iSphere: A free-hand 3D modeling interface

Chia-Hsun Jackie Lee, Yuchang Hu, and Ted Selker

iSphere: A free-hand 3D modeling interface

Chia-Hsun Jackie Lee, Yuchang Hu, and Ted Selker

Making 3D models should be an easy and intuitive task like free-hand sketching. This paper presents iSphere, a 24 degree of freedom 3D input device. iSphere is a dodecahedron embedded with 12 capacitive sensors for pulling-out and pressing-in manipulation on 12 control points of 3D geometries. It exhibits a conceptual 3D modeling approach for saving mental loads of low-level commands. Using analog inputs of 3D manipulation, designers are able to have high-level modeling concepts like *pushing* or *pulling* 3D surfaces. Our experiment shows that iSphere saved steps in the selection of control points in the review of menus and leading to a clearer focus on what to build instead of how to build it. Novices saved significant time learning 3D manipulation by using iSphere to making conceptual models. However, one tradeoff of the iSphere is its lack of fidelity in its analog input mechanism.

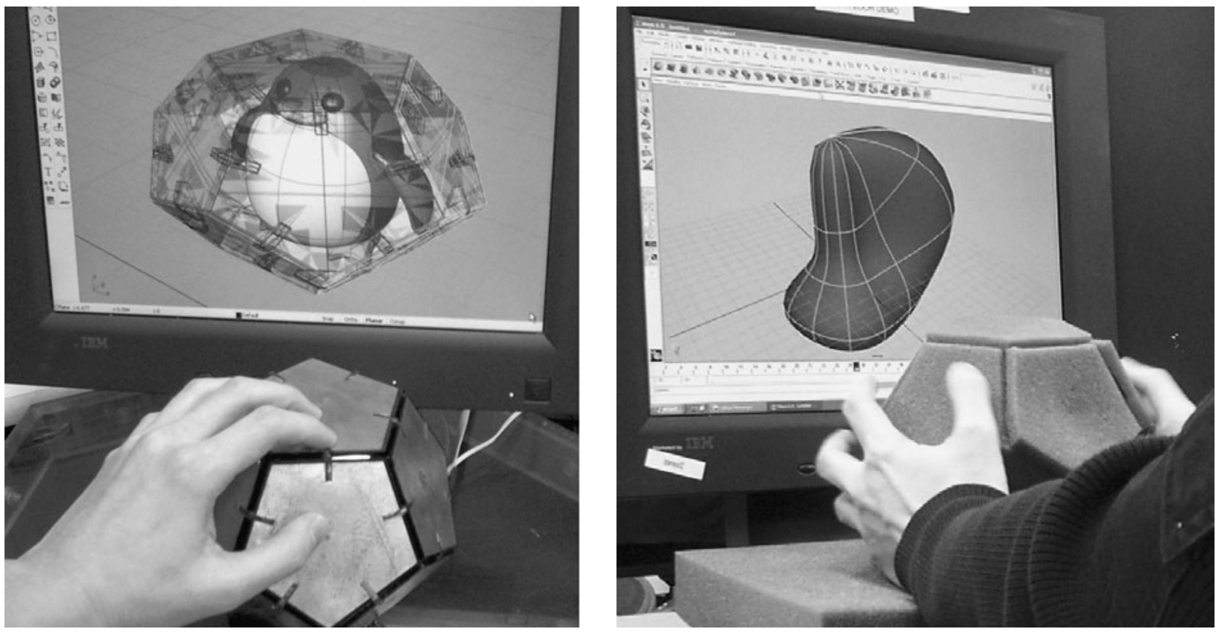
Keywords: 3D input device, proximity sensing, parametric modeling, human-computer interaction

1. Introduction

Freehand sketching is a common and effective way for designers to express ideas directly and interactively. However, creating shapes in 3D is much more complex and unintuitive. Modern CAD systems provide computational platforms for automating drawing processes and making novel representations. But input interfaces of most systems usually consist of machinery of low-level commands and trivial mode-switching manipulations. Usually, it takes months to become an expert in creating designs in those CAD systems. This paper introduces a novel way to manipulate 3D geometries by which novice users are able to build 3D models intuitively, with their own hands.

Making 3D models isn't an easy task for 3D designers. It requires complex tasks of visualizing, re-visualizing and acting. Commonly, it involves intensive mental activities in planning how to use commands rather than deciding what to make. Historically, modern CAD systems make designers take bottom-up approaches to 3D modeling. They have to set the goal of what to build and then decompose it into a series of smaller pieces before turning it into a set of modeling commands. Furthermore, it is not simple for designers to think and perform modeling commands at the same time. Dealing with trivial, disruptive and low-level steps of modeling commands uses up too much mental load. Building extra connections between representations requires more mental load than learning in meaningful process [1]. There is a critical need to quickly transform modeling concepts into shapes. Typical 3D modeling systems, like Rhino, 3D Studio MAX or Alias|Wavefront Maya, usually consist of sets of abstract commands. Users are always managing high-level modeling concepts and low-level manipulation commands, simultaneously. Although the 3D modeling functionality is mature in most 3D systems, the gap between realistic interaction and low-level commands is still present. To manipulate 3D environments efficiently, we need to simplify cognitive behavior to perform mappings and powerful commands intuitively.

We argue that a high-level modeling system can reduce users' cognitive load in 3D creations. An input device which can use a spatial metaphor to map hand actions into modeling commands can improve the processes of 3D modeling. In other words, designers should use natural gestures to manipulate 3D objects in a spatial way. 3D input systems should understand users' behaviors to provide interaction directly and meaningfully. We present iSphere, as in Figure 1, which is an input device responding to hand actions in order to eliminate software interface tasks. iSphere is a physical reference with hand-position aware mechanisms for 3D modeling. We have conducted a study to compare the performance between command-based interfaces 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 [2]. However, 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.

Ishii suggested a new concept to design Tangible User Interfaces (TUIs) by creating seamless interaction across physical interfaces and digital information [3]. Interacting with TUIs can be more meaningful and intuitive than using traditional GUIs. iSphere extends the concept of TUI with responding user's hand behavior in order to provide relevant modeling functions. The realistic 3D inspecting approach of turning a 3D view port into a 3D world-view has been a conceptually important idea since people started creating 3D computer graphics [4].

A desirable spatial controller for a 3D environment might be a 6 degree of freedom device like a SpaceBall and a DataGlove [5]. The SpaceBall allows pressure control in X, Y, Z and rotation. It maps 3D navigation experience to a physical reference. For 3D editing, it requires significant work with keyboard and mouse to control points and other desired functions. A DataGlove usually works with 3D stereo glasses and positioning sensors. Users have to wear sensors and learn to map hand actions into manipulation commands in a 3D virtual environment. The advanced versions of DataGlove provide 6-Degree-Of-Freedom control and force feedback for users to model 3D in a rich immersive 3D environment. However, lacking physical references easily makes users get lost. Working with stereo glass and wearable sensors for fatigue tasks may not yet be a good way. In [6], Zhai concluded that existing 6DOF devices cannot 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 are relevant. In [7], Zhai suggested that designing the physical affordance of input device (i.e. shape and size) should take the advantage of human motor system to have more degrees of freedom of input modes (i.e. using finger actions).

A 3D volume control system using resistive foam was demonstrated in [8]. With cubical input channels and pressure-based deformation, it provides intuitive visual feedback for deforming shapes based on a physical cube. In [9], gesture modeling provided a novel way of interacting with a virtual model directly, but the limitation of the modality of gesture language, fatigue problem, and fidelity remained unsolved.

In [10], SmartSkin introduced a novel way of bimanual interaction on the desktop. By using capacitive sensing and embedded sensors, users interact with digital information projected on a table by hands. In [11], 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.

Learning from past experiences in order to minimize the complexity of 3D modeling processes, we suggests that a physical modeling reference and the capability of using realistic hand interaction can enhance the experience of 3D modeling. Low-level operations of commands are time-consuming and cost extra efforts to complete tasks in a 3D environment. We present iSphere to simplify the mappings between low-level manipulation commands and high-level modeling concepts.

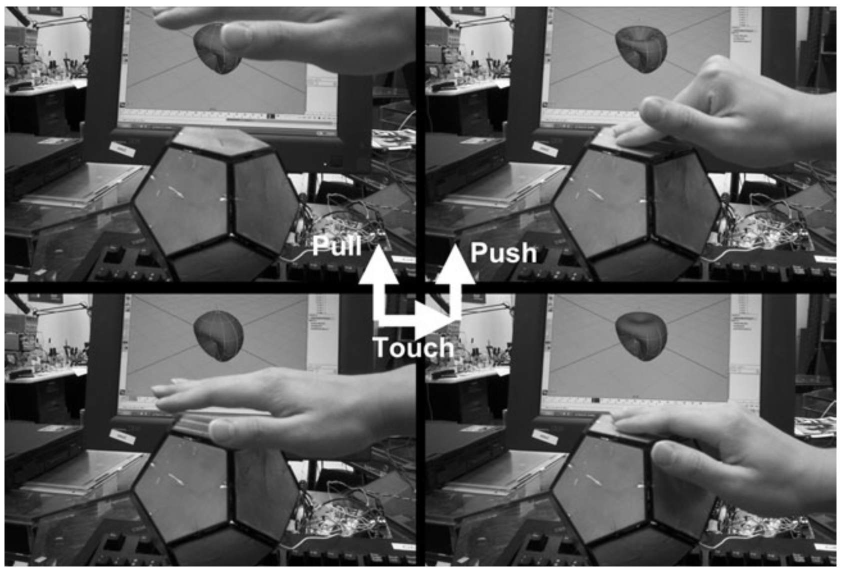
3. Interactive technique

iSphere is a physical reference for controlling 3D models interactively in a spatial way. It enables bi-manual interaction to shape 3D geometries through bare hands that makes interaction more realistic and playful which offload mental activities.

3.1. Realistic interaction

We claim that making interaction in realistic and spatial ways enhances the experience of 3D modeling. In most 3D modeling systems, keyboards and mice are good for command-executing and mode-switching. However, it still can not allow us to use a single and intuitive action for a small adjusting function. Designing is a different cognitive behavior from decomposing shapes into sequential machinery commands. Pointing and selecting actions also require intensive amounts of mode-switching activities for 2D representations. The command-based processes are usually trivial, disruptive with little relation to a design, therefore, a user does not receive direct feedback from 2D representations and command-based manipulation.

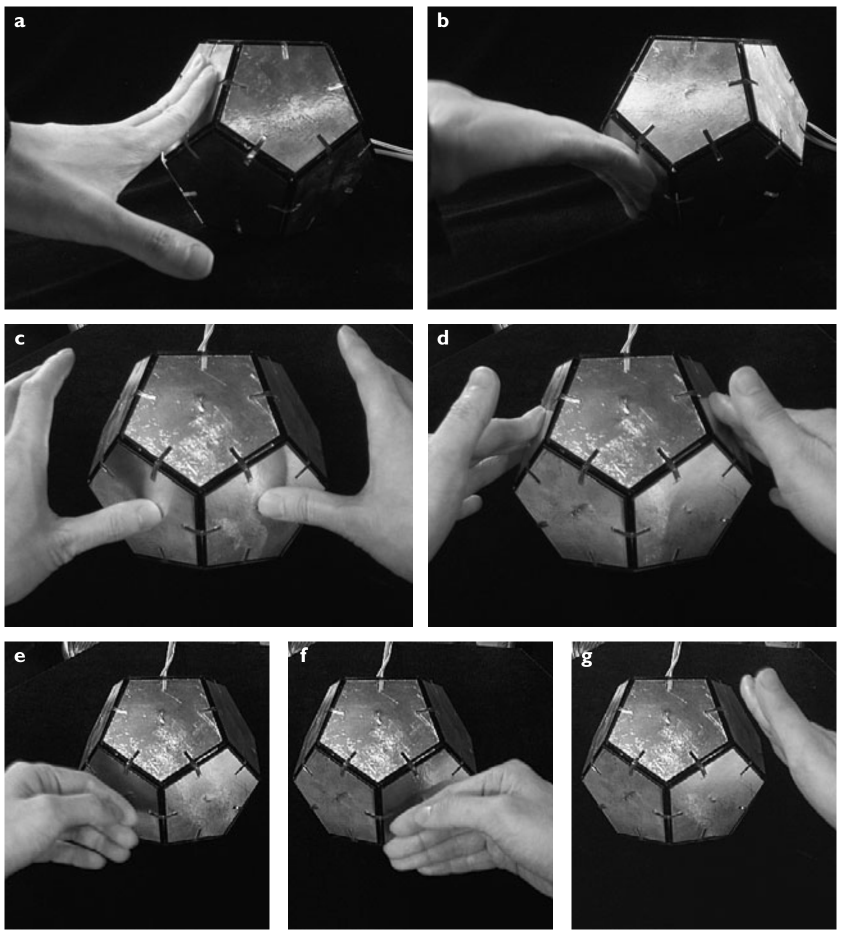
Often, spatial characteristics cannot be fully represented on a 2D plane. The fragmented metal views and visual representation should be coupled in order to give designer expressive ways to model intuitively. Traditionally, 2d maniputation limits the diversity of design outcomes during the early design stage. To enhance input device, we present iSphere that has physical and spatial references in addition to the traditional 2D representations. In Figure. 2, iSphere is able to cope with Z axis manipulation by hand actions on the physical interface. A designer uses iSphere as a dummy object for manipulation of 3D geometry and its natural mappings of gestures and physical reference make the interactive processes more realistic. Building 3D scenes can be playful and direct using iSphere for creating and playing shapes in physical and spatial ways.



◀ Figure 2. Push command will be triggered if hands are closer to the surfaces. Pull commands will be triggered if hands are away from the surfaces.

3.2. Play and build

iSphere presents a top-down 3D modeling approach that allows designers to *play and build* 3D models and develop their concept through physical interactivity. iSphere has an editing mode for modeling 3D geometries and an inspecting mode for navigating 3D scenes. We expect users of iSphere to model 3D as playing with a piece of clay. Natural hand actions are mapped to the modeling commands, such as pushing multiple facets to squeeze the 3D model on that direction. The appearance of this input interface is a 6"x6"x6" dodecahedron for holding it in hands. Each facet of iSphere is designed as a capacitive electrode that detects the distance from the human body. iSphere interprets hand actions around it into high level modeling concepts which aggregate modeling commands and x, y, z manipulations. This dodecahedron maps spatially into a meta-sphere which controls any 3D model as the starting shape. This meta-sphere also stores interaction history into a time dimension for playing back the modeling history.



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

We propose iSphere to act as a hand sensor that knows about levels of actions, like hand positions, touching, pushing and twisting actions. When the hands are away from the sensors up to 6 inches, it detects an 8 degree action of 'pull'. When the hands are very close (< 1 inch) on press, it considers the user denting with the model. In the inspecting mode, it detects hand positions around the device. It rotates the corresponding camera viewpoint when a hand approaches the surface, as shown in Figure 3. The 3D model will automatically get oriented when a user touches anyone of the surfaces. In order 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. The modeling sequences can be recorded playback and used as a motion capture device to make 3D animation from realistic interaction.

4. Implementation

iSphere is implemented with physical materials, capacitive sensors, a low-cost circuit board, and software plug-ins of modern CAD systems.

4.1. Physical interface

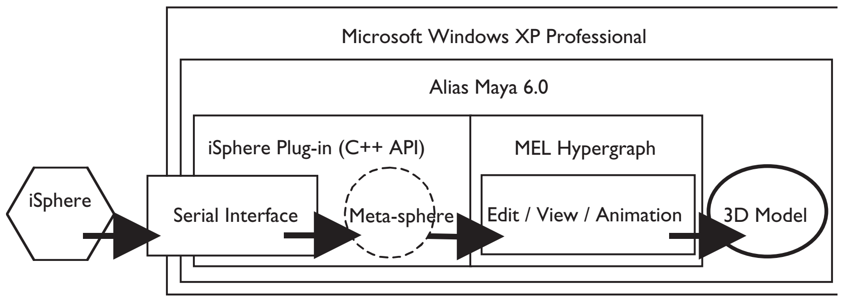
iSphere dodecahedron is made by acrylic by means of a laser cutter to make a foldable pentagonal surface. Each side of the dodecahedron is a plastic acrylic piece, designed, with a copper backing and foam. It was assembled with fitting pieces that snap it together. Each of them is capable of sensing eight degrees within six inches above the surface of the pentagon when a hand is placed over it. The 12 facets were also covered with foam in order to provide instant feedback that allows user to feel the distance from hands and the surface. Capacitive sensors are connected in parallel into multiplexers. They 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. A circuit board which incorporates a PIC16F88 microcontroller and a RS232 serial interface was embedded in the device. A microcontroller is used for getting the digital inputs from the sensors and output the signals via the serial port to a PC.

4.2. System architecture

The system architecture is described as a flowchart, as shown in Figure 4. First, the hardware of iSphere connected via the RS232 Serial Interface. Second, a meta-sphere maps raw data into a data structure in order to control any 3D object.

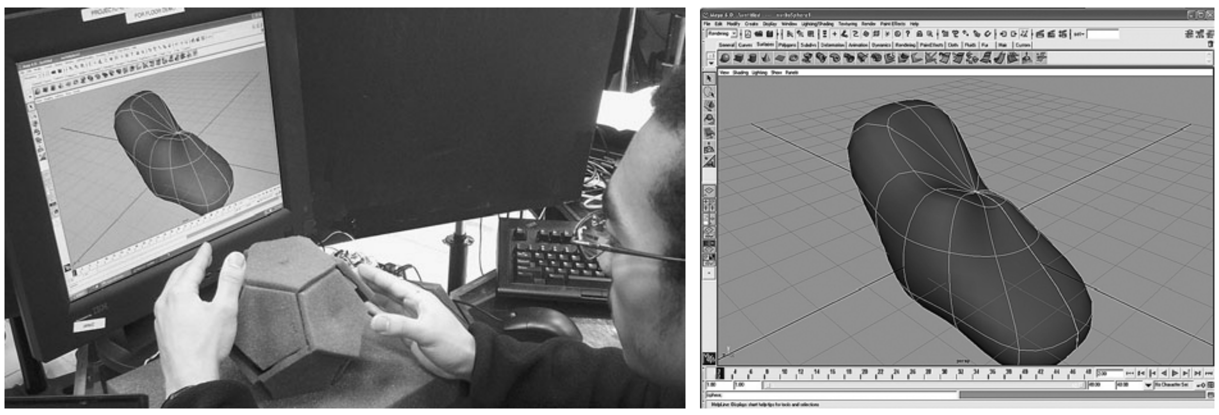
iSphere maps the analog input signals into a meta-sphere in order to drive manipulations over the target 3D object. This meta-sphere is the data structure of the dodecahedron which stores the deformation data from the sensors and maps the deformation to the desired 3D object. We utilized the Alias|Wavefront 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. MEL (Maya Embedded Language) is handy for describing 3D manipulation. MEL also has the advantages of the Hypergraph interface to easily apply 3D modification functions by drawing relationships of data flows. The system architecture also reserves flexibility to upgrade in the future. New functions can easily be added by insert new code or nodes into the system. iSphere is able to manipulate 3D mesh-based model in Alias|Wavefront Maya, 3DS Max or Rhino.

► Figure 4. The system architecture of iSphere includes an input device and a 3D software plug-in.



5. Experiment

A pilot experiment was conducted to examine potential problems before formal evaluation. The goal of the experiment is to study how novices and experts adapt 3D input techniques in different input devices. In our definition, expert means designers who are familiar with traditional 3D modeling interface, while novice are designers who are not familiar with 3D modeling interface. Our hypothesis is that when working in traditional 3D modeling interfaces, the performance of experts is much better than novices, because they are capable to employ different commands to reach their goals. If there is a more intuitive and efficient input interface to eliminate the gap, novices should have similar performance as experts.



▲ Figure 5. A subject is shaping a shoe using iSphere.

5.1. Experiment 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 5. To improve proximity sensing, subjects were asked to sit on a chair where a metal strip attached on the edge provides a harmless reference signal (5Volts-20kHz) which make the user become an antenna to increase the resolution of sensing. 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.

Experimental condition

To examine our hypothesis, both novices and experts were asked to perform modeling tasks using iSphere. In this pilot study, we allowed subjects to hold their tasks and re-start again until they felt confident in tasks. Using keyboards and mice, experts usually do tasks with procedures and actions in routine. In order to analyze knowledge of how to do each task, we decided to employ the KLM-GOMS (Keystroke-Level Model GOMS) method [12] to calculate approximate time that subjects accomplish tasks using keyboards and mice. This data is compared with studies by novices and experts using iSphere.

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 appearing in the middle of the screen. First, in the 3D Pull-up task, subjects were asked to pull the top surface up to 3 units. Second, the 3D Expend task is to expend the bottom of the sphere to 3 units. Third, in the Making-Apple test, subjects were asked to make an apple. Finally, the free-modeling task is to make any shape in five minutes. In the first two tests, subjects were given 3 minutes to finish the modeling tasks. In the following two tests, subjects had to finish them in 5 minutes. Each subject was asked to fill the post-test questionnaires after four tasks.

Experimental design

Subjects were divided into two groups by their expertise in 3D modeling. One was the novice group, the other one was expert group. Both of them used iSphere as the input device. This condition was given about 30 minutes of exposure, which comprised a pre-test questionnaire, a short demonstration, and four tasks. During the study, investigators videotaped each test.

5.2. Experimental results and discussion

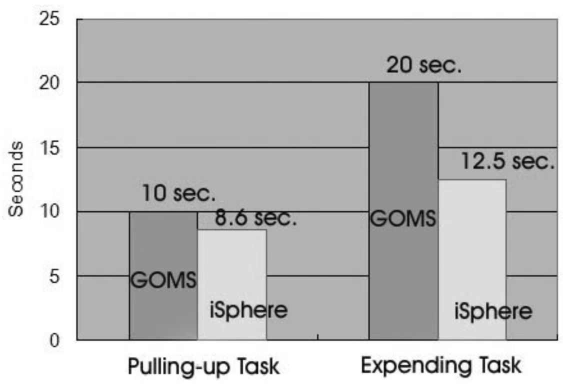
Six volunteers were recruited in this pilot study. Four of them had no previous experience with Maya. Two of them had intermediate level of skills in Maya. Their ages ranged from 17 to 27, with a median of 21.2. All subjects were right hand dominated. All subjects finished the four tasks.

Analysis of the overall results

We used KLM-GOMS to calculate the performance of routine tasks in Maya using keyboards and mice and compared to the performance of novice who used iSphere, as show in Table 1. To summarize the routine actions, we find that utilizing keyboards and mice cost much time and actions to move the cursor to reach icons or menus on left and top of the screen. Each movement may cost 1 to 1.5 second. Selecting CVs (control Vertex) professionally and moving those to appropriate positions 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 for 10 to 15 times. All movements totally cost around 10 seconds for the first Pull-up task and 20 seconds for the second Expending task.

► **Table 1. GOMS analysis is used for calculating time consumption of routine actions.**

Actions	Seconds
Mentally prepare	1.35
Move cursor	1.10
Click mouse button	0.2
Press/release mouse button	0.1



► Figure 6. Using iSphere for basic 3D edit tasks is less time-consuming.

After demonstrations of using iSphere, all subjects understood how to reach the goal by controlling corresponding facets. As shown in Figure 6, the novice group spent average 8.6 seconds on the Pulling-up task and average 12.5 seconds on the Expending task. Although most of them spent much time to modify the model back and forth, they finished the two tests in a shorter span than those intermediate Maya users.

The preliminary result shows that iSphere exposes controls in a spatial way that allows users to manipulate the surface directly. It takes less time expense than selection such controls from tool bars and menus. iSphere reduces time consuming and actions for making simple models. Furthermore, iSphere allows users to move multiple facets at the same time. For mouse users, iSphere provides ease of use that combines selection, direction and commands. Currently, iSphere did not have the speed and accuracy of control compare to a mature analog input device. It is not able to perform actions precisely. Therefore, in the experiment, users learned to gain control over iSphere to move 3D surfaces back and forth, because they need to get used to the way of shaping the model.

6. Discussion

The paper suggests that freehand 3D modeling can off-load low-level manipulations from modeling processes and take advantages of modeling in realistic and spatial ways. Designers should be able to think and evaluate the shape continuously without involving too many interruptions from the manipulation commands. iSphere is a 3D input device using a spatial modeling reference. It has capability for using natural hand interaction in order to enhance the experience of 3D modeling. Using iSphere, the user can pay more attention to and focus on what's in their mind and hands instead of menus or commands.

iSphere changes the way 3D designers work from abstract commands into natural hand interaction and intuitive 3D modeling processes. Spatially mapping hand actions into analog inputs on a parametric model can eliminate a series of viewing and editing commands. Direct mapping of realistic modeling concepts, such as push, pull and twist actions should be easy to learn and remember. However, the lack of fidelity and accuracy is an aspect in need to be improved.

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 modeling interface is designed for general input device. Mouse and keyboard are very good at mode-switching tasks. It is important to note that each performs better in certain modes of modeling. Future work includes dealing with robust mappings across shapes, methods of creating models simply, and improving algorithms for sensing control.

Acknowledgements

We thank ITRI for sponsoring this research work and Rob Gens, Minna Ha and Elliott Pretcher for their MIT UROP contributions.

References

1. Mayer, R., Moreno, R., Nine Ways to Reduce Cognitive Load in Multimedia Learning. *Educational Psychologist*, Vol 38 Issue 1, 2003, 43-52.
2. Aish, R., 3D input for CAAD systems. *Computer-Aided Design*, 1979, 66-70.
3. Ishii, H. and Ullmer, B., Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. *Proc. Of CHI 97*, ACM Press, 1997, 234-241.
4. Van Dam, A., Computer graphics comes of age: an interview with Andries Van Dam. *Communication of the ACM*, Vol 27 Issue 7, 1984, 638-648.
5. Zhai, S., Kandogan, E., Smith, B., and Selker, T., In Search of the "Magic Carpet", Design and Experimentation of a 3D Navigation Interface. *Journal of Visual Languages and Computing*, 10(1), 1999, 3-17.

6. Zhai, S., User Performance in Relation to 3D Input Device Design, *Computer Graphics*, 32(4), 1998, 50-54
7. Zhai, S., Milgram, P. and Buxton, W., The Influence of Muscle Groups on Performance of Multiple Degree-of-freedom input, *Proc. Of CHI 96*, ACM Press, 1996, 308-315.
8. Murakami, T. and Nakajima, N., Direct and Intuitive Input Device for 3-D Shape Deformation. *Proc. Of CHI 94*, ACM Press, 1994, 465-470
9. Gross, M.D., Kemp, A., Gesture Modeling: Using Video to Capture Freehand Modeling Commands, in: De Vries, B., ed., *Computer Aided Architectural Design Futures 2001: Proceedings of the Ninth International Conference*, Kluwer Academic Publishers, 2001, 271-284
10. Rekimoto, J., *SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces*. Proceedings of the CHI 2002, ACM Press, 2002, 113-120
11. Llamas, Ignacio., Kim, B., Gargus, J., Rossignac, J. and Shaw, C.D., Twister: a space-warp operator for the two-handed editing of 3D shapes. *ACM Transactions on Graphics (TOG)*, 22(3), 2003, 663-668.
12. John, B. and Kieras, D., The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast. *ACM Transactions on Computer-Human Interaction*, 3(4), 1996, 320-351.

Chia-Hsun Jackie Lee, Yuchang Hu, and Ted Selker,
Massachusetts Institute of Technology,
Context-Aware Computing Group, The Media Laboratory,
20 Ames ST., E15-324, Cambridge, MA 02139,
+1 617 253 4564

{jackylee, yhu, selker}@media.mit.edu