptic-based solution is prohibitively expensive for most artists, and adding an active haptic device can also severely limit the working volume for 3D sketching as compared to VR environments, such as CAVEs^{TM} or HMDs, where users may stand up and move more freely.

Another option for addressing the control problem is to add more constraints to the geometric modeling operations. Several successful systems have been developed in this style. MakeVR [16], CaveCAD [13], and MockupBuilder [8] define 3D forms by building up simple geometric shapes. Other systems, such as ImmersiveFiberMesh [32] and SculptUp [33] rely on optimizations and volumetric techniques to inflate geometry. Either approach can result in a useful VR tool for certain applications, and interfaces for this type of geometrically constrained 3D modeling in VR continue to improve. (See, for example, a recent hybrid interface that combines direct spatial interactions for coarse input with 2D touch input on custom designed controllers for more precise interactions [30].) However, the key limitation of current immersive 3D modeling tools that address control through adding geometric constraints is that they also constrain the style of the resulting form, which tends to lack the handcrafted, organic quality that has attracted artists to freehand 3D sketching in VR.

Lift-Off addresses the issue of control through constraints, but rather than constraints based upon geometric primitives, the constraints originate from lines extracted automatically from artists' 2D sketches created outside of VR. Since these lines are originally hand-drawn, they have the desired style, and since artists are already skilled with traditional drawing media, they have no trouble making these lines controlled and accurate if they wish.

2.2 Modeling with 3D Curves

As 3D models created with Lift-Off are based upon 3D curves and surfaces swept between these curves, our work is related to other 3D modeling tools that use curves as a core modeling primitive (e.g., Just DrawIt [11], Masry et al. [28]). Introduced nearly 15 years ago, the FreeDrawer system [40] is still one of the most impressive *immersive* modeling tools in this style. From a technical standpoint, our work follows a similar approach: curves are created in 3D space, joined together to form a curve network, and then 3D surfaces are swept out following these curves. Lift-Off extends this fundamental strategy in two important ways. First, in addition to freehand sketching, Lift-Off supports a new method for creating organic-looking 3D curves in space by *lifting* curves off a hand drawn image and then bending them via a bimanual 3D user interface into an appropriate 3D shape. Second, Lift-Off's surface sweeping operation does not require a closed curve network or a process of selecting individual curves in the situation where a closed loop is difficult to identify. Instead, surfaces are swept out using one, two, three, or four surrounding curves for context, and the choice of which sweep surface to create is made automatically based upon the surrounding context and analysis of the user's sketch-based input. In this way, Lift-Off aims to operate as a "modeless" user interface in the style of recent 2D sketch-based user interfaces developed for tablet computers (e.g., Lineogrammer [42]). Multiple selection operations and toolbar widgets are avoided in favor of heuristics that select the appropriate operation to perform based upon the context provided by the current state of the drawing and characteristics of the user's sketch-based input. For example, many of the Lift-Off interactions are enabled based on proximity of the input stylus to existing geometry in the 3D model. Physical bendable curves have also been employed as a 3D user interface for surface modeling [3]; in contrast, Lift-Off uses a bimanual bending metaphor but supports a wider variety of curve shapes.

Lift-Off's 3D curve drawing operation is a two-step process, and in this way it is similar to the two-step approach presented by Grossman et al. [12]; however, there are some key differences. In Lift-Off, the first step does not include drawing but rather selecting a pre-drawn 2D curve from imported imagery. In addition, the fact that Lift-Off is designed to work in a stereoscopic VR environment has major implications for the user interface and the modeling subjects that can be addressed. For example, the second step in Lift-Off's 3D curve drawing interface (curve bending) takes advantage of the user's ability to walk around the 3D model so that it can be viewed from an appropriate angle and then uses six degree-of-freedom input from both hands to bend the curve into the desired shape. Although we have not formally evaluated this aspect, we also believe that the ability to work directly in a head-tracked stereoscopic environment where it is easy to view the model from any vantage point lends itself to 3D modeling with a handcrafted organic style that would be more difficult to achieve when drawing in multiple orthographic views as done in modeling tools for car or other mechanical design applications.

2.3 Supporting Creative Workflows

It is difficult for artists to create in a vacuum. When we create VR modeling applications that start with a blank canvas, we place artists in this position. Too many times in our lab, we have said to artists: *Please step into the CAVE^{TM} as I turn off the lights. Now, put on these glasses (which make it even darker). Now (as the artist looks into a blank 3D scene), do something creative.* Artists and designers work with sketchbooks and samples and photographs and more [6], but these design aids are typically left behind when they enter VR. Prior work has noted the benefits of the simple action of bringing scanned or photographed inspirational imagery into VR and either placing these digital images in 3D space as floating slides, displaying the imagery on one wall of a CAVE^{TM} as a reference, or viewing the sketch in a mixed reality environment with other design aids placed around the canvas [18, 23, 41]. Lift-Off takes this idea a step further, enabling artists to create 3D models directly from the preparatory sketches that serious artists already routinely create.

One way of viewing this approach is that it enables artists to create a 3D model *within the context* provided by an image. In this regard, our work is closely aligned and motivated by Lao et al.'s Modeling-in-Context system [26], which uses 2D pen strokes made on top of a photograph of a real-world scene to create 3D models that can be physically manufactured to fit into the scene using a laser cutter or 3D printer. Similarly, Tsang et al.'s [39] suggestive interface also uses 2D pen strokes on an image to create wireframe outlines, even going a step further to attract the user's input curves towards curves in the image. With only 2D sketching as the input, the resulting models of these systems must be relatively simple from a geometric standpoint; however, the interaction metaphor is powerful and one that we believe can be just as valuable in immersive environments.

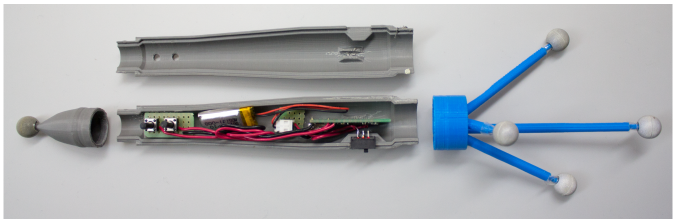


Fig. 2. Custom interaction stylus designed to be lightweight and to feel natural to artists who are used to holding a pen.

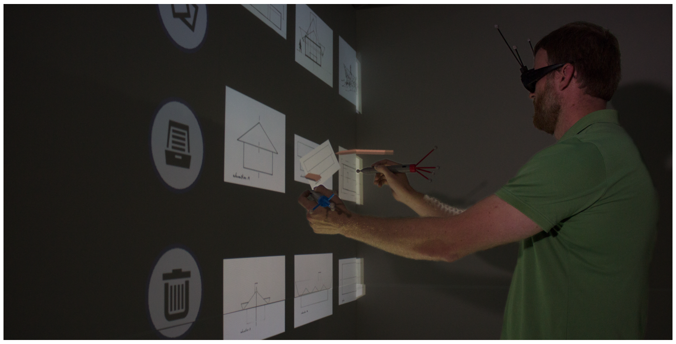


Fig. 3. An artist chooses a sketch to place in 3D space.

3 LIFT-OFF

Lift-Off is designed for use in a 4-wall CAVE^{TM} environment. The artist is head-tracked, and the graphics are rendered in quad-buffered stereo to provide a first person perspective. Artists hold a custom 3D printed stylus (Fig. 2) in each hand to support bimanual input. The primary drawing stylus is held in the dominant hand, and a secondary stylus is held in the non-dominant hand. Each stylus has two push buttons. A consistent pattern used throughout the interface is that the primary button on the drawing stylus creates or confirms a selection, while the secondary button deletes.

Artists create 3D forms using a series of six steps. First, artists use traditional design processes, such as sketching with pen and paper or a digital tablet, in order to explore a variety of visual ideas. The resulting set of hand-drawn 2D sketches (e.g., Fig. 1a) are digitized and imported into the VR environment as virtual slides. Second (Fig. 3), artists select a slide from a widget displayed on the left wall of the CAVE^{TM}. Third, artists place the slide in 3D space. Fourth (Fig. 4a-b), artists select a curve of interest from the imagery in the slide. Fifth (Fig. 4c), artists lift the curve into 3D space. We call the resulting 3D curve a rail. Note that rails are not just translated copies of the original planar curves. While lifting the curves into 3D space, artists use the styluses as handles, bending the curve to adjust its depth, which varies along the length of the curve, relative to the original slide. Finally, in the sixth step (Fig. 5), artists create surfaces, called sheets, by selecting one or more rails and performing a sweeping gesture. When this gesture could have multiple interpretations, the system takes the context provided by the rail system into account in order to infer the correct surface to sweep. Although these steps are described sequentially, the order of creating rails and sheets can be mixed. For example, an artist might start by placing a slide and creating several rails before sweeping a surface sheet.

3.1 Creating 3D Rails

Rails can be created in two ways. The first method is via freehand 3D sketching in the style of the series of VR-based sketching tools described in Section 2.1. Holding the primary button on the dominant hand's stylus will always begin a freehand drawing operation unless an existing rail, rail connection, or slide has been selected. There are many situations where this is useful; however, one of the key contributions of Lift-Off is addressing situations where there is a need to create

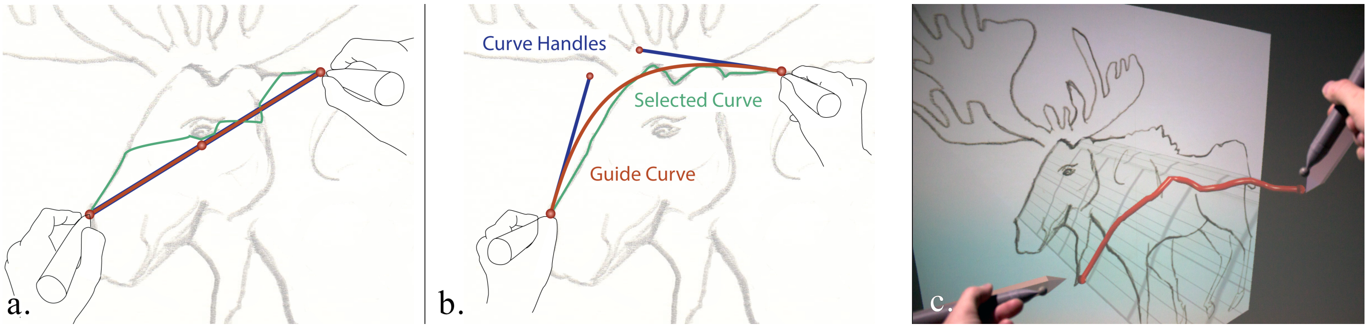


Fig. 4. (a) The first step of the curve lifting process is selecting an important curve from an image on a slide. When both hands are within the activation distance of a slide, a curve guide (shown in red) is projected on the slide. (b) Rotating the styluses moves the curve handles changing the shape of the curve guide. The selected curve (shown in green) is influenced by the guide but is constrained to follow curve features identified automatically in the image. (c) In the second step the selected curve is lifted off the slide and placed in space. Again, rotation of the styluses changes the shape of the curve, but this time the curve is bent to adjust depth.

3D curves with more control than what can typically be accomplished via freehand input. This leads to the second method for creating rails, which is to lift them off of slides placed in space. This interface requires two steps.

3.1.1 Selecting Important Curves from Slides in 3D Space

When a Lift-Off modeling session starts, slide images are automatically loaded from a special folder into which the artist places scanned or photographed images. One slide is created for each image in the directory, and these are displayed in VR along one side of the CAVE$^{\text{TM}}$ in a slide selection grid (Fig. 3). To place a slide in space, the artist first selects the slide from the grid by reaching for it with both styluses, similar to how one might lift a picture off of a wall. When both styluses are close enough to the slide a copy is made, and the slide copy animates to look like it is grasped between the hands. A button click on the primary drawing stylus confirms the selection. The artist then positions the slide, moving the styluses further apart or closer together to scale the image and rotating the styluses to adjust the orientation. When the correct placement is achieved, a second button click on the primary drawing stylus locks the slide in place. Once a slide is placed it cannot be moved because rails that had been lifted out of the slide would no longer project back to the original curves in the image.

To select an important curve (e.g., a contour line, another key feature in the drawing), the artist moves both hands toward a slide that was already placed in space. When both styluses are within the activation distance of the slide (5 cm in our implementation), the slide becomes active, and the user interface displayed in Fig. 4 is displayed. The interface includes a guide curve that is a cubic Bézier curve drawn between the two styluses. Initially this curve is a straight line, but as the styluses are rotated, the guide curve bends in response as shown in Fig. 4. The guide curve does not depend upon the underlying image data and is meant solely as a means of visual feedback for the interface. It can be thought of as a magnetic rope controlled by the user. Using the guide curve, the user pulls the selected curve, which is also visualized on the slide, in a desired direction, but rather than following the guide exactly, the selected curve settles onto the closest curve identified in the underlying pixel data of the image. When the desired curve is displayed, a click of the primary button on the drawing stylus confirms the selection.

There are several nuances to this design and implementation details that we established through iterative testing. A Bézier curve was chosen to serve as the guide because it limits the degrees-of-freedom that the artist must manipulate to select a curve in the image. To interactively manipulate the guide curve, each stylus is made to control two of the four Bézier curve control points. The control points at the beginning and end of the curve as set by the positions of the styluses are projected onto the plane of the slide. The two interior control points are calculated based upon the orientation of the styluses. Initially when the artist places the stylus close to the slide, the inner points are created by translating half the distance towards the opposite stylus, thus preventing loops in the curve (which would be caused by the inner control points extending past each other), and projecting these locations onto the surface of the slide. By limiting the degrees-of-freedom (the translation distance for interior control points cannot be changed) the user interface is simplified while still providing enough freedom to choose desired curve shapes because the selection curve settles onto the underlying image data. When the slide is first activated, the initial orientation of each stylus is recorded. From this point on any relative pivoting of the styluses will map a corresponding rotation to the inner Bézier control points. If a rotation would cause the artist to move into an uncomfortable position, then she may simply pull her hands away from the slide, leaving the activation area, rotate into a more comfortable position, and then reactivate the interface, enabling a wider range of motion.

Many algorithms (e.g., active contour snakes [17]) can be used to morph the spline to the closest contour in the imagery. Our approach is similar to the algorithm described by Jackson et al. [15]. First, the 2D slide image is thresholded to identify drawn lines and a distance map is calculated. Then, 3D points are interpolated along the Bézier guide curve and converted into u and v parametric coordinates in the 2D slide image. These 2D points are iteratively refined by gradient ascent, using the gradient derivatives of the image's distance map. When a point converges, the outcome is that it lies on the centerline of the drawn line. These final (u,v) positions are converted back to their 3D locations to obtain the final curve of interest in the plane of the slide.

3.1.2 Bending Curves to Form Rails

Once a curve is selected on the slide, the next step is to bend the curve into a 3D rail. As soon as the curve selection is confirmed by clicking the primary button on the drawing stylus, a 3D guide surface is created, as shown in Fig. 4c. This surface follows the shape of the curve and extends perpendicularly outward from the slide in both directions along the slide normal. The artist's job is to now use the bimanual interface to bend the curve along the guide surface, adjusting the depth of the curve (relative to the plane of the slide) along its length.

Here, the styluses are used in a similar manner, but there is a mapping from the 3D space of the guide surface to a (u,v) 2D parametric space defined on the surface. The 3D coordinates of the styluses are used to calculate a 2D Bézier curve in the (u,v) space of the guide surface. The control points at the beginning and end of the curve are set to the point along each of the two edges of the guide surface that are closest to the styluses, and the interior control points are set again based upon the orientation of the styluses. Rotating and translating the styluses, therefore, results in a change in the 2D Bézier curve, which is projected onto the guide surface using the same (u,v) coordinates that would be utilized during a texture mapping application. The result is a 3D curve that follows the guide surface, which we resample using a 3D Catmull-Rom spline, and render as a rail. We developed this approach

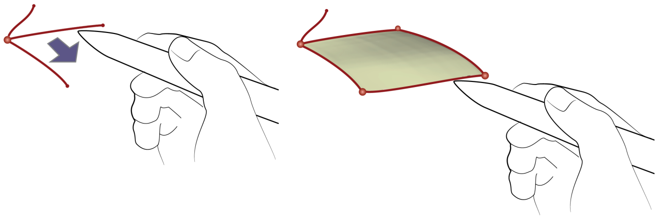
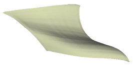
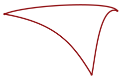
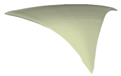
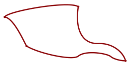


Fig. 5. Selecting a rail and pulling in the direction of a connecting raill will sweep a sheet surface.

Table 1. Possible surface sweeps depend on the number of connecting rails.

Sweep Type	Rail Connections	Resulting Surface
One Rail		
Two Rails		
Three Rails		
		
Four Rails		

through iterative design. In particular, when working with concave guide surfaces, defining the bending operation within the (u,v) parametric space of the surface rather than in 3D space ensures that the resulting rail does not contain a discontinuity or jaggy segment.

3.2 Connecting Rails

To use rails as a guide when creating surfaces the rails must be connected. Snapping is used to form these connections. When creating a rail via freehand sketching, the virtual representation of the drawing stylus will snap to an existing rail endpoint if the tip is within a close radius. Our implementation uses a snap radius of 2.54 cm. Starting to draw a rail from a snapped endpoint will connect the rails at the starting point. Similarly, an artist can connect the end of a rail he or she is currently drawing by releasing the drawing button within the snap radius of an endpoint. While the drawing button is held down, the virtual cursor does not snap to avoid inadvertently changing the drawn curve, but endpoints within the snap radius are highlighted in yellow to indicate that a possible connection will be created if the button is released.

When creating a rail from a 3D slide, possible connections are highlighted directly on the slide. While the interface for selecting important curves from the slide is active, the endpoints of all existing rails are projected onto the slide surface where they are displayed for the user. If the cursors move within the snap radius of a point, then this point is highlighted and a dashed line is displayed connecting the highlighted point on the slide to the corresponding rail endpoint in 3D space. If the selection of the important 2D curve is confirmed by clicking the primary button on the drawing stylus while the highlight is displayed, then a connection to the existing rail is made. If desired, connections at both endpoints of a new rail can be made using this interface.

3.3 Sweeping Surface Sheets

Surface meshes are created by sweeping a profile rail along guide rails, in an approach that is similar to the Birail operation in Maya [1] or the Sweep2 operation in Rhino [29]. However, unlike these conventional modeling tools, which require the artist to carefully specify each curve, our approach automatically determines the rails that are involved in the modeling operation. To create a surface, the artist selects a rail and pulls it in a direction. The resulting surface is based on two contextual factors: (1) the direction the artist initially pulls, and (2) the number and direction of the rails connected to the selected rail.

A taxonomy of rail-based free-form 3D modeling is presented in Table 1. In Lift-Off, rails are represented in an undirected graph data structure, where the nodes represent rail endpoints and the edges represent connections between rails. This allows us to categorize modeling operations based on the topology of the graph. The taxonomy uses this information, defining surfaces created using one, two, three, or four connecting guide rails.

The direction the artist sweeps the initial profile rail is used to determine which connecting rails serve as guides for the surface creation. The sweep direction is compared against the "direction" of each rail that connects to the initial profile rail. Calculating a rail's direction is trivial if it is straight, as the average direction of the individual line segments that form the rail can be used. However, this approach does not work as well when the rail is curved. Consider the case where the user draws a semi-circular rail: we found that averaging the segment directions for the first fifth of each rail worked well as a compromise between avoiding noise from averaging a low number of line segments and avoiding errors from highly curved rails.

Once the directions for each rail are calculated, they are compared with the direction the artist initially pulled. If the angle between the two vectors is within a threshold (60 degrees in our implementation), we assume that the artist meant to sweep along the rail. If there are multiple connecting rails at a single endpoint of the selected rail, only the one with the orientation most similar to the initially pulled direction is considered for determining the sweep case.

This approach for determining guide rails is works particularly well for selecting a single guide rail; however, selecting multiple guide rails (Three Rail and Four Rail cases in Table 1) is still difficult when the rails have different directions. In this situation, the user must sweep in the average direction of both rails. In future work, we intend to explore alternatives such as being able to change the guide rails after the initial sweep by dragging the stylus closer to a different connected rail.

Once the connections and sweep type are determined, a surface mesh is calculated. Throughout the rest of this section we will refer to the specific rails involved in the sweep operation using the following terminology: the initially selected profile rail is called the beginning rail; the rails that the profile is swept along, creating the sides of the surface, are called the guide rails; the profile at the end of the sweep is called the end rail.

The algorithm proceeds using the following steps:

1. Identify or create the rails that define the surface boundaries
2. Calculate profiles between beginning and end rails.
3. Align the profiles with points along the guide rails.
4. Create vertices at each interpolated point, offset along the positive and negative normal directions to add surface thickness.
5. Triangulate between the vertices to create a mesh.

First, the rails that form the boundaries of the surface must be identified or created. (As shown in Table 1, the one, two, and three rail sweep cases may require missing rails to be created by duplicating and translating existing ones.) The interior surface profiles are then calculated by interpolating between the beginning and end rails. The first step is to resample the beginning and the end rail so that they have the same number of points. Then we align and scale the end rail so that its endpoints match the endpoints of the beginning rail, putting

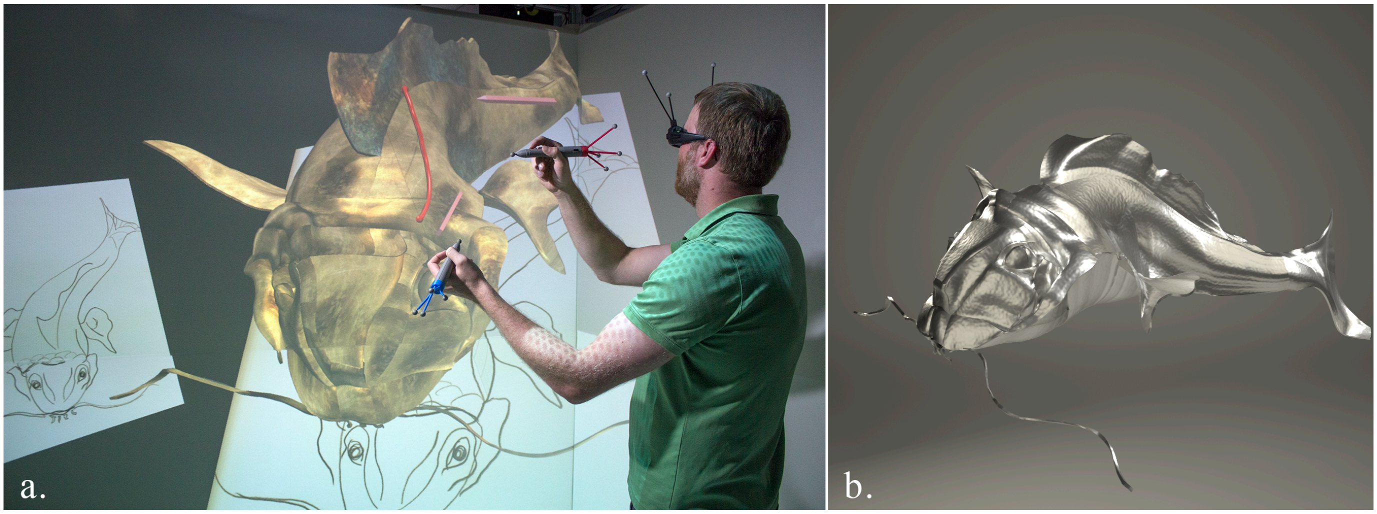


Fig. 6. (a) An artist lifts a rail off of a pencil sketch of a fish. (b) The resulting model (rendered in Blender).

both rails in the same coordinate space. Next, we resample the guide rails so that they have a corresponding number of points. Finally, we loop through the resampled guide rail points. For each pair of guide rail points, we calculate an interpolated profile rail in the combined beginning/end rail space and then transform this to align with the guide rail points.

Surface normals are calculated from adjacent profiles. The profile points are duplicated and offset along the positive and negative normal vectors to create vertices, adding thickness to the mesh. Finally, a triangulated mesh is created by joining adjacent vertices along the top and bottom surfaces. The two sides are connected along the thin edge by additional triangle strips linking the vertices that were offset along the normal directions.

A detail of our implementation is that a complete sweep surface is calculated as soon as the sweep operation is recognized. This enables the artist to interactively adjust the amount of the surface to sweep based upon movement of the stylus. The distance the artist has pulled the stylus from his or her initial click point is mapped along one of the guide rails. Although the surface is calculated for the full length, only the surface vertices that connect to the sides before this point are displayed as a mesh to the artist.

3.4 Combining, Dividing, and Deleting Rails

Placing the drawing stylus near a rail or surface highlights the object, then a press of the secondary button on the primary drawing stylus deletes it. This can be useful for deleting rails or surfaces that were placed incorrectly. More interesting operations can be accomplished by adding or deleting connections between rails. To divide a rail in two, the artist moves the drawing cursor to a desired location along the rail and clicks the primary drawing button to create a new connection point. The original rail is divided and replaced by two rails that join at the new connection. Alternatively, to merge two rails into a single rail, the artist snaps the drawing cursor to an existing connection and clicks the secondary button to delete it, merging the two original rails into a single continuous rail. Since the surfaces generated during sweeps depend upon the specific rails chosen (even in a closed loop, selecting a different rail for the beginning rail will result in a different surface), it is useful for modelers to work at this level, experimenting with different sweep options.

3.5 Reorienting, Scaling, and Rendering

Reorienting the artwork is accomplished using the stylus held in the non-dominant hand. Clutching the primary button grabs the virtual model, which is then reoriented via a one-to-one mapping with the translation and rotation of the stylus. If the primary button on the drawing stylus is pressed and held during a reorientation operation, then a scaling mode is activated with the scale of the model changing interactively in proportion to the changes in distance between the two hands. The interface currently allows the user to choose between three different surface materials using the keyboard, and the surface thickness can be set in a configuration file before starting to model.

Fig. 7. Lion sculpture created with Lift-Off and rendered in Blender. Inset shows the original sketch drawn on a Wacom tablet.

4 RESULTS AND DISCUSSION

In this section we report on several types of results. First, we describe our own longterm use of Lift-Off. (The second author of the paper has exhibited his artwork internationally, and the first author has experience with metal sculpture and fine woodworking using traditional physical materials.) Next, we present results from similar longterm tool use, but by an architect collaborator who used Lift-Off for a current professional project. Finally, to better understand the learning curve and strategies that a variety of new users might adopt, we present results from more controlled but short-term use of the tool by novices. The results include both qualitative insights, such as observations, lessons learned, and refinements made during iterative development, as well as quantitative data recorded from logging features that we designed into the tool.

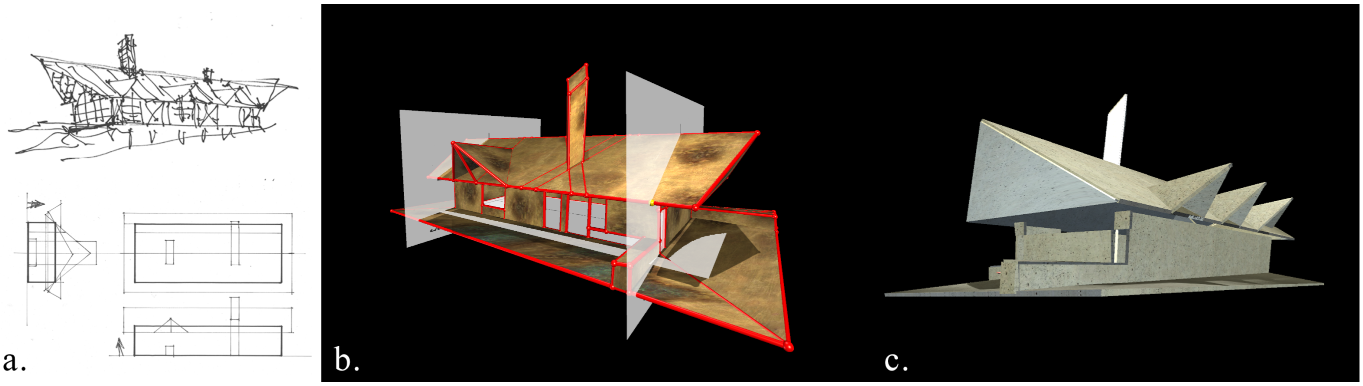


Fig. 8. Architectural design of a cabin created with Lift-Off. (a) Perspective sketch and orthographic drawings used during design. (b) Slide placement, rails and surfaces. Note the additional construction lines used to position the chimney and dormer on the roof. (c) Final cabin model.

4.1 Application 1: Artistic Modeling in the Style of Virtual Sheet Metal

To test Lift-Off we created several 3D models representing different subjects. For each model, the design process started with traditional 2D sketching, either with pencil and paper or on a Wacom tablet.

Figs. 1 and 6 were created by the second author based upon a series of his pencil sketches of animals. The moose sculpture in Fig. 1 is, perhaps, the best example to date of a match with the aesthetics of traditional sheet metal sculptures, which is a goal we set for the interface, since we believe it represents a significant technical challenge for freehand 3D modeling in VR. Each surface has some thickness, but the collection of surfaces does not form a watertight volume as in a traditional 3D computer graphics character model. Rather, the artist has made a conscious choice about when to create a more volumetric effect (as in the legs, which have a rounded cross-section) and when to use a flat sheet with controlled profiles that suggest but do not fully specify the form (as in the face and antlers). From a technical standpoint, this model was created using two separate sketches. The body comes from the first sketch (Fig. 1a). The antlers comes from a second detail sketch of several possible antler profiles. Two slides of the antler sketch were placed into the scene at slightly different angles so that the forms of the left and right antlers could be lifted off at the desired angle for each side. One refinement made to the interface in response to this experience was to the rail selection algorithm. There were several instances in this model where a surface sweep operation can have multiple interpretations. In other words, different rails will be chosen and a different surface will result depending upon the direction of the user's gesture. In this situation, the algorithm acts as an expert system, and it is frustrating if this expert is wrong, a situation that arose when there were rails with high curvature. In response, we modified the sweep surface selection algorithm described in section 3.3 from using an original value of the first one third of the rail segments to determine the rail direction to only using the first fifth. This improved performance, but we believe an even better refinement would be to interactively switch from one guide rail to another during the sweep operation by pulling in a different direction after the extrusion has started.

Fig. 6 was created from the pencil sketch seen in the screenshot. Here, the surfaces are arranged in space so as to create more of a full volume. When viewed in stereo, the curving forms of the fins are one of the most compelling visual aspects of the work. The majority of the surfaces were created by lifting rails off of the sketch. Some of the rails used to define the eyes and the whiskers were created by freehand drawing. One surprising observation from the workflow was that there was an ergonomic preference for orienting the slide horizontally, as if it were a virtual table rather than a canvas on an easel. The table orientation seemed to better facilitate walking around the model and lifting curves into space.

In a similar style, Fig. 7 is a rendering of a lion mask created by the first author in about an hour and a half from the 2D digital sketch shown in the inset.

The total modeling time as well as other quantitative data for all the modeling results are included in Table 2 (excluding the lion which predates the logging features).

4.2 Application 2: Architectural Design

In a second longterm application of Lift-Off, we invited our collaborator, Bruce Cornwall, to use Lift-Off. Bruce is an architect with over 30 years of experience and is one of the lead designers and the director of campus planning for a large regional architecture firm. He decided to use Lift-Off to experiment with a cabin he is currently designing. All in all, he used Lift-Off in our lab for six 2–3 hour sessions that included introduction to the tool, skill building in its use, and exploration of multiple design options for the cabin. Fig. 8c shows his final model.

4.2.1 Cabin Model and Feedback on Specific Features

The 3D cabin model was created using a variety of the spatial modeling techniques described previously. First, the sketched plan view was placed parallel to the floor of the CAVE^{TM} to use as a base. Then, he placed elevation drawings orthogonal to the plan on two sides. He created the building's walls and ridgeline by lifting contours out of either the plan or elevation drawings. Then he lifted contours from the side elevations to create the angled edges of the roof gables.

Early in the modeling process, he formed two sides of the cabin by selecting the same 2D contour and lifting it off of the slide by different distances. Because the endpoints of the resulting rails projected to the same points on the slide, there was no easy way to connect them. Thus, we added an additional drawing mode (activated via a floating menu on the side wall of the CAVE^{TM}) that allows the user to create a straight rail connecting the two styluses when the drawing button is pressed. If the button is held, the line can be adjusted interactively, snapping to existing rail endpoints as the user moves the styluses and completing the connection when the button is released.

Although this feature was designed to connect rails, our collaborator used it in an unexpected way to create new rails that were not in the original drawings with more control than directly drawing freehand. In one modeling session, he spent nearly the entire time adjusting the placement of the dormers in the roof, their proportions, and how far they extend from the roof. This provides further evidence of the importance of supporting the ability to rapidly create and explore design options. He said, "If you can do something quickly and see how it reads as a form, you can start to play around with different models and go back and forth between different mediums, which I think would be really fascinating". This also illustrates the technique he used of placing connecting rails and splitting them in order to build construction lines (e.g., the four rails creating a cross used for positioning the chimney in Fig. 8b).

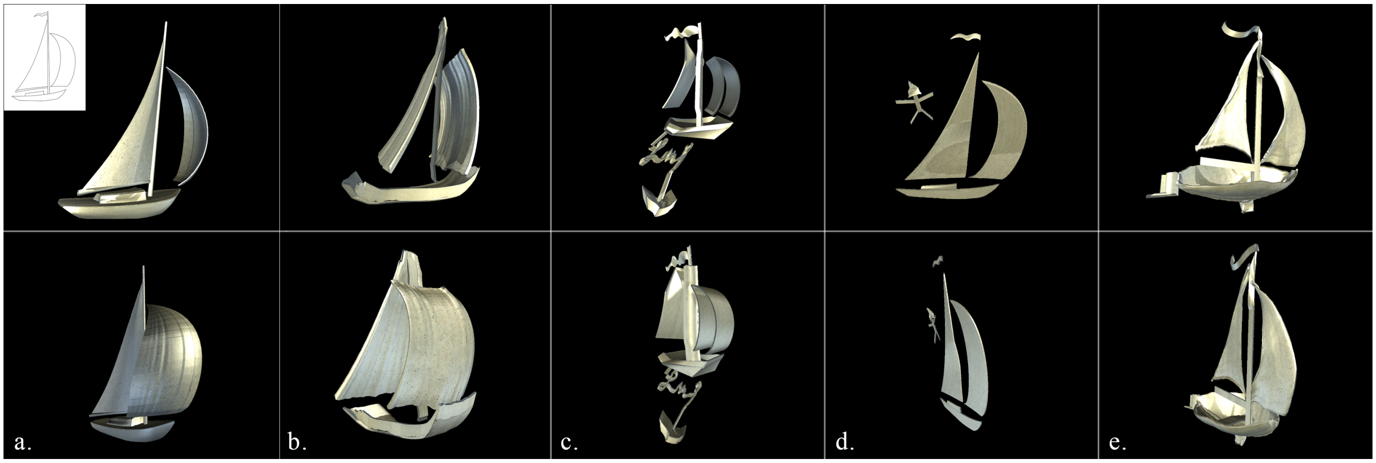


Fig. 9. Sailboat models created during user evaluation. (a) author's model and original sketch. (b) participant 1. (c) participant 2. (d) participant 3. (e) participant 4

His experience also resulted in additional suggested changes, specifically to the placement of the slides. He had difficulty aligning the plan and elevation slides with each other so that the drawn lines aligned and were perfectly orthogonal. When placing a slide, the system draws the projections of all the current rail endpoints on the slide to help with alignment; however, he requested a feature that would snap the slides to orthogonal angles and similar scales. Additionally, the ability to adjust the slide's position along its normal would avoid occlusion issues and enable an artist to place it closer to where it is needed for a specific rail.

4.2.2 VR Sketching as a Paradigm Shift

Several themes emerged from our observations and discussions during this collaboration. There is a problem with current computer-based design tools. Our collaborator states, "the problem with existing tools is that you have to know exactly what you want to draw before you can create". In contrast, he says of Lift-Off, "What is exciting is the potential to use this as a pure design tool that enables one to create rapid 3-D sketches that capture the spontaneity and serendipitous discoveries that are unlocked through pure sketching, but to achieve this in 'real 3-D' space! Wow." As we interpret this feedback, we must be mindful of the history of prior work in freeform modeling in VR. In this real application to architecture, what seems to have generated the most positive response is actually the exciting potential of 3D sketching in VR in general. The key idea behind Lift-Off is 3D sketching relative to reference imagery, and as we probed deeper into the feedback to better understand the impact of this, the theme that emerged was scale.

4.2.3 The Importance of Scale

We learned that, for architecture, the relative massing and the way that a shape or space reads to the viewer depends entirely on the scale, and it is critical to know this while designing. In fact, correctly indicating scale may be even more important than increasing control for new 3D sketching interfaces, particularly during the early stages of design where rough input is acceptable. Sketching relative to reference slides is one way to address this, but others might also be useful. When sketching in 2D, our collaborator frequently draws a stick figure to indicate the relative size of a building. In Lift-Off, he was able to sketch a similar figure by drawing it with freehand rails. He also suggested that loading in an avatar and being able to easily modify its size "when you realize a larger/smaller scale is more exciting" would be ideal. We have also found that viewing architectural models at life-size scale can be beneficial for evaluation. Lift-Off enables the user to scale to arbitrary sizes; however, an additional feature that quickly toggles between life-size scale and a smaller working scale would be beneficial.

While 3D sketching relative to slides was viewed as likely to be useful for evaluating scale, there was some skepticism about the other hypothesized main benefit of Lift-Off, the control afforded by lifting curves off of the slide imagery. It was clear to our collaborator that this can provide control and that this can be useful for users who are accustomed to 2D sketching to create 3D models. However, he was so taken by the spontaneity and freedom of 3D sketching that it was also clear that control should not come at the cost of these key benefits. We did not try to create a similar cabin using only freehand sketching to compare the results. However, our interpretation after this study is that for architectural models where there are many straight lines and geometric relationships, we are right at the cusp where the features Lift-Off provides beyond freehand 3D sketching might sometimes be useful but might other times slow down the modeling process. In contrast, for the complex, more organic modeling subjects presented earlier, we believe these would be extremely difficult to create in VR without the new tools provided by Lift-Off. We plan to continue evolving the interface for use in both styles of application, and our collaborator plans to start designing a city block scale development project in Lift-Off in the coming weeks.

4.3 Case Study with Novice Users

While the previous two case studies provide modeling results and qualitative feedback from extended use, we are also interested in feedback and quantitative data on how the individual features would be used by novices.

Participants. Three university students and a professional architectural designer participated in the study and were compensated for their time. Two of the participants were female. All reported limited prior use of virtual reality systems. Experience with 3D modeling software varied from 5–20 prior uses (2 participants) to more than 20 prior uses (2 participants). Two participants also reported at least occasional video or computer game use.

Task. Participants were asked to model a sailboat using the 2D sketch shown in Fig. 9. Participants placed the sketch in the VR environment, after which they were told to take as long as they needed to model a sailboat using any of the modeling techniques previously described.

Training. To start participants thinking creatively about the 3D form of a sailboat, each participant was asked to browse Google image results for the search term "sailboat" for two minutes. Following this introduction, the participant was given a tutorial of the program's features while performing each action. Individual actions were repeated until the participant was able to successfully complete them. Finally, the participant was asked to start modeling the base of the sailboat. The participant was encouraged to ask questions and was given reminders about how to use specific features. After demonstrating profi-

Table 2. Summary of logged data from a user evaluation modeling a sailboat. Values indicate the number of occurrences.

Model	Time (min)	Freehand Rails	Lifted Rails	Split Rails	Joined Rails	1-Rail Sweeps	2-Rail Sweeps	3-Rail Sweeps	Rail Deletes	Surface Deletes	Moves	Scales
Moose	113	88	104	41	35	8	12	73	136	33	214	36
Fish	154	106	59	90	72	9	35	90	181	64	565	102
Cabin	385	353	51	98	47	11	17	50	201	27	319	17
Boat P. 1	32	19	24	7	5	7	1	5	40	4	197	9
Boat P. 2	54	69	24	52	35	31	37	27	175	53	483	51
Boat P. 3	28	16	63	23	14	5	5	8	58	7	92	23
Boat P. 4	21	88	1	51	54	18	6	22	115	27	101	16
Boat Expert	32	8	57	7	15	2	1	23	34	9	186	24

ciency with the tool by creating several surfaces for the sailboat base, the program was restarted and the participant began the modeling task. Training times varied between 10 and 25 minutes for each participant. Overall, each participant spent about one hour in the VR environment.

4.3.1 Results

Models created during the sessions are shown in Fig. 9. The software automatically logged system events and created a summary report for each participant (shown in Table 2). The first author's results for the task are also included as a point of comparison for an expert user of the system (Fig. 9a).

4.3.2 Discussion

To understand the use of the tool, one aspect is the ease with which a novice can combine freehand drawing in space with curve selection from a slide. As shown in Table 2, all participants used a combination of both techniques.

Participant 4 used almost all freehand drawing (88 freehand rails, 1 lifted rail), describing herself as impatient. Even though the lifted rails created smoother curves, she felt that the two step process (indicating a 2D curve, then lifting it into space) was not as immediate as directly drawing. Her comment is also supported by the data – she was able to create her model more quickly (21 minutes) than the other participants. There is clearly a trade-off between the immediacy of freehand drawing and the control provided by the lifting approach.

Participant 2 also reflected on this trade-off, but expressed the need for both rail creation approaches. Like Participant 4, he also primarily used freehand rails (69 freehand rails, 24 lifted rails), although this can be explained in part by the additional features of the anchor and signature (Fig. 9c). He reported that he liked the "straight line control [of the lifted approach] but also the ability to express yourself [with freehand]".

Unsurprisingly, the two participants that used the most freehand rails also had the most rail deletions (175 for participant 2 and 115 for participant 4). We speculate that this is caused in part because the difficulty of freehand drawing caused more errors (i.e., rails that did not have the shape that the user wanted); however, we also saw that the immediacy of freehand drawing enabled more exploration of form. One user who did not participate in the study expressed the importance of easily deleting rails, saying "I like that I can make stuff and then make it go away". This supports previous research (e.g., [37]) that has found the ability to undo to be particularly important for encouraging creative exploration.

Some participants did not make use of the 3D space in the way we had imagined. In these cases, participants lifted contours out of the slide, but they kept them parallel to the slide surface, which resulted in a simple translation of the curve on the slide to a nearly identical curve a short distance away. This can be seen in the sails in the models by participants 1 and 2 in Fig. 9. Participant 3 used this strategy nearly exclusively, creating a completely planar sailboat. This demonstrates control, but not the type of control that we had envisioning, and we wonder if in some way the reference of a 2D slide caused these participants to think in a more two dimensional style that they would have without this reference.

In terms of workflow, most participants built a wireframe of rails to define the form before filling in surfaces. This can be seen by the greater number of two and three rail sweeps (with the exception of Participant 1), and also the greater number of rail deletions compared to surface deletions. From observing the modeling process, surfaces were most often deleted when they occluded parts of the slide. After the participants finished working in the occluded region they would then recreate the deleted surface. This indicates that some form of 'x-ray' lens feature might be useful.

5 LIMITATIONS AND FUTURE WORK

One limitation of the current implementation is that although it allows an artist to modify or delete rails and surfaces, and place new slides at any point while creating a model, it does not allow for editing the existing sketches that have already been placed as slides. In future work, we plan to complete this design loop, by responsively enabling artists to iterate in both 2D and 3D by maintaining a coupling between the original sketch and the resulting geometry. This might be implemented using a hybrid interface that integrates a portable tablet within the CAVE^{TM} environment. We would also like to support perspective projections in addition to current lift modes which assume an orthographic projection.

Like most spatial interfaces there is also the potential for user fatigue. We have tried to maximize comfort by using custom lightweight styluses and by minimizing the energy required for individual modeling actions so that artists may rest their arms between creating rails or sweeping sheets. Although the spatial interface presented here is more physically tiring than conventional modeling interfaces, this should be weighed against the advantages of a spatial interface. Full body movement allows users to better capitalize on proprioception; fluid body movement has been linked with enhanced creativity [38].

6 CONCLUSION

In conclusion, we have presented an immersive 3D modeling interface that enables artists to create complex 3D models with fine-control in VR. We attribute this ability to the strategic use of 2D sketches to constrain the degrees-of-freedom when creating 3D rails. This integration of 2D and 3D techniques allows artists to leverage the sketches they have created in the first step of the design process to build a 3D model using surface sheets and rails.

ACKNOWLEDGMENTS

Thanks to Bruce Cornwall of LHB, inc. for valuable feedback and significant time spend designing with Lift-Off. This work was supported in part by the National Science Foundation (IIS-1218058) and the University of Minnesota Doctoral Dissertation Fellowship. It utilizes the

VRPN library maintained by UNC-Chapel Hill's CISMM project with support from NIH/NCRR and NIH/NIBIB award 2P41EB002025.

REFERENCES

[1] Autodesk. Maya. `http://www.autodesk.com/products/autodesk-maya/`, 2015.

[2] S.-H. Bae, R. Balakrishnan, and K. Singh. EverybodyLovesSketch: 3D sketching for a broader audience. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, pages 59–68, New York, NY, USA, 2009. ACM.

[3] R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and K. Singh. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In *Proceedings of the 1999 Symp. on Interactive 3D graphics*, pages 111–118, New York, NY, USA, 1999. ACM.

[4] A. W. Brody and C. Hartman. BLUI: a body language user interface for 3D gestural drawing, 1999.

[5] J. Butterworth, A. Davidson, S. Hench, and M. T. Olano. 3DM: A three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, pages 135–138, New York, NY, USA, 1992. ACM.

[6] B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.

[7] J. H. Clark. Designing surfaces in 3-d. *Communications of the ACM*, 19(8):454–460, Aug. 1976.

[8] B. R. De AraúJo, G. Casiez, J. A. Jorge, and M. Hachet. Mockup builder: 3d modeling on and above the surface. *Comput. Graph.*, 37(3):165–178, May 2013.

[9] M. F. Deering. Holosketch: A virtual reality sketching/animation tool. *ACM Trans. on Computer-Human Interaction*, 2(3):220–238, 1995.

[10] R. O. Fernández. *Catálogo Museo Pablo Gargallo*. Zaragoza: Ayuntamiento de Zaragoza, 2004.

[11] C. Grimm and P. Joshi. Just DrawIt: A 3D sketching system. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pages 121–130, Aire-la-Ville, Switzerland, 2012. Eurographics Association.

[12] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton. Creating principal 3D curves with digital tape drawing. In *Proc. of the Conference on Human Factors in Computing Systems*, pages 121–128, New York, NY, USA, 2002. ACM.

[13] C. E. Hughes, L. Zhang, J. P. Schulze, E. Edelstein, and E. Macagno. CaveCAD: Architectural design in the CAVE. In *IEEE Symposium on 3D User Interfaces (3DUI)*, pages 193–194, 2013.

[14] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3D freeform design. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 409–416, New York, NY, USA, 1999. ACM.

[15] B. Jackson, T. Y. Lau, D. Schroeder, K. C. T. Jr., and D. F. Keefe. A lightweight tangible 3D interface for interactive visualization of thin fiber structures. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2802–2809, December 2013.

[16] J. Jerald, P. Mlyniec, A. Yoganandan, A. Rubin, D. Paullus, and S. Solotko. MakeVR: A 3D world-building interface. In *IEEE Symposium on 3D User Interfaces (3DUI)*, pages 197–198, 2013.

[17] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Intl. Journal of Computer Vision*, 1(4):321–331, 1988.

[18] D. F. Keefe. The making of La Guitarrista Gitana. In *SIGGRAPH 2002 Art Gallery Catalogue*, 2002.

[19] D. F. Keefe. From gesture to form: The evolution of expressive freehand spatial interfaces. *Leonardo*, 44(5):460–461, 2011.

[20] D. F. Keefe, D. Acevedo, J. Miles, F. Drury, S. M. Swartz, and D. H. Laidlaw. Scientific sketching for collaborative VR visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):835–847, 2008.

[21] D. F. Keefe and B. Campbell. Visweek art show catalog 2011. IEEE VisWeek 2011, October 23–28, Providence, RI, 2011.

[22] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, and J. J. LaViola Jr. CavePainting: A fully immersive 3D artistic medium and interactive experience. In *Proceedings of I3D 2001*, pages 85–93, 2001.

[23] D. F. Keefe, D. B. Karelitz, E. L. Vote, and D. H. Laidlaw. Artistic collaboration in designing VR visualizations. *IEEE Computer Graphics and Applications*, 25(2):18–23, 2005.

[24] D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Drawing on air: Input techniques for controlled 3D line illustration. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1067–1081, Sept 2007.

[25] D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Tech-note: Dynamic dragging for input of 3D trajectories. In *Proceedings of IEEE Symposium on 3D User Interfaces 2008*, pages 51–54, 2008.

[26] M. Lau, G. Saul, J. Mitani, and T. Igarashi. Modeling-in-context: user design of complementary objects with a single photo. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, pages 17–24. Eurographics Association, 2010.

[27] W. Mäkelä, M. Reunanen, and T. Takala. Possibilities and limitations of immersive free-hand expression: A case study with professional artists. In *Proceedings of the 12th Annual ACM Intl. Conference on Multimedia*, pages 504–507, New York, NY, USA, 2004. ACM.

[28] M. Masry, D. Kang, and H. Lipson. A freehand sketching interface for progressive construction of 3D objects. In *ACM SIGGRAPH 2007 Courses*, New York, NY, USA, 2007. ACM.

[29] R. McNeel and Associates. Rhinoceros. `http://www.rhino3d.com/`, 2015.

[30] M. Mine, A. Yoganandan, and D. Coffey. Principles, interactions and devices for real-world immersive modeling. *Comput. Graph.*, 48(C):84–98, May 2015.

[31] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge. Sketch-based modeling: A survey. *Comput. Graph.*, 33(1):85–103, Feb. 2009.

[32] H. Perkunder, J. H. Israel, and M. Alexa. Shape modeling with sketched feature lines in immersive 3D environments. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM '10, pages 127–134, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[33] K. Ponto, R. Tredinnick, A. Bartholomew, C. Roy, D. Szafir, D. Greenheck, and J. Kohlmann. SculptUp: A rapid, immersive 3D modeling environment. In *IEEE Symposium on 3D User Interfaces (3DUI)*, pages 199–200, 2013.

[34] C. B. Rubin and D. F. Keefe. Hiding spaces: A CAVE of elusive immateriality. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, pages 192–192, New York, NY, USA, 2002. ACM.

[35] S. Schkolne, M. Pruett, and P. Schröder. Surface drawing: Creating organic 3D shapes with the hand and tangible tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 261–268, New York, NY, USA, 2001. ACM.

[36] U. Schultheis, J. Jerald, F. Toledo, A. Yoganandan, and P. Mlyniec. Comparison of a two-handed interface to a wand interface and a mouse interface for fundamental 3D tasks. In *IEEE Symposium on 3D User Interfaces (3DUI)*, pages 117–124, 2012.

[37] B. Shneiderman, G. Fischer, M. Czerwinski, M. Resnick, and B. Myers. Creativity support tools: Report from a U.S. national science foundation sponsored workshop. *International Journal Of Human-Computer Interaction*, 20(2):61–77, 2006.

[38] M. L. Slepian and N. Ambady. Fluid movement and creativity. *Journal of Experimental Psychology: General*, 141:625–629, Nov 2012.

[39] S. Tsang, R. Balakrishnan, K. Singh, and A. Ranjan. A suggestive interface for image guided 3D sketching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 591–598, New York, NY, USA, 2004. ACM.

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

[41] M. Xin, E. Sharlin, and M. C. Sousa. Napkin sketch: Handheld mixed reality 3D sketching. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, pages 223–226, New York, NY, USA, 2008. ACM.

[42] R. C. Zeleznik, A. Bragdon, C.-C. Liu, and A. Forsberg. Lineogrammer: Creating diagrams by drawing. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, pages 161–170, New York, NY, USA, 2008. ACM.

[43] J. Zen. Painting in air. *Comput. Graph.*, 38(3):7–9, Aug. 2004.