iSphere

A Proximity-based 3D Input Interface

LEE Chia-Hsun, HU Yuchang and SELKER Ted

The Media Laboratory, Massachusetts Institute of Technology, USA

Keywords: 3D input device, proximity sensing, parametric modeling, human-computer interaction

Abstract: This paper presents a 24 degree of freedom input device for 3D modeling. iSphere uses the proximity information of pulling-out and pressing-in capacitive sensors to manipulate 12 control points of a 3D surface simultaneously . The iSphere dodecahedron is demonstrated manipulating an analog parametric model with high-level modeling concepts like *push* or *pull* the 3D surfaces. Our pilot experiment shows that iSphere saved many steps of selecting the control point and going through menus. Experts were used to those extra steps and still found themselves doing them but novices saved significant time for surface shaping tasks. 3D systems are benefited to execute high-level modeling commands, but lacking of fidelity is a great issue of analog input device.

1 INTRODUCTION

This paper introduces a realistic way of 3D input and manipulation interfaces. Making 3D models wasn't an easy task for 3D designers. There is a strong need to quickly transform their concepts to certain shapes. Typical 3D modeling systems, like Rhino, 3D Studio MAX or Alias|Wavefront Maya, usually consist of sets of abstract commands. Users are always performing the mediating actions between high-level modeling concepts and low-level manipulation commands. Although the 3D modelling functionality was mature in most 3D systems, the gap between realistic interaction and low-level commands is still left unsolved. To manipulate 3D environments efficiently may be the result of simplifying cognitive behavior to perform mappings and powerful commands intuitively. Designing a system which is aware of user's intention can possibly reduce users' cognitive load.

We argue that an input device which can use a spatial metaphor to map hand actions into modeling commands can improve the processes of 3D modeling. Mapping hand actions into an analog parametric model can eliminate a series of viewing and editing commands. In other words, users can use natural gestures to control the parametric model. Understand hand actions can offload some tasks from the software interface. We also argue that 3D input systems should understand user's behavior to provide interaction directly and meaningfully. Direct mapping of realistic modeling concepts, such as push, pull and twist actions should be easy to learn and remember.

iSphere is the dodecahedron with two analog sensing modes per face, as in Figure 1. Hand-position aware mechanism has been equipped into the 3D input device. It also acts like a hand interpreter which maps designer's hand signals into commands. A study was conducted to compare the performance using standard mouse and iSphere.

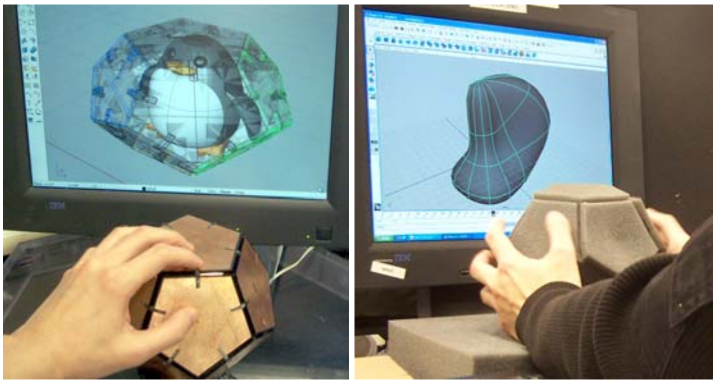

Figure 1 iSphere is a dodecahedron with capacitive sensors to interpret hand positions into high-level 3D modeling commands

2 RELATED WORK

User interface designers have dealt with 3D input problems for decades. Aish claimed that 3D input systems should be able to create and modify 3D geometry intuitively in order to interpret and evaluate the spatial qualities of a design directly (Aish 1979). But in most 3D modeling systems, command-based input and Graphical User Interfaces (GUIs) still dominate 3D Computer-Aided Design systems and have been optimized for 3D modeling. Keyboards and mice are also essential for users to type in or select commands. 3D manipulations are usually sequential and abstract. Users have to aggregate a serial of simple and abstract commands into a bigger modeling concept. It partially occupied mental resources so that designers are limited to act and think differently. There are always trivial steps before inspecting and editing 3D models that makes 3D modeling complex.

Ishii suggested a new concept to design interfaces integrating both physical and digital systems (Ishii and Ullmer 1997). Designing Tangible User Interfaces (TUIs) is to create seamless interaction across physical interfaces and digital information. Interacting with TUIs can be more meaningful and intuitive than using traditional GUIs. iSphere also extends the concept of TUI with understanding user's hand behavior in order to provide relevant modeling functions at the right time. The orienting approach of a 3D view port into a 3D world-view has been a conceptually important idea since people started creating 3D computer graphics (Van Dam 1984).

A desirable controller for a 3D environment might be a 6 degree of freedom device like a SpaceBall (Zhai et al. 1999). The space ball allows pressure forward aft side to side and up and down and rotation in X, Y, Z to control modeling. SpaceBall provides an intuitive 3D navigation experience with rotating a physical ball. But it also requires significant work with keyboard and mouse to map it into the control points and other desired function. It still takes time and steps to use physical navigation tool with mouse in order to complete a task like getting the right viewpoint and pulling the surface for 10 units along certain axis.

DataGlove usually works with 3D stereo glasses and positioning sensors. Users have to wear sensors and learn to map hand actions into manipulation commands. The advanced versions of DataGlove can provide 6DOF control and force feedback for users to model 3D under a rich immersive 3D environment. However, lacking physical references is easy to make users get lost. Working with stereo glass and wearable sensors for a long period of time may not yet be a good way. In (Zhai 1998), Zhai concluded that none of the existing 6DOF devices fulfills all aspects of usability requirement for 3D manipulation. When speed and short learning is a primary concern, free moving devices are most suitable. When fatigue, control trajectory quality and coordination are more important, isometric or elastic rate control devices should be selected. In (Zhai 1996), Zhai suggested that designing the affordance of input device (i.e. shape and size) should consider finger actions.

A 3D volume control system using foam resistance sensing techniques was demonstrated in (Murakami, T. et al. 1994). With cubical input channels and pressure-based deformation, it could provide intuitive visual feedback for deforming shapes based on a physical cube.

In (Rekimoto 2002) , SmartSkin introduced a new way of bimanual interaction on the desktop. By using capacitive sensing and embedded sensors, users can naturally control digital information projected on the table by hands. In (Llamas et al. 2003), Twister was presented as a tool of 3D input device using two 6DOF magnetic trackers in both hands to deform a sphere into any shape.

Learn from past experience in order to minimize the complexity of 3D modeling processes, this paper suggests that a physical modeling reference and the capability of using realistic hand interaction will enhance the intuitive experience of 3D modeling. Low-level operations of commands are time-consuming and costing extra efforts to complete a task in a 3D environment. A user does not have direct feedbacks from command-based manipulation and has to break concepts into trivial steps. The fragmented metal views and visual representation should be coupled in order to give designer expressive ways to model intuitively. iSphere is able to simplify the mappings between low-level manipulation commands and modeling concepts, such as pushing and pulling 3D geometries and viewpoints.

3 INTERACTIVE TECHNIQUE

3D users should be expected to consume more cognitive load on designing rather than modeling. We propose iSphere acting as a hand sensor knowing about levels of actions, like hand positions, touching, pushing and twisting actions. In most 3D modeling systems, keyboards and mice are good for command-executing and mode-switching. However, it still can't allow us to perform an editing command by a single and intuitive action. We claim that making the interaction more realistic can enhance the experience of 3D modeling.

3.1 Realistic Interaction

iSphere has been used with an editing mode for modeling 3D geometries and an inspecting mode for navigating 3D scenes. The natural mapping for an enclosed object is to map the dodecahedron to pulling on and pushing in on the surfaces as though it were clay.. Natural hand actions are mapped to the modeling commands, such as pushing multiple facets to squeeze the 3D model on that direction, as shown in Figure 2(a-d).

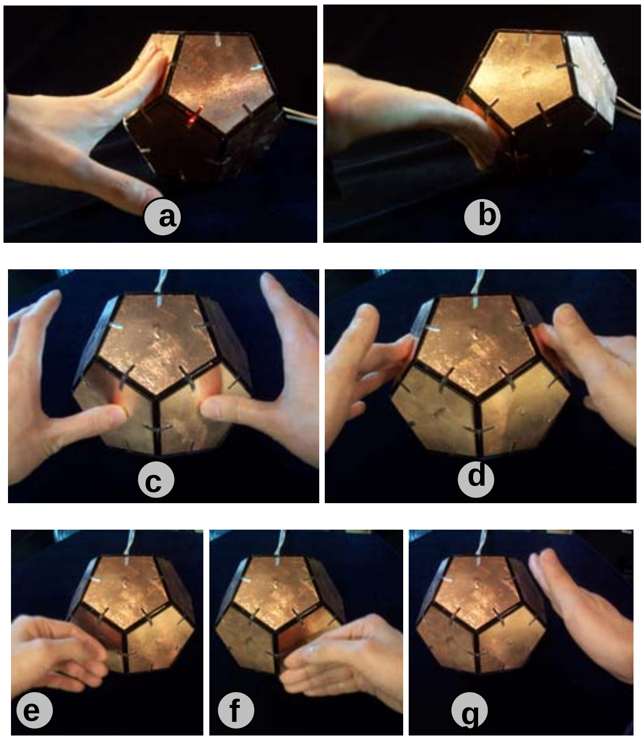

Figure 2 Hand movements as metaphors for editing and inspecting 3D scenes as realistic interaction

Visual feedback is provided in 3D software responding the 3D warp effect like playing with virtual clay when a user's hand is attempting to stretch the 3D object. In the inspecting mode, it acts as a proximity sensor which can detect the hand positions around the device. It is connected to the 3D software that rotates the corresponding camera viewpoint when a hand approaches the surface, as shown in Figure 2 (e-g). The 3D model can automatically get oriented when a user touches anyone of the surfaces. To switch the editing and inspecting mode, a functional button was installed on the desktop which allows users to switch between them by touching it or leaving it.

3.2 Play and Build

Making a 3D model requires visualizing, re-visualizing and acting. But today's CAD system is a complex enough mechanical effort that the review process might come later. Having a design goal and shaping it into 3D objects involves a series of mode-switching activities. The processes are usually trivial, disruptive, and have little relation to design. Designers designed a 3D shape and then switched to the modeling mode. Obviously, designing and decomposing shapes into sequential machinery commands are two totally different cognitive behaviors. This bottom-up approach limits the diversity of design outcomes and the expressiveness of 3D representation during the early design stage. In order to reduce the cognitive load of fragmented design mode and modeling mode, we purpose a top-down 3D modeling approach that allows designers to play and build 3D models and develop their concept directly.

4 IMPLEMENTATION

The hardware consists of 12 facets of capacitive sensors. Each of them is capable of sensing ten degrees within six inches above the surface. Push command will be triggered if hands are closer to the surfaces. Pull commands will be triggered if hands are away from the surfaces. iSphere is a dodecahedron made by acrylic. To create this device, a laser cutter was employed to make a foldable pentagonal surface. This was assembled with fitting pieces that snap it together. A circuit board which is incorporated a PIC microcontroller PIC16F88 and a RS232 serial interface was embedded in the device.

As shown in Figure 1, each side of the dodecahedron is a plastic acrylic piece, designed, with a copper backing and foam. Capacitive sensors are connected in parallel into multiplexers are able to detect the proximity of hands from twelve different directions. For long-distance proximity sensing, we use a transmitter-and-receiver setting in the capacitive sensing circuit. The small capacitance is generated by a surface that is approximately 6 inches per side of the pentagon when a hand is placed over it. A microcontroller is used for getting the digital inputs from the sensor and output the signals to the serial port to a PC.

We utilized the Alias|Wavefront Maya 6.0 C++ API (Application Programming Interface) and implemented the functions into an iSphere Plug-in. It provides us a more flexible environment to design the iSphere system. The plug-in can be loaded automatically in the command prompt in Maya.

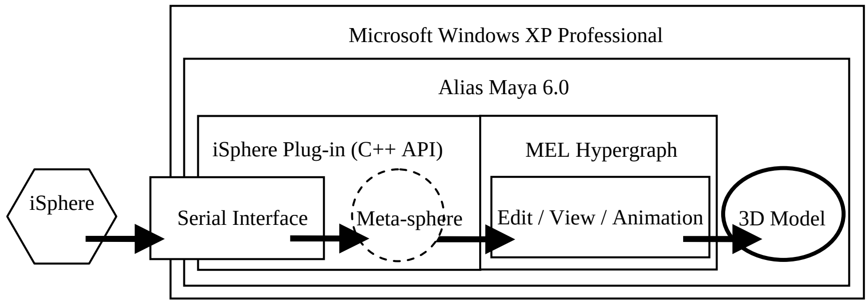

Figure 3 Software Architecture of iSphere

The software architecture can be described as a flowchart, as shown in Figure 3. First, the hardware of iSphere connected to the RS232 Serial Interface (COM1 or COM2 on an IBM PC). Second, a meta-sphere maps raw data into a data structure in order to control any 3D object. MEL (Maya Embedded Language) is handy for describing 3D modification. MEL also takes great advantages in its Hypergraph interface to easily apply 3D modification functions by drawing relationships of data flows.

The software architecture also reserves flexibility to upgrade in the future. New functions can easily be added by insert new codes or nodes into the system. Another advantage is when iSphere provides more commands, switching from different commands can be done easily by connecting the links between different nodes. iSphere is able to manipulate 3D mesh-based model in Alias|Wavefront Maya, 3DS Max or Rhino.

5 PILOT EXPERIMENT

A pilot experiment was conducted to examine potential problems before the formal evaluation. The hypothesis is that iSphere is more intuitive and efficient in modifying clay-like geometry than a general 3D modeling interface. We designed the experiment to study how efficient novices and experts can adapt 3D input techniques in different input device. The experiment contains four 3D modeling tasks and two conditions of user interface.

5.1 Experimental Set-up

A desktop 3D modeling environment was set up in the experiment. It consists of an IBM Graphics Workstation, 19” LCD display monitor, Alias|Wavefront Maya 6.0 with iSphere plug-in software, standard keyboard, mouse, and the iSphere device, as shown in Figure 4. Subjects were asked to sit on a chair where a mental strip attached on the edge providing a harmless reference signal (5Volts-20kHz) which makes the user become an antenna that can greatly improve proximity sensing.

Capacitive sensors can sense hands up to six inches above the surface. The 12 facets are also covered with foam in order to provide feedback when you touch the surface. The iSphere was installed on a soft-foam base to provide arm supports for the user. In order to enhance the 3D visualization, all 3D objects were rendered in shading mode.

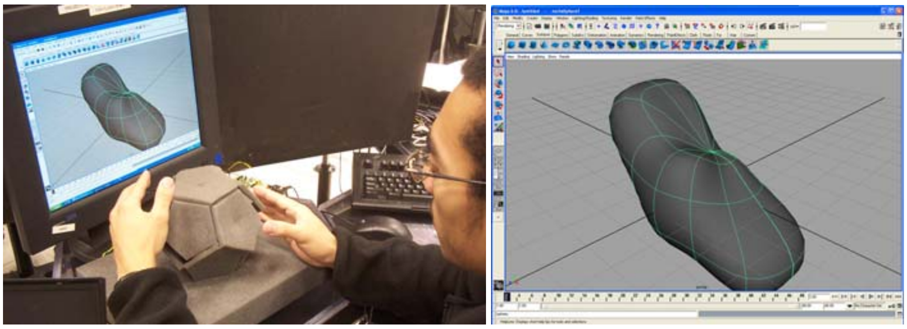

Figure 4 A pilot study of iSphere (left) and a shoe shaped by a subject (right)

5.1.1 Experimental Condition

Two experimental conditions where compared the keyboard-and-mouse desktop environment and the iSphere, were used in this experiment. For novices with no experience in Maya, using standard desktop setting to perform modeling tasks by themselves needs a learning period. Otherwise, we have to provide them more hints to get used to the interface. But for expert users, they can usually perform modeling tasks as routines. We calculated the time expense of each task using KLM-GOMS (Keystroke-Level Model GOMS) which is a model of how experts perform routine tasks (John and Kieras 1996). In this pilot study, subjects were asked to use the iSphere to perform the modeling tasks. We allowed subjects to hold their tasks and re-start again until they felt confident in tasks.

5.1.2 Experimental Task

Four 3D modeling tasks were designed in this study. Each task represents a typical 3D surface shaping procedure involving a series of view and edit commands. Subjects were asked to do four tasks in a sequence. At the beginning of each task, subjects started with a new scene with a default 3D sphere appeared in the middle of the screen. In the first task, subjects were asking to pull the top surface up to 3 units. The second task is to expend the bottom of the sphere to 3 units. In the third test, subjects were asked to make an apple. The final task is to make any shape in five minutes.

5.1.3 Experimental Design

A Two-Group design was used in comparing the performance of novices using iSphere and experts using mice. The tasks for expert users were formulated into a calculable task flowchart. In this experiment, all subjects were asked to perform tasks using iSphere. This condition was given about 15 minutes of exposure, which comprised a pre-test questionnaire, a short demonstration, and four tasks. In the first two tests, subjects were given 3 minutes to finish simple modeling tasks. In the following two tests, subjects had to finish two modeling tasks in 5 minutes. Each subject was asked to fill the post-test questionnaires after four tasks.

5.2 Experimental Results and Discussion

Six volunteers were recruited in this pilot study. Four of them had no previous experience with Maya. Two of them had intermediate level of skills in Maya. Their ages ranged from 17 to 27, with a median of 21.2. All subjects were right handed. All subjects finished the four tasks.

5.2.1 Analysis of the Overall Results

We used KLM-GOMS to analysis subjects who had intermediate experience in Maya and then compared to novice who used iSphere. Expert users used combination of shortcut keys and mouse so that they completed the tests effectively and precisely. In the first two tasks, they presented the same routine to reach the goal state. Before they started moving, they spent 3 to 5 seconds to think over possible solutions. To summarize their actions during the tasks, they spent much time and actions to move the mouse cursor to reach icons or menus on left and top of the screen. Each movement may cost 1 to 1.5 second and all movements cost around 20 to 25 seconds depending on different tasks. The next is selection, they selected corresponding CVs (control Vertex) professionally and move them to appropriate positions that the tasks asked for. The selecting and moving actions cost around 5 to 7 seconds. Clicking mouse buttons cost the shortest time in the experiment, but was the most frequently action. Each click cost around 0.2 second and 10 to 15 times.

Comparing to subjects using keyboard and mouse, the results conducted by subjects using iSphere is relatively simple t. According to our demonstration before the experiment, they all could well know how to reach the goal state by controlling corresponding facets. Therefore, all of them spent less than 2 seconds to think over the solution, before they started. The novice group spent average 8.6 seconds on the first task and average 12.5 seconds on the second task. In the third and fourth task, we weren’t able to calculate using KLM-GOMS. Although most of them spent much time to modify the model back and forth, they finished the two tests with shorter than those intermediate Maya users.

The preliminary result shows some important phenomena between using mouse and iSphere. iSphere exposes controls in a spatial way allowing users to directly manipulate the surface. It takes fewer steps than selection such controls from a tool bar and associations. Using iSphere can also reduce time consuming and actions to make simple models. Furthermore, iSphere allows users to move multiple facets at the same time. Comparing to mouse users, iSphere combines selection, direction and commands. iSphere can measure pressure an pulling at a 4 bits resolution. Currently, iSphere did not have the speed and accuracy of control that a mature analog input device gives. It is not able to perform actions precisely. In the experiment, users spent almost half of the time to move back and forth, because they cannot shape the model exactly as they wanted.

6 DISCUSSION

This paper presents a 3D input device that uses a physical modeling reference and the capability of using natural hand interaction to enhance the intuitive experience of 3D modeling. By using the information collected from its user, the system can simplify 3D modeling user interface. For modeling 3D geometries, iSphere is an input device that allows users to model 3D through hand manipulation. Using proximity of hand positions, designers can map their actions into an analog parametric model. It allows users to manipulate 3D geometries using high-level modeling concepts like push or pull the 3D surfaces. This physical device also doesn't require wearing any sensors. iSphere could change the way 3D designers work with abstract commands into natural hand interaction and intuitive 3D modeling processes, but lacking of fidelity to make detail modification is the main problem for this interface.

Designing new inputs to 12 surfaces is a complex goal. It has to do with choosing, adding and making sensors with enough fidelity, choosing metaphors that match the device, and creating a transfer function that makes sense for what is it being used for. One may argue that iSphere is a specialized device for certain specific modes, while general modelling interface is designed for general input device. Mouse and keyboard are very good at mode-switching tasks. It's not fair to compare them in certain modes of modelling.

Using iSphere, in a sense, limits the ways to model. Users can only interact with this device by hand interaction, however, it can help users to finish specific task quickly. Using mouse and keyboard, on the other hand, has more freedom to perform jobs, but in most actions, it is time consuming. Both of them represent parts of our needs when modelling. Therefore, the next step of making new modelling tools may combine these two concepts. Future work will deal with making mappings robust across shapes, methods of creating models simply and improving algorithms for sensing control. To extend the analog 3D input approach using proximity sensor, we found there are several ways to go. For example, the modeling sequences can be recorded and playback. It can be used as a motion capture device to make 3D animation from realistic interaction.

REFERENCES

Aish, R. 1979. 3D input for CAAD systems. *Computer-Aided Design* 11(2): 66-70.

Ishii, H., and B. Ullmer. 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. *Proceedings of the CHI 97,* 234-241. New York: ACM Press.

John, B., and D. Kieras. 1996. The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast. *ACM Transactions on Computer-Human Interaction* 3(4): 320-351.

Llamas, I., B. Kim, J. Gargus, J. Rossignac and C.D. Shaw. 2003. Twister: a space-warp operator for the two-handed editing of 3D shapes. *ACM Transactions on Graphics (TOG)* 22(3) 663-668.

Murakami, T., and N. Nakajima. 1994. Direct and Intuitive Input Device for 3-D Shape Deformation. In *Proceedings of the CHI 1994*, 465-470.

Rekimoto, J. 2002. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proceedings of the CHI 2002*, 113-120. New York: ACM Press.

Van Dam, Andries. 1984. Computer graphics comes of age: an interview with Andries Van Dam. *Communications of the ACM* 27(7): 638-648.

Zhai, S., E. Kandogan, B. Smith, and T. Selker. 1999. In Search of the "Magic Carpet", Design and Experimentation of a 3D Navigation Interface. *Journal of Visual Languages and Computing* 10(1): 3-17.

Zhai, S. 1998. User Performance in Relation to 3D Input Device Design. *In Computer Graphics* 32(4): 50-54.

Zhai, S., P. Milgram, and W. Buxton. 1996. The Influence of Muscle Groups on Performance of Multiple Degree-of-freedom input. In *Proceedings of the CHI 1996*, 308-315. New York: ACM Press.