

Napkin Sketch

Min Xin

Ehud Sharlin

Mario Costa Sousa

University of Calgary
2500 University Drive NW
Calgary, Alberta, Canada, T2N 1N4
{mxin, ehud, mario}@cpsc.ucalgary.ca

Abstract

This paper describes, Napkin Sketch, a novel 3D sketching interface which attempts to mimic the qualities of conventional sketching medium and tools both in terms of physical properties and interaction experience. A portable tablet PC is used as the sketching platform, and mixed reality techniques are employed to allow 3D sketches to be created on top of a physical napkin. Intuitive manipulation and navigation within the 3D design space is achieved by visually tracking the tablet PC with a camera and mixed reality markers. For sketch input, we improve upon the projective 3D sketching approach with a one stroke plane creation technique. This coupled with the hardware interface produces a fluid and freeform sketching experience. Sketch recognition is also integrated as an alternative method for 3D sketch input, providing users the ability to quickly instance existing 3D models.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques – Interaction techniques; I.3.4 [Computer Graphics]: Graphics Utilities – Paint systems; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – Curve, surface, solid, and object representations; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – Artificial, augmented, and virtual realities; J.5 [Computer-Aided Engineering]: Computer-aided design

Keywords: sketch-based modeling, mixed reality, 3D design

1 Introduction

A product designer sits with his colleagues in a coffee shop. He casually sips his coffee as he scans the room. His eyes settle on a curious looking coffee cup on the counter. “What an elegant shape!” he gasps. He hastily grabs a clean napkin and starts to sketch out the shape of the coffee cup. As he renders some rough strokes on the napkin, he realizes the design, with some minor additions, would be perfect for the new product packaging he is working on. He quickly makes the modifications and adds more details to refine the design. He shows his colleagues the napkin, and a discussion breaks out amongst them.

Sketching has long been recognized as an essential process of cognition and design [Do and Gross 1997]. It is a staple in various creative fields such as architecture and engineering

especially for the preliminary stages of design and ideation. Sketching is used to quickly record ideas, explore new concepts, and communicate with others. It provides cognitive support during the design process and opens a visual dialogue between the designer and others [Plimmer and Apperley 2002]. However, despite its importance, it has been difficult to integrate the expressive but non-committing “spirit” of sketching in computer design systems. Most designers prefer using conventional sketching tools and mediums such as pencil and paper over computer interfaces for jotting down ideas or exploring new concepts. This is mainly due to the rigidity and heavy weight nature of typical computer-supported design systems both in terms of input methods and interaction styles. Often, computer-supported design requires users to be situated within a desktop setting and awkwardly manipulate graphical objects onscreen with a mouse or tablet. Bottom-up design approaches are also common, where excessive precision and details are demanded from users before an overall graphical representation is displayed. Such systems hinder rather than support the creative process because the critical thinking of users can be easily disrupted by having to attend to premature decisions or complex interfaces. Pencil and paper work well because they are ubiquitous and effortless to use. Their flexibility allows users to be informal and work in the more natural top-down design approach, where users go from rough sketches to concrete ideas through iterative exploration and refinement. However, computer systems do offer some distinct advantages over conventional design tools and mediums. Ease of editing, storage, and retrieval are certainly convenient features to have, but the most promising is the ability to actively explore the 3D design space even at preliminary design stages.

Traditionally, sketches are 2D representations of 3D ideas. This characteristic allows them to be created quickly at the early design stages, but users must mentally and physically recompose the 2D sketches into 3D representations at later design stages. The process of transferring from 2D to 3D is not only cumbersome but also redundant since users are already evaluating their designs in 3D when generating 2D sketches. Sketching directly in 3D not only streamlines the design process, but also provides users with more opportunities for visual feedback and exploration [Lim 2003]. However, the challenge is to create a computer-supported design system which offers the same ease of use, portability, flexibility, and fluidity of conventional sketching tools and mediums. In this paper, we are presenting our attempt at an intuitive 3D sketching system, Napkin Sketch, which allows users to create 3D sketches of coffee cups in coffee shops.

Our research has two main contributions. First, we present a



Figure 1: Creating 3D sketches on top of a physical napkin

novel hardware interface for 3D sketching and browsing. This involves a light weight tablet PC being used as a handheld mixed reality sketch pad and a paper napkin printed with mixed reality markers for visual tracking. The napkin serves as design medium as it did in the 2D design scenario presented earlier. It anchors the 3D design space, allowing 3D sketches to be created and viewed directly on top. Users interact with the napkin using a tablet PC. The design space can be viewed through the tablet PC via a front-facing camera, and sketches can be made by simply drawing on the tablet PC. Because the 3D design space is visually tracked on top of the napkin, navigating around the 3D sketches is straight forward and intuitive as the camera or the napkin can be physically moved to achieve different views. This interface setup is beneficial for various view dependent graphics modeling and rendering methods. Our second contribution is providing a complementary 3D sketching software interface which takes advantage of the intuitive 3D navigation provided by our hardware interface. The utility of the system is demonstrated by allowing users to construct 3D scenes using 3D perspective sketching similar to [Dorsey et al. 2007; Kallio 2005; Piccolotto 1998], where sketches created on a 2D surface are projected onto 3D surfaces (typically planes) in the scene. A novel one stroke method for quickly and intuitively creating and positioning the target 3D surfaces for projection is also presented. Sketch recognition is also implemented as a method to quickly instance existing 3D models into the scene, and with the features of the interface, these models can be arbitrarily oriented and positioned simply by sketching on different surfaces and from different views.

The rest of the paper is structured as follows. We first provide an overview of related research, highlighting the concepts and methods incorporated in our system and noting the differences. We then outline the design features of our system along with motivations for our design choices. Next, we describe the implementation of the system followed by a discussion of the preliminary assessment of the system's usability. We conclude with a look at future research directions including potential extensions of the system using other modeling techniques and providing collaborative capabilities.

2 Related Work

Computer-supported sketching and 3D design is a broad and well researched area. Topics range from direct 3D sketching in space [Sachs et al. 1991] to quasi-3D explorations of projected sketches [Tolba et al. 1999]. The challenge of creating intuitive design experiences similar to pencil and paper can be approached from many different directions. Our system makes improvements and advancements in a number of sub areas, including special 3D input devices, projective 3D sketching, and gesture-based modeling.

2.1 Special 3D Input Devices

One approach to improving the 3D design experience and avoiding the complexity of conventional 2D interfaces for 3D design is to use 3D interfaces and interaction techniques. Various systems allow users to directly create and manipulate sketches and geometry in 3D with the help of 3D tracking. Sachs et al. [Sachs

et al. 1991] introduced the concept of "design directly in 3D" with the 3-Draw system which makes use of a pair of handheld 6 degrees of freedom trackers. One tracker is used as a reference for the 3D design space, while the other is used as a stylus. The 3DM system [Butterworth et al. 1992] uses a head-mounted virtual reality display to simplify the problem of 3D model manipulation and understanding. It supports a handheld pointing device to allow users to build the 3D geometry from within the virtual 3D world. Similarly, HoloSketch [Deering 1995] uses head-tracked stereo shutter glasses, a desktop CRT display configuration, and a 3D wand manipulator to create virtual objects directly in front of the user. These direct manipulation systems are simple in concept and allow unconstrained interaction and exploration within the 3D design space, but they often require heavy weight and expensive setup and can be difficult to use due to the lack of a clear design medium and haptic feedback. For example, users of the 3DM system reported difficulties of keeping polygons parallel [Butterworth et al. 1992]. Our system draws from the advantages of direct manipulation achieved through 3D tracking for camera movements but employs a much more light weight and portable solution, using visual tracking and a handheld display.

The semi-immersive technique of mixed reality used in our system is also adopted by other works such as Cheok et al.'s [2002] curve and surface modeler. It uses a visually tracked glove and design setting to create and manipulate geometry with hand gestures. Similar systems which use tangible modeling techniques include [Piekariski and Thomas 2002; Schkolne et al. 2001]. Unlike some virtual reality systems, one advantage of mixed reality is its ability to aid the user in spatial understanding while avoiding disorientation since the virtual world is anchored on top of the physical world.

Although our system uses mixed reality and 3D tracking for camera manipulation, we have not explored the technique of direct 3D sketching due to the haptic feedback problems. Sketching is traditionally a 2D interaction technique, and moving a stylus in 3D may be unnatural for users. Therefore, we wanted users to always sketch on a 2D surface very much like they would on paper. One unique 3D input device shares this philosophy. The 3D Tractus [Lapides et al. 2006] uses a tablet PC placed on top of a mechanical table which can be moved up and down to record the third dimension. By moving the table and drawing with the pen on the tablet, complex non-planar curves can be generated. However, since the 3D Tractus design space can only be explored through axis-aligned vertical volume slices, it is not suitable for sketching arbitrary 3D geometry. Our system does not support bimanual interaction for creating non-planar curves as one hand is required to hold the tablet PC, but it does compensate for this limitation by allowing 2D sketches to be projected onto 3D surfaces. The concept of creating planar sketches on 3D planes is adopted from this system, and our system is also able to quickly generate arbitrary planes in the 3D design space.

2.2 Projective 3D Sketching

The software interface of our system and the dominant sketching interaction experience are most closely related to works involving the use of raw sketches as the visual representation of 3D designs. Often, the method for creating these sketches is through the

projection of 2D strokes onto 3D surfaces (typically planes). Users can sketch in perspective as if in 2D and still produce the desired 3D representation provided the 2D strokes are projected to the appropriate positions in the 3D design space. Sketchpad+ [Piccolotto 1998] allows users to draw strokes with a pen on a large tilted digital design table and generate 3D sketches by projecting them onto user defined grids in 3D. The positioning and orientation of these grids are specified using typical 3D manipulation operations such as rotation and translation. This system also allows the strokes to be interpreted as boundaries which enclose surfaces. 3D6B [Kallio 2005] is a similar system which uses the projected strokes approach. Its goal is to produce sketches which can be incomplete and ambiguous in nature. Therefore, it does not support surface generation through interpretation. Users can transform grid planes in 6 degrees of freedom and move them using keyboard commands while they sketch, enabling non-planar curves to be created in a similar fashion to the 3D Tractus. Tsang et al. [2004] devised a system which makes use of existing 2D images as guides for sketching 3D wire frame models. 3D models are created by sketching 2D profile curves on construction planes from top, side, and front viewpoints. Construction planes can also be moved to allow users to sketch curves in different locations, and non-planar curves are supported by projecting 2D strokes onto non-planar surfaces. The system also employs many suggestive techniques by using the guide image and interpreting the context of the sketches [Tsang et al. 2004].

Another relevant work to our system is Mental Canvas [Dorsey et al. 2007]. This system is designed to allow architects to organize concept drawings in 3D. Architects first make several regular 2D sketches of their design from different viewpoints. These are stored by the system and later fused together to generate a 3D sketch representation. The process of combining the 2D sketches from various viewpoints is similar to the projected strokes approach. 3D planes are first defined in the appropriate locations using typical 3D manipulation operations. Selected 2D strokes are then pushed or projected onto these planes. The system takes an interesting approach for dealing with occlusion: users can paint parts of a plane opaque with a 2D binary texture map to hide the strokes shown behind it. This system does not technically support 3D sketching but rather makes the process of transferring from 2D to 3D more efficient because the strokes of the 2D sketches are reused.

Although projective 3D sketching provides the closest sketching interaction experience to pencil and paper, a major bottleneck of this approach is the placement of the surfaces or planes which the sketched strokes are projected on. The systems described above make use of conventional 3D manipulation operations to position and orient the surfaces and planes in the design space. These operations are often slow and non-trivial, potentially disrupting the users' creative process. Our system improves upon this problem with a one stroke method to allow users to quickly continue sketching as they switch between surfaces or planes. Occlusion is also an issue with the associated 3D representation approach since 3D models are not explicitly defined. Instead of adding complexity to our system with a surface interpretation feature, we follow a similar approach to Mental Canvas and allow users to use oversized strokes to cover up large surfaces.

2.3 Gesture-based Modeling

The approach for evoking the one stroke method for projective sketching and the sketch recognition functionalities of our system fit in the sub area of gesture-based modeling. Gesture-based modeling is geared towards creating 3D models through a set of specialized 2D gestures or sketches. Often, interpretations and assumptions are made by the system in order to deal with the multiple mappings of 2D to 3D. Works such as SKETCH [Zelevnik et al. 1996], Teddy [Igarashi et al. 1999], and Cherlin et al.'s system [2005] make use of gesture sets for different modeling operations to create 3D geometry from 2D sketches. Sketch recognition can be seen as a special case which attempts to map a single gesture or sketch to a modeling operation or 3D model [Severn et al. 2006]. Our system employs this approach for quickly instantiating existing 3D models and activating the operation to generate arbitrary surfaces for projective sketching in the 3D design space. We take advantage of the intuitive 3D navigation of our system to establish various visual contexts which are used to automatically orient and position the created 3D geometry.

3 System Description

The scenario presented at the beginning of the paper motivates the design of our Napkin Sketch system (Figure 1). Our goal is to create a 3D sketching interface which comes close to the ease of use, portability, flexibility, and fluidity of conventional sketching tools and mediums. In this section, we will describe the features of our system and justify our design choices.

3.1 Sketch Pad and Napkin

Most of the systems described in the previous section are implemented in stationary desktop settings. Although some can be ported to run on portable computing devices like tablet PCs, their interfaces are not designed to function fluidly on such platforms. For example, many of the systems make heavy use of keyboard controls which would be difficult to access on tablet PCs. With our Napkin Sketch system, portability is a primary concern in the design of the input interface. We want users to be able to use the system anytime anywhere because inspiration comes spontaneously, and a sketching tool must be always on hand to allow users to jot down their ideas. Also, being able to design in everyday environments with a wide range of rich visual stimuli rather than sterile designated work settings is beneficial for creative tasks. Therefore, we have chosen to use an ultra portable tablet PC which can be easily held in one hand as the sketch pad of our input interface. We believe a variety of similar mobile platforms such as PDAs, camera phones, UMPCs, and portable gaming devices can eventually rival the ubiquity of pencil and paper. To sketch in our system, users simply make strokes with a pen on the 2D surface of the tablet PC. These strokes are recorded by the system and displayed at the exactly location where the pen touches the surface. This gives users a direct correspondence between their hand movements and the sketches being created, providing a similar experience to pencil and paper.

One of the significant values of sketches is using them to explore and understand design subjects. In a 3D design setting, exploration requires examining sketches or subjects from various viewpoints. In most 3D design systems, camera control often requires the use of special modes and extra interface controls to perform rotation and translation in a serial fashion. This is a bottleneck for achieving an intuitive and fluid 3D sketch system. Research has shown that users spend a significant amount of time inspecting drawings in a computer-supported pen-based system [Lim 2003]. This translates to a heavy cognitive load when having to manipulate conventional camera controls while sketching. To solve this problem, we have incorporated 3D tracking and mixed reality to allow users to intuitively navigate around the 3D design space. The approach is similar to many handheld mixed reality systems [Wagner et al. 2005]. Our sketch pad is equipped with a front facing camera delivering live video of the physical space in front. It is then used as a mixed reality viewport to present the virtual design space superimposed on the paper napkin medium in the physical environment. Our napkin is printed with markers for visual tracking and effectively anchors the 3D design space on top of its surface. Physically changing the relative positions of sketch pad and napkin changes the viewpoint of the design space. This allows effortless camera control by simply moving the sketch pad or the napkin. The mapping is direct and significantly reduces the users' cognitive load. For example, to view sketches in detail, users would just move closer to the napkin with the sketch pad. A design choice we made with this interface is to restrict the creation of sketches or 3D geometry underneath the surface of the napkin. Although this restriction can be simply avoided technically, from a usability point of view we believe that the composed virtual scene with the physical setting would be visually confusing and less coherent. Therefore, our 3D design space is strictly "above ground" and this characteristic is enforced in other interface components displayed on the napkin.

The combination of the sketch pad and the napkin provides a light weight and relatively inexpensive handheld mixed reality interface which is easy to set up and use. Because of the strong visual coherence of the virtual scene and the physical environment, navigating around the design space is intuitive and efficient. Furthermore, mixed reality allows the physical environment to be seen by users while they sketch, providing opportunities for spontaneous visual stimulation.

3.2 Freeform 3D Sketching

The main design approach for our software interface is the notion of freeform interaction. This concept is introduced in Moran et al.'s work [1995] on implicit structures for pen-based systems. Typically, computer systems deal with information as formalized representations. This means explicit structures are defined and maintained by the system to manage the representations. Text is an example where words are arranged in a sequential fashion and cannot appear on top of each other. Analogously, in 3D design, surface models are formalized representations because there is an explicit structure to the way vertices, edges, and faces are organized. With formalized representations, if an element is created, modified, or deleted, other elements are affected as well. For example, if the vertex of a face is removed, the face no longer exists. Informal representations such as sketches in the traditional

sense also have structure, but this structure is implicit as it exists only when perceived by the user. For example, 4 lines forming a rectangle can be interpreted as a plane. In contrast to formal representation, when the user removes a line, none of the other lines are affected; only the interpretation of the lines may have changed. Keeping the structures of representations implicit or temporary is the essence of freeform interaction. We feel this approach fits our design goal because it allows the system to be flexible and easy to use. One major advantage of freeform interaction is its ability to support ambiguity which is a highly valued feature of pencil and paper. By incorporating freeform interaction in 3D sketching, users can suggest different designs with unconstrained strokes to layout their rough ideas and not worry about having to deal with explicit geometric structures before they are ready.

Following the concepts of freeform interaction, we are using projective 3D sketching as the main method of creating 3D representations in our system. Users sketch on the tablet PC as they view the 3D design space on top of the napkin. The recorded 2D strokes are projected onto a 3D surface in a way such that the projected 3D stroke looks identical to the original 2D stroke. This visual correspondence along with the directness of sketching on the tablet provides a natural sketching experience for users. Because no explicit structures need to be interpreted for the sketches, users are free to use solid, overlapping, and stipple strokes to indicate contours or hatching and scribbling to suggest surfaces.

3.3 Frames

One important element in our system is the concept of frames. These are temporary 3D surfaces which are placed in the scene for the sketches to be projected on. In terms of freeform interaction, these are the implicit structures we can define to guide the sketching process. Frames can be regular primitives such as spheres, cones, and cylinders. They can also represent more complex surfaces such as b-spline patches or mesh models. The simplest frame is a plane. Cognitively, users can think of the surfaces of the 3D geometry they wish to design as frames or parts of frames. For example, a vertical plane can be temporarily perceived as the wall of a house. The user can then sketch the appropriate boundaries of the wall on the plane to make the idea more concrete or scribble and fill in the area to further solidify the concept. Currently, only plane frames are supported in our system, but the interaction techniques presented can also be transferred to other types of frames with minor modifications. In our system, only one frame is active or can be sketched on at any time during the sketching process. This avoids confusion as frames can possibly occlude each other, and some frames have infinite extent. Adhering to the concept of the napkin as the ground plane of the 3D design spaces, frames can only be situated on top of the napkin. Frames with infinite extent are truncated at their intersection with the napkin plane. This ensures that no strokes can be sketched underneath the napkin.

As noted in the related work section, the major interaction bottleneck of the projective 3D sketch approach is the creation

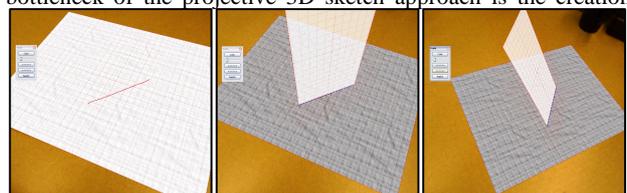


Figure 2: One stroke technique (left to right: frame creation stroke, created perpendicular frame, and rotated frame)

and positioning of the surfaces for the sketches to be projected on. Many [Dorsey et al. 2007; Kallio 2005; Piccolotto 1998] have taken the conventional 3D manipulation approach of translating and rotating the surfaces to the desired location in the 3D design space. This non-trivial process can be disruptive to the creative thinking of the users as they must stop sketching, correctly position a surface, and then resume sketching again. In our system, the actions of sketching and switching frames are closely intertwined. Our approach to frame creation and positioning tries to minimize the interruption this process can cause to users. Currently, we use a one stroke gesture to create a new plane frame in the approximate desired location of the 3D design space and rotation around one axis to precisely position the plane frame if needed.

The concept of our one stroke method (Figure 2) is to take advantage of the implicit geometric relationships of the strokes in a sketch. Strokes rarely float in space, visually disconnected from others. For example, in the sketch of a cube, the lines sketched for the front face of the cube are perpendicular to the receding lines of the side face of the cube. Therefore, when users finish sketching the front face of the cube on one plane frame, it is natural for them to wish to switch to a plane frame that is perpendicular to the previous plane and also intersects the previous plane at one of the edges sketched for the face (Figure 3). This would allow them to quickly continue sketching the side face of the cube. Our one stroke method supports this process. The system allows users to make a special frame creation stroke. This stroke is projected on the current active frame just like regularly sketched strokes. Since the system currently only supports plane frames, a relatively straight stroke is interpreted as a straight line, and a new plane frame is created which includes the line and is perpendicular to the plane frame that the line is sketched on. For example, if a frame creation stroke is sketched on the napkin ground plane, a vertical plane frame would be created at the location where the stroke is made and oriented to be parallel with the stroke. However, the one stroke only allows new plane frames that are perpendicular to the previous plane frame to be created in one step. Therefore, we also allow users to rotate the created plane frame along the axis of the sketched frame creation stroke to cover all possible plane frame orientations. However, non-perpendicular plane frames can also be created in two or more steps without rotation, but this makes the frame creation process more complex and less intuitive. We feel that generating perpendicular planes with one stroke is a logical choice because many everyday objects have contours with perpendicular relationships. Our plane switching technique allows users to transition fluidly from one plane to another as they follow the contours and surfaces of their design. Note that non-planar frames can also be generated in this fashion. For example, a curve stroke instead of a straight stroke can create a lofted curve surface.

Sometimes, users may want to backtrack to previous frames to generate new planes in the desired locations. With the cube example, a user may sketch the front face of the cube, finish working on one side face, and then wish to sketch the other side face. Although a continuous flow of sketching may take the user to the back face first and then around to the other side face, this is also a logical work flow since the user may be already comfortable with the way one side is sketched and want to replicate it immediately on the other side. Our system supports

frame backtracking by storing several previous frames in a buffer. Users can cycle through previous frames and use them as a starting point for creating new frames. This allows the system to better deal with the unpredictability associated with the sketch flow of different users. Since the napkin ground plane anchors the 3D design space, it is often a good starting point for sketching or creating frames. Therefore, we allow users to easily set the napkin plane frame as the current active frame for convenience and also if they happen to become disoriented in the 3D space.

Since it is easy to become lost in virtual 3D space without proper visual cues, we have carefully designed the appearance of the plane frame in our system. Because plane frames are infinite, they are difficult to visualize. In practice, quads must be rendered to indicate their position in 3D. However, if rendered too big, the quad would cover the display, and it would be difficult to judge its location. If rendered too small, it may be challenging for users to judge the positions of their sketches outside the boundaries of the quad. We try to infer the proper size of the newly created plane frames from the frame creation stroke. If no other display adjustments apply such as truncation by the napkin, new plane frames are created with the same width as the creation stroke and twice the width of the stroke is set as the height. The display boundaries of the plane frames are rendered as stipple red lines, and they gradually fade into space to indicate the infinite nature of the plane. The creation stroke is replaced with a straight blue stipple line which indicates the plane frame's axis of rotation. The quad itself is highlighted in translucent white if the plane frame is the current active frame. To provide a 3D visual cue, we also display the previous plane frame in translucent gray. Since the napkin ground plane is a critical visual cue for anchoring the 3D design space, we reorient the visual boundaries of the plane frames so that the top and bottom edges of the frame are always parallel to the napkin. We also try to extend the boundaries of the plane frame so the bottom edge touches the napkin, provided the plane is not parallel to the napkin, and the extension does not make the boundaries too expansive (Figure 3). This allows users to easily find the surface of the napkin if required and helps them judge the 3D positions of their sketches. Faint grid lines are also displayed on the plane frames to provide a perspective cue and help users sketch straight lines.

3.4 Sketch Recognition

In our system, we use sketch recognition as an alternative way of sketching 3D content (Figure 4). Although this technique does not follow the freeform interaction concept, we want to show that various techniques can be combined together to create an enjoyable 3D sketching experience and demonstrate the flexibility of our hardware interface. Sketch recognition does not allow users to create new 3D content, but it does allow them to quickly instance existing 3D models. This is beneficial for composing a scene with commonly used models such as trees. Although this

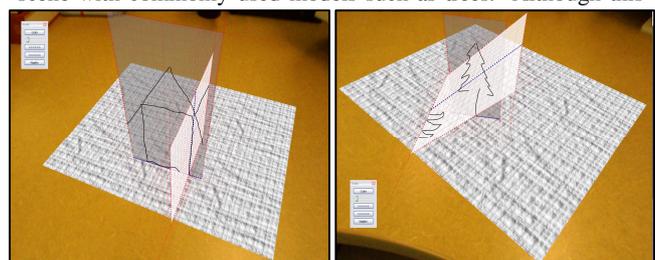


Figure 3: Intuitive frame switching with one stroke technique and frame extension display method

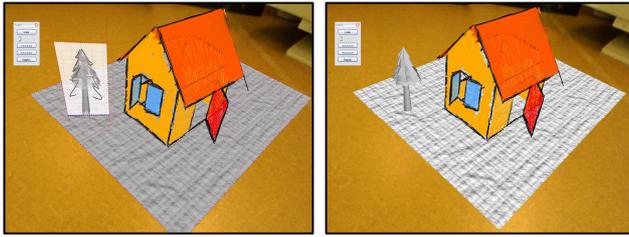


Figure 4: *Sketch recognition*

feature of the system is vastly different from the main sketching process, we have tried to integrate them in a seamless manner. Users still sketch by projecting strokes on 3D surfaces as before, but the system organizes strokes into a set based on time delay. The recognition engine then tries to match the set of strokes to predefined templates. If a match is found, the model is displayed on top of the strokes rendered as a translucent overlay. Users can ignore the suggestion by simply continuing to sketch or they can decide to use the recognized model; in which case it will be rendered with solid colors. Our system tries to match the position and orientation of the instanced model to achieve visual coherence with the recognized strokes. This is accomplished using the orientation of the plane frame which the strokes are sketched on and the position of the camera. We project the horizontal edge of the screen on the active plane frame, and use it as a reference for deriving a bounding quad for the set of strokes on the plane. The bounding quad is then used to resize and transform the instanced model. This allows users to quickly and intuitively place 3D geometry in the 3D design space. Because we only provide the recognized model as a suggestion which the users can easily ignore, this technique does not affect the fluidity of the freeform sketching or force users into early decisions. Note that sketch recognition can also be used with the frame creation stroke to efficiently create and position frames such as spheres, cones, cylinders, and mesh models.

3.5 Interface Controls

Because our goal is to provide a simple to use tool for users, most of the operations of the system are performed using the stylus of the tablet PC and its external buttons. Normal sketching is achieved by simply drawing with the stylus (left click and move with a mouse), and frame creation strokes are made by drawing with the stylus while holding down the stylus button (right click and move with a mouse) or holding down an external tablet PC button on platforms with no stylus button. We use two other external tablet PC buttons to control the cycling of the frames (Figure 5). Currently, we have a few conventional GUI components such as buttons and sliders to perform functionality that cannot be mapped to external tablet PC buttons. These include changing the color and size of the strokes sketched and rotating the plane frames around their axis.

4 System Implementation

The Napkin Sketch system (Figure 5) is designed to run on mobile platforms such as tablet PCs or UMPCs running Windows XP or Windows Vista. Currently, we have two tablet PCs for

testing the system: a Toshiba M200 with a Wacom active digitizer weighing at 2kg (4.4lbs) and a LG C1 with a passive touch digitizer weighing at 1.3kg (2.9lbs). Although the LG C1 is lighter and much easier to hold and move around, the lack of a stylus button makes sketching frame creation strokes a little more difficult. We have also tested the system on an OQO UMPC, but it did not run smooth enough to be interactive. For the mixed reality setup, we use a Creative notebook web cam with a resolution of 640 x 480 running at 30 frames per second to deliver video, and the software development system, ARToolKitPlus [Wagner and Schmalstieg 2007], is used for visual tracking. Our napkin is a sheet of standard letter size paper printed with ARToolKitPlus markers positioned in a grid. The toolkit processes the video for target markers and derives OpenGL projection and modelview matrices for displaying the virtual 3D design space. Sketched strokes are polished by first applying a reverse Chaikin subdivision filter to de-noise the input. Then they are re-sampled as quadratic b-splines. The points of a stroke are projected into the 3D design space immediately as they are recorded. This allows the system to test for points that may be sketched underneath the napkin. Sketch recognition is implemented using a technique which compares quantized angular features of the strokes [Olsen 2007], and the instanced models can be rendered in an artistic fashion using the edge buffer method [Buchanan 2000].

5 Preliminary Assessment

Although no formal user studies have been conducted with our Napkin Sketch system, we have performed an informal usability evaluation with seven people of varying artistic and drawing skills. The results are positive (see Figure 6 for sample sketches), but we have also discovered some major areas for improvement. We are excited to see that users interacted with our interface in ways which are similar to conventional pencil and paper. Without explicit instructions, users took advantage of the flexibility of the system to perform sketch operations which are analogous to those in conventional 2D sketching. For example, one user started his sketches, drawing in very thin gray lines and later went back to emphasized the lines with thicker and darker strokes. This display of over sketching demonstrates the natural top-down design thinking. Other users appropriated the ability to sketch large strokes as a way to provide occlusion for suggested surfaces. They filled in surfaces with colored scribbles to block out strokes seen in the background. This shows the concept of freeform interaction. The users did not explicitly define surfaces but rather used visual representations to suggest their presence. We are also happy to see that many of the interface features we designed are useful to the users. For example, most of them found the grid

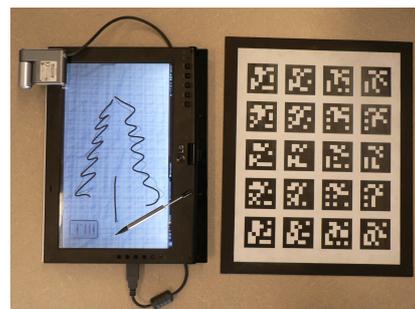


Figure 5: *Tablet PC Sketch Pad and Napkin*

lines displayed on the plane frames to be useful for drawing in perspective and measuring, all of them made heavy use of the frame cycling capability, and many of them found the 3D visual cues beneficial such as displaying the previous frame and extending the plane frame visual boundaries to the napkin. Many users also commented on the intuitiveness of camera control by physically moving the napkin.

Although we have provided perspective and 3D visual cues, the users still became lost on occasion as to how to connect lines in space or make proper sketches in perspective. Perspective sketching is not as intuitive as we anticipated. In fact, most of the users mentioned that they do not normally sketch in perspective even in 2D. Because of this, many of them frequently changed the view of the scene using the napkin to find a visual representation which minimized perspective distortion. One user even sketched explicit visual cues to help connect lines in 3D. We still feel sketching in perspective is important, but the system can definitely benefit from a richer set of visual guides which can possibly be added by the users to assist them in perspective sketching. Although the direct camera manipulation with the napkin worked well with the exception of minor tracking jitters, we found it to be awkward to use in two situations. One was when users wished to view the scene in profile, and the other was when they wished to view the scene from overhead. Viewing the scene in profile means the camera must be located at an extreme angle to the napkin, where the markers are hardly visible. Viewing the scene from overhead often causes users to stand up which was uncomfortable for them. This issue can be solved by providing better visual guides to make users more confident about sketching in perspective or allowing them to temporarily sketch in the profile and overhead views and resume napkin tracking of the scene when they are done. One last interesting finding is that users had many different types of work flows when switching planes and sketching. Some followed connected frames in sequence, and others used one frame as the reference frame and always went back to that frame for creating new frames. One user wanted to create a set of frames all at once and cycle through them to sketch without having to stop and create new frames, but this type of work flow is not supported by our system. A solution to this may be to allow users to explicitly save frames. This finding shows the diversity of the ways users sketch and hints at the need for more flexibility in our system.

6 Discussion and Future Work

The approaches we have taken for the design of our Napkin Sketch system appear promising. Judging from the limited but insightful findings from our preliminary evaluation, it seems users are adopting the interface in ways similar to pencil and paper rather than conventional 3D modeling software. One telling sign is that users are taking a variety of different approaches to interact with the system. This indicates that the users' creative thinking is not restricted by the structure of the interface. They can make their sketches from many different views and create sketches in many different ways. Therefore, although 3D sketching may take longer than conventional 3D modeling software to construct certain designs, the future direction of our system is not to find an optimal method for 3D sketching in terms of efficiency but rather to provide users with more resources for creating design solutions. For instance, implementing non-planar frames would

allow users to explore non-planar 3D curves, and allowing them to quickly select previous frames they have worked on by selecting existing sketches would provide more flexibility for their work flow. Aside from creating more alternatives for sketching and frame selection, the other major concern is how to help users (even ones with little 3D design experience) better understand the 3D design space. It is evident that our current efforts are insufficient. We need to explore better 3D visualization techniques and try to understand where bottlenecks occur in the users' 3D perception. Furthermore, we want to continue to explore the utility of our interface by incorporating more modeling techniques and provide users with more flexibility to create sketched 3D content. Currently, we are considering integrating the rotational blending method of generating freeform round surfaces from [Cherlin et al. 2005] to allow users to create a wide range of odd-shaped frames for sketching. Finally, we would also like to look at collocated collaborative sketching with several users working together on a single 3D design task through multiple individual napkin sketch interfaces.

7 Conclusion

In this paper, we have presented a novel interface for 3D sketching. Both the hardware and software interface design reflect our desire to create a 3D computer-supported design system which offers the same ease of use, portability, flexibility, and fluidity as conventional sketching tools and mediums. We start by moving the design space into the everyday environment with a portable tablet PC and napkin, mixed reality tracking, and intuitive camera controls to explore the design space. We then follow up with a fresh outlook for 3D design, grounded in the principles of freeform interaction. These, combined with projective 3D sketching and a one stroke plane creation technique, allow us to evaluate the effectiveness of our system. We hope our research can provide insight and motivation for others to explore alternative 3D design approaches.

References

- BUCHANAN, J., and SOUSA, M. C. 2000. The edge buffer: A data structure for easy silhouette rendering. In *Proceedings of NPAR 2000*.
- BUTTERWORTH, J., DAVIDSON, A., HENCH, S., and OLANO, M. T. 1992. 3DM: a three dimensional modeler using a head-mounted display. In *Proceedings of the 1999 Symposium on Interactive 3D graphics*, ACM Press, New York, 17-21.
- CHEOK, A. D., EDMUND, N. W. C., and ENG, A. W. 2002. Inexpensive Non-Sensor Based Augmented Reality Modeling of Curves and Surfaces in Physical Space. In *Proceedings of International Symposium on Mixed and Augmented Reality*, 86-94.
- CHERLIN, J. J., SAMAVATI, F. F., SOUSA, M. C., and JORGE, J. A. 2005. Sketch-based modeling with few strokes. In *21st Spring Conference on Computer Graphics*.
- DEERING, M. F. 1995. HoloSketch: a virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction*, 2, 3, 220-238.
- DO, E., and GROSS, M. D. 1997. Inferring Design Intention from Sketches - an investigation of freehand drawing conventions in design. In *Proceedings of CAADRIA 1997*, 217-227.

- DORSEY, J., XU, S., SMEDRESMAN, G., RUSHMEIER, H., and MCMILLAN, L. 2007. The Mental Canvas: A Tool for Conceptual Architectural Design and Analysis. To appear in *Proceedings of Pacific Graphics 2007*.
- IGARASHI, T., MATSUOKA, S., and TANAKA, H. 1999. Teddy: a sketching interface for 3D freeform design. In *SIGGRAPH 1999*, ACM Press / Addison-Wesley Publishing Co., 409-416.
- KALLIO, K. 2005. 3D6B editor: projective 3D sketching with line-based rendering. In *Proceedings of Eurographics Workshop on Sketch-based Interfaces and Modeling*, 73-79.
- LAPIDES, P., SHARLIN, E., SOUSA, M. C., and STREIT, L. 2006. The 3D Tractus: A Three-Dimensional Drawing Board. In *Proceedings of the 1st IEEE International Workshop on Horizontal Interactive Human-Computer Systems*.
- LIM, C. -K. 2003. An insight into the freedom of using a pen: Pen-based system and pen-and-paper. In *Proceedings of 6th Asian Design International Conference*.
- MORAN, T. P., CHIU, P., VAN MELLE, W., and KURTENBACH, G. 1995. Implicit Structure for Pen-based Systems Within a Freeform Interaction Paradigm. In *Proceedings of ACM CHI 1995*, ACM Press / Addison-Wesley, 487-494.
- OLSEN, L. SAMAVATI, F. F., and SOUSA, M. C. 2007. Fast Stroke Matching by Angle Quantization. In *Proceedings of the 1st International Conference on Immersive Telecommunications*.
- PICCOLOTTO, M. A. 1998. *Sketchpad+ Architectural Modeling through Perspective Sketching on a Pen-based Display*. Master's thesis, Cornell University.
- PIEKARSKI, W. and THOMAS, B. 2002. Tinmith-Hand: Unified User Interface for Mobile Outdoor Augmented Reality and Indoor Virtual Reality. In *Proceedings of IEEE Virtual Reality 2002*.
- PLIMMER, B., and APPERLEY, M. 2002. Computer-aided sketching to capture preliminary design. In *Proceedings of the Third Australasian Conference on User Interfaces*, Australian Computer Society, 9-12.
- SACHS, E., ROBERTS, A., and STOOP, D. 1991. 3-Draw: A tool for designing 3D shapes. *IEEE Comput. Graph. Appl.*, 11, 6, 18-26.
- SCHKOLNE, S., PRUETT, M., and SCHRODER, P. 2001. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of CHI 2001*, 261-268.
- SEVERN, A., SAMAVATI, F. F., and SOUSA, M. C. 2006. Transformation Strokes. In *Proceedings of the 3rd Eurographics Workshop on Sketch-Based Interface and Modeling*.
- TANG, S., BALAKRISHNAN, R., SINGH, K., and RANJAN, A. 2004. A suggestive interface for image guided 3D sketching. In *Proceedings of CHI 2004*, 591-598.
- TOLBA, O., DORSEY, J., and MCMILLAN, L. 1999. Sketching with projective 2D strokes. In *Proceedings of the 12th annual ACM Symposium on User Interface Software and Technology*, ACM Press, New York, 149-157.
- WAGNER, D., PINTARIC, T., LEDERMANN, F., and SCHMALSTIEG, D. 2005. Towards massively multi-user augmented reality on handheld devices. In *Proceedings of the 3rd International Conference on Pervasive Computing 2005*.
- WAGNER, D., and SCHMALSTIEG, D. 2007. Artoolkitplus for pose tracking on mobile devices. In *Proceedings of 12th Computer Vision Winter Workshop*.
- ZELEZNIK, R. C., HERNDON, K. P., and HUGHES, J. F. 1996. SKETCH: an interface for sketching 3D scenes. In *Proceedings of ACM SIGGRAPH 1996*, ACM Press, New York, 163-170.

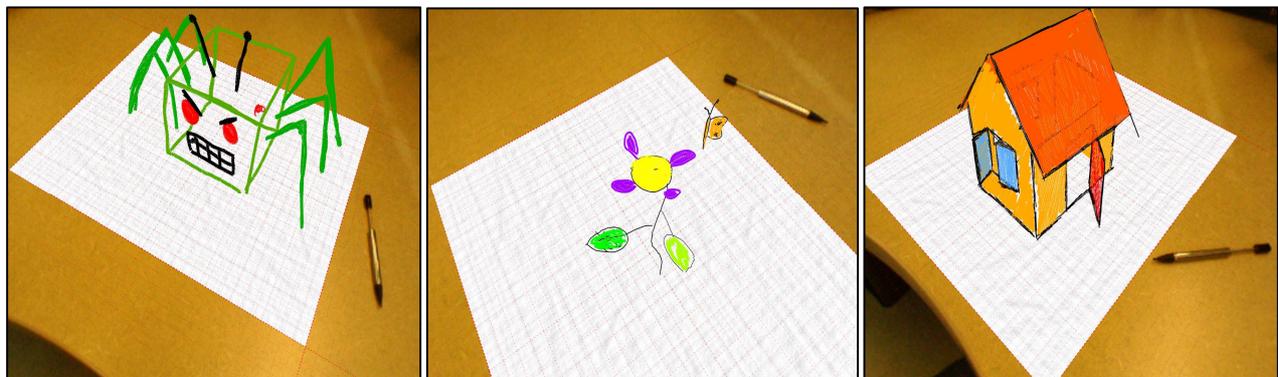


Figure 6: Interaction examples from our preliminary evaluation