Usability evaluation of interfaces for 3D interaction

Grissom, Scott Benson, Ph.D.

The Ohio State University, 1992
USABILITY EVALUATION OF INTERFACES
FOR 3D INTERACTION

DISSERTATION

Presented in Partial Fulfillment of the Requirements
for the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Scott Benson Grissom, B.E.D, M.S.

* * * * *

The Ohio State University
1992

Dissertation Committee:
W. Carlson
G. Perlman
R. Parent

Approved by

Advisor
Department of Computer and Information Science
In Memory Of My Father
ACKNOWLEDGEMENTS

There are several people I wish to thank for providing assistance in completing this work. My major advisor, Wayne Carlson, provided guidance and direction for this project and thoughtfully reviewed several drafts of this dissertation. My minor advisor, Gary Perlman, taught me about sound experimental design, statistical analysis, and was very helpful with defining the outline of this dissertation. Rick Parent also provided several helpful comments.

This work could not have been completed without the help of thirty-six volunteer subjects who each spent approximately two hours performing SEP(3D) tasks. Several system administrators allowed access to their software and computer labs to conduct the experiments. Brenda Stern and Lynda Grissom reviewed video tapes for the reliability study of evaluator bias. The Department of Computer and Information Science provided outstanding facilities for the development of TeeterTable.

I am most grateful to my family. My mother provided financial and emotional support when needed. My adorable daughters, Samantha and Alexandre, provided much needed distractions along the way. My lovely wife, Jane, provided support and encouragement throughout this process. I am especially grateful for her love and belief in me.
VITA

April 20, 1963 ........................................... Born - Dallas, Texas
1985-87 ............................................... Research Assistant, College of Architecture,
                                      Research Center, Texas A&M University
1986 ................................................... Bachelor of Environmental Design, Texas
                                      A&M University, College Station, Texas
1987-89 ................................................ Graduate Teaching Associate, Department of
                                      Computer and Information Science,
                                      The Ohio State University
1989 ................................................... M.S., Computer and Information Science,
                                      The Ohio State University, Columbus, Ohio
1989-92 ................................................ Undergraduate Advisor, Department of
                                      Computer and Information Science,
                                      The Ohio State University

PUBLICATIONS


FIELDS OF STUDY

Major Field: Computer and Information Science
Minor Fields: Experimental Psychology and Human Factors
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>DEDICATION</th>
<th>ii</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>VITA</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
</tbody>
</table>

## CHAPTER

<table>
<thead>
<tr>
<th>I. INTRODUCTION</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motivation</td>
<td>1</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>2</td>
</tr>
<tr>
<td>Previous Standardized Evaluations</td>
<td>2</td>
</tr>
<tr>
<td>Previous Approaches To Usability Evaluation</td>
<td>5</td>
</tr>
<tr>
<td>Summary</td>
<td>12</td>
</tr>
<tr>
<td>Overview Of Remaining Chapters</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>II. INTERFACES FOR 3D INTERACTION</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction Hardware</td>
<td>13</td>
</tr>
<tr>
<td>Visual Presentation</td>
<td>14</td>
</tr>
<tr>
<td>Interaction Styles</td>
<td>17</td>
</tr>
<tr>
<td>Previous 3D Interaction Techniques</td>
<td>17</td>
</tr>
<tr>
<td>Design Guidelines For Usability</td>
<td>22</td>
</tr>
<tr>
<td>TeeterTable: A New Metaphor For 3D Interaction</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>III. EVALUATION PLAN</th>
<th>33</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective</td>
<td>33</td>
</tr>
<tr>
<td>Subjects</td>
<td>34</td>
</tr>
<tr>
<td>Task Analysis</td>
<td>35</td>
</tr>
<tr>
<td>Stimuli</td>
<td>39</td>
</tr>
<tr>
<td>Environment</td>
<td>40</td>
</tr>
<tr>
<td>Usability Measures</td>
<td>42</td>
</tr>
<tr>
<td>Covariate Measures</td>
<td>45</td>
</tr>
<tr>
<td>Procedure</td>
<td>46</td>
</tr>
<tr>
<td>Confounding Factors</td>
<td>47</td>
</tr>
</tbody>
</table>
APPENDICES

A. SEP(3D) Instruction Booklet ............................................................ 113
B. SEP(3D) User Satisfaction Questionnaire ................................... 139
C. Subject Responses To Open-ended Items ................................. 147

LIST OF REFERENCES ..................................................................................... 156
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURES</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Visual Cues (a) wireframe (b) solid (c) shaded (d) shaded with shadows</td>
</tr>
<tr>
<td>2</td>
<td>Cutplane (Edwards et al. 1988)</td>
</tr>
<tr>
<td>3</td>
<td>Cursor movement zones for a triad mouse</td>
</tr>
<tr>
<td>4</td>
<td>The skitter moves on a face of the object (Bier 1986)</td>
</tr>
<tr>
<td>5</td>
<td>Handle Box (Houde 1992)</td>
</tr>
<tr>
<td>6</td>
<td>3D Pointer</td>
</tr>
<tr>
<td>7</td>
<td>TeeterTable Modeling Environment</td>
</tr>
<tr>
<td>8</td>
<td>Rotating the table counterclockwise (a) before, (b) during, (c) after</td>
</tr>
<tr>
<td>9</td>
<td>Tilting the table upwards (a) before, (b) during, (c) after</td>
</tr>
<tr>
<td>10</td>
<td>Moving an object (a) selecting object, (b) moving along XY grid, (c) moving in vertical direction above the grid</td>
</tr>
<tr>
<td>11</td>
<td>Sizing an object by using a face (a) selecting face, (b) stretching along a perpendicular axis, (c) shear object by stretching along a different axis</td>
</tr>
<tr>
<td>12</td>
<td>Sizing an object by using a vertex (a) selecting the vertex, (b) increasing width and depth, (c) increasing height</td>
</tr>
<tr>
<td>13</td>
<td>Object rotation (a) selecting object, (b) rotating about vertical axis, (c) rotating about horizontal axis</td>
</tr>
<tr>
<td>14</td>
<td>Creating a cube (a) selecting first corner, (b) dragging out base, (c) adjusting the height</td>
</tr>
<tr>
<td>15</td>
<td>SEP(3D) tasks selected to represent all possible tasks</td>
</tr>
<tr>
<td>16</td>
<td>Sample task instruction</td>
</tr>
<tr>
<td>17</td>
<td>Sample questions for an individual task</td>
</tr>
<tr>
<td>18</td>
<td>TeeterTable Genealogy</td>
</tr>
<tr>
<td>FIGURES</td>
<td>PAGE</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>19</td>
<td>Visual cues for rotation (a) original method indicated a plane around the object to be controlled in the same way as the TeeterTable, (b) modified method clearly indicates the vertical and horizontal rotation axes</td>
</tr>
<tr>
<td>20</td>
<td>Interface comparisons across several dimensions</td>
</tr>
<tr>
<td>21</td>
<td>Pearson's correlation between performance times and covariate measures (n = 27). An asterisk indicates a significant correlation (p &lt; .05)</td>
</tr>
<tr>
<td>22</td>
<td>Description of box plot</td>
</tr>
<tr>
<td>23</td>
<td>Paired comparison figure</td>
</tr>
<tr>
<td>24</td>
<td>Performance times for Move XY</td>
</tr>
<tr>
<td>25</td>
<td>Paired comparisons of performance times for Move XY</td>
</tr>
<tr>
<td>26</td>
<td>Performance times for Move XYZ</td>
</tr>
<tr>
<td>27</td>
<td>Paired comparisons of performance times for Move XYZ</td>
</tr>
<tr>
<td>28</td>
<td>Average questionnaire ratings for Move</td>
</tr>
<tr>
<td>29</td>
<td>Performance times for Size XY</td>
</tr>
<tr>
<td>30</td>
<td>Paired comparisons of performance times for Size XY</td>
</tr>
<tr>
<td>31</td>
<td>Performance times for Size XYZ</td>
</tr>
<tr>
<td>32</td>
<td>Average questionnaire ratings for Size</td>
</tr>
<tr>
<td>33</td>
<td>Performance times for Rotate V</td>
</tr>
<tr>
<td>34</td>
<td>Performance times for Rotate H</td>
</tr>
<tr>
<td>35</td>
<td>Paired comparisons of performance times for Rotate H</td>
</tr>
<tr>
<td>36</td>
<td>Average questionnaire ratings for Rotate</td>
</tr>
<tr>
<td>37</td>
<td>Performance times for view manipulation</td>
</tr>
<tr>
<td>38</td>
<td>Paired comparisons of performance times for view manipulation</td>
</tr>
<tr>
<td>39</td>
<td>Average questionnaire ratings for view manipulation</td>
</tr>
<tr>
<td>40</td>
<td>Performance times for Create</td>
</tr>
<tr>
<td>41</td>
<td>Paired comparisons of performance times for Create</td>
</tr>
<tr>
<td>42</td>
<td>Average questionnaire ratings for Create</td>
</tr>
<tr>
<td>43</td>
<td>Average questionnaire ratings for Visual Feedback</td>
</tr>
<tr>
<td>FIGURES</td>
<td>PAGE</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>44 Average performance times for all tasks</td>
<td>80</td>
</tr>
<tr>
<td>45 Paired comparisons of average performance for all tasks</td>
<td>80</td>
</tr>
<tr>
<td>46 Average questionnaire ratings for general characteristics</td>
<td>80</td>
</tr>
<tr>
<td>47 Average Ranks for Performance Times and Ratings</td>
<td>81</td>
</tr>
<tr>
<td>48 Performance Times for Size XYZ</td>
<td>83</td>
</tr>
<tr>
<td>49 Performance Times for Rotate H</td>
<td>84</td>
</tr>
<tr>
<td>50 Average performance times for each trial. The first session includes</td>
<td>85</td>
</tr>
<tr>
<td>trials one through five. The second session includes trials six through</td>
<td></td>
</tr>
<tr>
<td>ten</td>
<td></td>
</tr>
<tr>
<td>51 Pearson's correlation between trials (n = 64)</td>
<td>85</td>
</tr>
<tr>
<td>52 Overall Average performance times for the first and second sessions</td>
<td>88</td>
</tr>
<tr>
<td>53 Performance times for Move XY during the first and second sessions</td>
<td>88</td>
</tr>
<tr>
<td>54 Performance times for Move XYZ during the first and second sessions</td>
<td>88</td>
</tr>
<tr>
<td>55 Performance times for Size XY during the first and second sessions</td>
<td>89</td>
</tr>
<tr>
<td>56 Performance times for Size XYZ during the first and second sessions</td>
<td>89</td>
</tr>
<tr>
<td>57 Performance times for Rotate V during the first and second sessions</td>
<td>89</td>
</tr>
<tr>
<td>58 Performance times for Rotate H during the first and second sessions</td>
<td>90</td>
</tr>
<tr>
<td>59 Performance times for View during the first and second sessions</td>
<td>90</td>
</tr>
<tr>
<td>60 Performance times for Create during the first and second sessions</td>
<td>90</td>
</tr>
<tr>
<td>61 Unintended shear (a) before adjustment is made, (b) after adjustment,</td>
<td>95</td>
</tr>
<tr>
<td>(c) side view of unintended change</td>
<td></td>
</tr>
<tr>
<td>62 Validation of each task. A check mark indicates a significant</td>
<td>105</td>
</tr>
<tr>
<td>difference was detected.</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

This chapter begins with an introduction to the motivation for this work and a problem statement. The chapter continues with a description of standardized evaluation plans and a review of software evaluation literature. An overview of the remaining chapters is also provided.

MOTIVATION

Interactive computer graphics is a powerful tool for the design of everything from molecular structures to jet aircraft. Until recently, most 3D design work was done with multiple 2D views such as orthographic projections. A variety of 3D systems exist, but most of them are difficult to use. Users must understand engineering-oriented concepts such as X, Y, Z coordinates and local versus absolute frames of reference (Houde 1992). Many of these systems are used exclusively to prepare engineering drawings after the design has been performed. Designers often use paper to capture their initial ideas because existing 3D systems do not offer the freedom and flexibility of a pencil (Sachs et al. 1989).

Few interfaces of 3D systems have been empirically evaluated. Designers frequently claim that their technique is ‘better’ than other techniques but with no evidence to prove this claim. It is therefore important to develop a method for evaluating the usability of interfaces for 3D systems. Usability can be defined and measured in a number of ways, but for the purpose of this study it is considered to be the ease with which a system can be used to perform 3D manipulation tasks.

Evaluating 3D interfaces is particularly difficult because they are used for a wide variety of purposes. Landscape architects manipulate trees and shrubs in a design while biochemists manipulate molecules. Only a few empirical comparisons are found in the
graphics literature but these studies only evaluated individual techniques and were not a thorough evaluation of interfaces for 3D manipulation (Chen, Mountford and Sellen 1988; Grissom, Carlson and Perlman 1989; Ware and Osborne 1990).

PROBLEM STATEMENT

A standardized evaluation plan (SEP) is designed to evaluate or compare a wide variety of systems that share certain capabilities. This is contrasted with a specific evaluation that is tailored to a particular purpose or situation. A SEP for 3D interaction would be useful in comparing existing and proposed interfaces. It would also be useful in extending the knowledge of what makes an effective interface for 3D interaction. Therefore, the objective of this work is to develop a standardized evaluation plan for the usability of interfaces for 3D interaction, called SEP(3D). To my knowledge, such a plan does not exist.

Several factors were considered when developing SEP(3D): 1) What are the core 3D tasks that most effectively represent the 3D design process? 2) What usability measures are appropriate for evaluating these techniques? 3) What steps can be taken to insure the plan is reliable and valid for comparisons across a wide range of 3D systems? 4) What steps can be taken to make the plan simple enough for almost anyone to use without special equipment or training?

PREVIOUS STANDARDIZED EVALUATIONS

Several standardized plans have previously been used to evaluate different types of software. A review of their methods and objectives is useful before presenting the specifics of SEP(3D).

SEP Objectives

Roberts and Moran (1983) explain that a standardized evaluation does not make assumptions about particular situations and does not attempt to analyze every aspect of a system. It focuses on common properties of systems rather than specialized differences. The evaluation is applicable to a wider range of systems by focusing on fundamental issues. For example, a word processing package may have an advanced feature such as
automatic generation of an index. This would not be an appropriate requirement for a SEP because many word processing packages do not have this capability. This advanced feature would preclude more basic systems from comparison and therefore limit the scope of the evaluation.

An evaluation plan must produce reliable (consistent) and valid (measuring what is intended) results (Hix and Schulman 1991). It must not be biased in favor of any particular system's conceptual structure and it must be thorough enough to include most important aspects of the typical system. Ideally, the plan should be simple enough that anyone without specialized equipment or expertise can use it. A benefit of using a standardized evaluation over time is the accumulation of consistent information about the class of systems being evaluated. It also allows independent verification of the plan's validity (Roberts and Moran 1983).

Procedures For Developing A SEP

Roberts (1991) suggests the first step when developing an evaluation plan is to select a representative set of tasks that are common to all systems. Consideration should be given to what people do with these kinds of systems. Once the tasks are developed, experiments can be generated that compare users with similar experience.

The major difficulty in developing a method for evaluating software is deciding what makes the software 'good' or 'bad'. The evaluation becomes straightforward when these objectives have been established. Since people use software to achieve goals, the system is successful if it is helpful to the person in achieving these goals. Evaluations should provide a measure of how easily users can perform tasks to achieve goals (Karat 1988).

Text Editors

Roberts' (1980) intent was to develop a SEP for text editors that could be used to obtain objective and replicable results. She used four criteria to evaluate text editors:
1) Time to perform basic editing tasks by experts. 2) Error time of experts while performing the core tasks. 3) Learnability of basic editing tasks by novices. 4) Functionality over a wide range of editing tasks.
Roberts identified a small subset of all possible editing tasks as the \textit{core editing tasks}. These tasks include basic operations such as insert, delete, and copy to basic text objects such as characters, words and paragraphs. A set of \textit{benchmark tasks} was used to determine performance time of experts. The amount of observed \textit{error time} was subtracted from the total elapsed time to arrive at the \textit{error-free time}. Error time was approximated with the use of a stopwatch by the evaluator. Roberts and Moran claim that this approximation is accurate enough and helps to keep the plan simple. Borenstein (1985) questioned the use of the stopwatch as a reliable measure of time but concluded from his experiments that it was adequate.

\textbf{Comment:} \textit{The only data used in Roberts' study were objective performance measures. The evaluation plan ignored potentially useful subjective data that measure user satisfaction of the system. The plan only allowed comparisons between the collection of core tasks rather than between individual tasks.}

\textbf{User Interface Development Tools}

Hix and Schulman (1991) developed a standardized plan to evaluate user interface development tools. Experts completed a detailed checklist organized around two criteria: functionality of the tool and usability of the tool. \textit{Functionality rating} is a percentage of the number of possible tasks that can be performed with a tool with respect to the total number of tasks in the domain. \textit{Usability rating} is a measure of the subjective analysis of the tool. Evaluators rate each task with one of three subjective categories: difficult to use, adequate to use, and easy to use.

They attempted to validate the plan by showing that the results were reliable. Functionality was shown to be more reliable than usability. Overall, Hix and Schulman (1991) claim that the plan was shown to be reliable.

\textbf{Comment:} \textit{This plan relies only on subjective data and potentially valuable performance data are ignored. A thorough evaluation should provide a variety of data.}
2D Tasks

Baecker and Buxton (1987) suggested six generic 2D tasks to compare input devices:
1) pursuit tracking, 2) target acquisition, 3) freehand inking, 4) tracing and digitizing, 5) constrained linear motion, and 6) constrained circular motion. They suggested experiments to test each task and discussed several measures to consider for each. The experiments were mainly designed for use as student projects.

Comment: Although several core tasks and appropriate measures are suggested, this work was not intended as a standardized plan. The authors' main intent was to provide a set of experiments to compare input devices on a single computer. The stimuli are not clearly specified or standardized and could not be used to reliably compare devices on different computer systems.

PREVIOUS APPROACHES TO USABILITY EVALUATION

Designing an effective evaluation plan requires selecting appropriate usability measures and following a sound experimental method. A significant amount of work has been done on general software usability evaluation. A review of the literature indicates several strategies that can be included in SEP(3D).

Bury describes two different types of usability testing performed at IBM (Mills et al. 1986). The first is a carefully controlled experimental comparison of specific design variables. These experiments are used to answer well-defined interface design questions. The second general category is iterative product testing. Subjects perform a series of representative tasks. Their behaviors while performing the tasks are monitored and problems are identified. Changes are made to the product and the test is repeated. This testing and product redesign sequence is performed several times until some criterion of acceptance is obtained. Only a few subjects are needed for this technique because it is less formal than most experimental comparisons. There are usually no attempts to obtain statistically significant results.
Questionnaires

Evaluations often include questionnaires to capture a user's experience (Bailey and Pearson 1983; Coleman et al. 1985; Rushinek and Rushinek 1986; Chin et al. 1987). Although these data are sometimes criticized for being subjective, responses can be carefully taken to provide reliable results (Karat 1988). Responses should be obtained from at least an interval scale rather than a rank ordinal scale. For example, an ordinal scale may indicate that System A is better than System B which is better than System C. The differences between the systems cannot be estimated. Results from an interval scale are more useful because they not only provide rank order but they also provide meaningful differences between pairs. For example, System A is 5 ‘units’ better than System B and 10 ‘units’ better than System C. The ‘unit’ of measurement is determined by the scale provided for responses. Two of the most common scales provided for responses are one through seven and one through ten.

Root and Draper (1983) investigated the strengths and weaknesses of questionnaires as software evaluation tools. Questionnaires were found useful because they are inexpensive, easy for software teams to apply, and effective at identifying the good and bad aspects of an interface. Results were most reliable when questions were answered by subjects immediately after using a system. Checklist-style questions about specific aspects of the interface were found to be the best type of question. However, they are not effective at identifying features that should be added to a system. Appropriately phrased, open-ended questions can be used for this purpose.

Checklist Method

Ravden and Johnson (1989) proposed a checklist method for evaluating user interfaces. A series of realistic tasks is performed by the user before completing the checklist. Questions are based on a set of goals that a well-designed user interface should aim to meet. The method is versatile because the questions can be answered by a variety of people with different levels of expertise. It is not used to solve problems but to identify difficulties, weaknesses, and areas for improvement. Each question is answered with one of the following responses: always, most of the time, some of the time, and never.
Nine sections of questions are designed to provide information about the interface and how it meets the following goals: 1) visual clarity, 2) consistency, 3) compatibility, 4) informative feedback, 5) explicitness, 6) appropriate functionality, 7) flexibility and control, 8) error prevention and correction, and 9) user guidance and support. Each section begins with a description of the design goal to inform the evaluator in what context the questions should be answered.

**QUIS**

Chin, Diehl, and Norman (1988) developed the Questionnaire for User Interface Satisfaction (QUIS) to measure the user's subjective rating of the interface. The short version of QUIS had twenty-seven items organized into the following groups: 1) overall reactions to the software, 2) screen, 3) terminology and system information, 4) learning, and 5) system capabilities. Each item had a response scale from 0 to 9 and anchored at both ends with bipolar adjectives (i.e., difficult / easy).

Comment: *An important aspect of QUIS is that numerical responses allow statistical analysis of the results.*

**Controlled Experiments**

Controlled experiments are used to obtain valid and reliable data. These typically include measures such as time to complete tasks, error rate, and time to learn a system. Care must be taken to include appropriate subjects as well as useful measures. Review of empirical comparisons in the literature suggests appropriate measures and experimental controls for SEP(3D).

**Comparing Input Devices**

Several experiments have been reported that evaluate input devices for the task of pointing and dragging (Epps 1986; Gillan et al. 1990; Boritz, Booth, and Cowan 1991; MacKenzie, Sellen, and Buxton 1991). Card, English, and Burr (1978) compared a mouse, rate-controlled isometric joystick, step keys, and text keys for the task of text
selection. The target was controlled for size, distance and approach angle. Factors that were considered included learning curves, speed, and accuracy. The mouse was found to be fastest and had the lowest error rates. They discovered that for continuous movement devices, positioning time can be predicted with Fitts' Law. This law describes the speed-accuracy tradeoff: movement time and target size are reciprocally related. Longer movements can be made, but if time is to be kept constant, then the target size must be increased (Wickens 1984). However, positioning time using the key devices could be predicted with a different model, the number of keystrokes.

Comparing Selection Techniques

Potter, Weldon, and Shneiderman (1988) evaluated three strategies for touch screens to perform selection. Data were collected for performance time, error rate, and user satisfaction. After using all three strategies, subjects were asked to rate the strategies on the dimensions of ease of learning, awkwardness, and satisfaction. The combination of objective and subjective data provided useful information for comparison. The authors also note that just watching the users perform tasks provided valuable information for improving the strategies.

Comparing Interaction Styles

A usability study measured performance time and user satisfaction for performing a standardized task to compare seven different interfaces representing command, menu, and iconic interface styles (Whiteside et al. 1985). The tasks included simple file manipulation operations such as displaying, merging, and sending files to another user. Subjects rated their reaction to the system on a seven point scale. The authors drew three general conclusions. 1) There are large usability differences between contemporary systems. 2) There is not necessarily a tradeoff between ease of use and ease of learning. 3) Interface style is not related to performance or preference (but careful design is).

Comment: Although several significant differences were found by analyzing the performance data, only one difference was found when analyzing user
satisfaction. This suggests that subjective data was not as sensitive as objective data in identifying differences.

Comparing 3D Rotation Techniques

Chen, Mountford, and Sellen (1988) compared several mouse-based methods for performing 3D rotation. Subjects were shown a house on one side of the screen and were asked to match its orientation to a rotated house on the other side of the screen. Performance measures of completion time and accuracy were recorded. Tasks were divided between simple rotations that involved a single axis and complex rotations that involved multiple axes. Subjects responded to a single question, “Which method do you prefer?”

Comment: The limited subjective data from this study cannot be used to predict the degree of differences between systems. For example, does the fact that all subjects prefer one method over another indicate that it is twice as preferred or maybe just marginally preferred? In addition, asking only one general question does not provide enough useful information. Subjective data should include responses to specific questions and require a scaled response to be useful.

Comparing 2D Interaction Techniques

Performance measures of completion time, accuracy and user satisfaction were used in a study to compare the use of hand gestures and virtual buttons to control the movement of 2D objects (Grissom, Carlson and Perlman 1989). Direct manipulation techniques were superior when a high degree of accuracy was required. However, both methods were similar when low accuracy was allowed. Subjects indicated which method they preferred.

Comment: The core tasks used in this study appear to be suitable for standardized comparisons but the evaluation plan was not designed to be used with other computers. Stimuli were presented on the computer screen, which means that the software would have to be ported to other computers. This would not be
convenient when comparing a variety of computers. In addition, the subjective data were limited because they was based on an ordinal scale instead of an interval scale.

Comparing View Manipulation Techniques

Ware and Osborne (1990) evaluated three metaphors for virtual camera control using a 6D input device. Subjects were asked to manipulate three simple environments. The authors deliberately departed from traditional experimental studies by collecting a limited amount of data. They used a technique called "semi-structured" interviewing to compare the three metaphors. Subjects were asked several questions to determine the relative comparisons of the metaphors and the individual strengths and weaknesses. Subjects ranked the metaphors in response to three questions. However, the authors note the most useful information came from observation of subjects performing the tasks.

Comment: Although this method seemed to provide useful information for direct comparison of the three strategies, results cannot be transferred to other studies since objective measures were not taken.

Observation

Lund (1985) describes a method of evaluating usability that includes video taping subjects performing benchmark tasks. The technique was used to discover problem areas in the interface and to verify that changes made have improved the user interface. Reviewing a session allows designers to see where users have difficulty. Lund focused primarily on help functions and associated documentation of a system. Revisions were made to the system after specific problems were identified and then tests were repeated to determine if the changes improved performance. He found that subjects made some of the same errors but recovery was quicker. Subjects were asked to verbalize their thoughts while performing the tasks.
Comment: An advantage of the technique is that a limited number of subjects can provide meaningful information. System designers gain insight to user problems because they are able to 'see' the problems for themselves and perhaps recognize why the problem occurred. Subject verbalization gives otherwise unavailable insight to the observer.

**Formal Design Analysis**

Error-free performance of an expert can be predicted analytically using the Keystroke-Level Model (Card, Moran and Newell 1980). This model is used to predict performance time by counting the number of physical and mental operations required to complete a task and assigning a standard time for each operation. This method proved to be useful in predicting the time of text editing tasks (Roberts and Moran 1983).

Comment: This model would not be useful with 3D interaction techniques because the mental operations involved are more difficult to predict. Using this type of model would require more evaluator experience than SEP(3D) requires.

**Comparison of Evaluation Techniques**

An experiment was performed to compare several evaluation techniques (Jeffries et al. 1991). *Heuristic evaluation* involves a user interface (UI) specialist studying the interface in depth and looking for properties that they know from experience will lead to usability problems. The authors claim that this method is limited because UI specialists are scarce. *Usability testing* is performed during actual use of a system by intended users. However, this is time consuming to acquire and evaluate the large amounts of data that are collected. A third method involves using *guidelines* that provide evaluators with specific recommendations about the design of an interface, such as the organization of screens or the proper use of color. This method forces a careful examination of the entire interface regarding the accepted guidelines.
For comparison, each evaluation method was used to evaluate the same software product. Evaluators completed problem report forms. Problems were categorized for severity by three independent raters. Results indicated that each evaluation technique has strengths and weaknesses. The heuristic technique was used to recognize the largest number of problems at the lowest cost. However, a team of UI specialists is necessary to pool their suggestions. Usability testing was used to identify several problems but at a very high cost. Guidelines evaluation was the best technique for finding general problems and can be used by people that do not necessarily have UI expertise. However, this method failed to discover several severe problems. Specific problems were not identified in the article but comparisons were based on the number and severity of problems.

SUMMARY

SEP(3D) would be useful for comparing interfaces and extending the knowledge of what makes an effective interface for 3D interaction. An evaluation plan must produce valid and reliable results. Results will be valid if subjects perform a set of tasks that accurately represent the domain. Results will be reliable if appropriate measures are obtained and if the experiment is designed to reduce the effects of confounding factors. Appropriate measures for usability are performance time, user satisfaction, accuracy, and learning time.

OVERVIEW OF REMAINING CHAPTERS

The remainder of this work is divided into six chapters. Chapter two is an introduction and review of the literature associated with 3D interfaces. A description of a new 3D system that was developed to evaluate the capabilities of SEP(3D) is also provided. Chapter three presents the details of SEP(3D). Chapter four describes two experiments to test the reliability and validity of SEP(3D). Results of the experiments are provided in chapter five and a discussion of the results in chapter six. Conclusions about SEP(3D) and suggestions for future work are presented in the final chapter.
CHAPTER II
INTERFACES FOR 3D INTERACTION

This chapter provides an introduction to several basic elements of 3D interfaces: interaction hardware, visual presentation, interaction styles, and interaction techniques. In addition, a review of the literature reveals design guidelines for improving usability. Finally, an experimental system is presented that was developed for this study.

INTERACTION HARDWARE

Computer hardware for interaction can be divided into devices for input and devices for output.

Input Devices

Input devices can be evaluated on three levels: device, task, and dialogue, which is a sequence of several tasks (Foley et al. 1990). The device level is related to the hardware characteristics of the device. For example, using a touch screen may cause more arm fatigue than using a mouse. Specific interaction techniques are evaluated at the task level. For example, selecting an object with a mouse might be faster than typing a long name to specify the object. Finally, we evaluate sequences of tasks at the dialogue level. Experiments may indicate that three separate devices are ideal for performing the specific tasks A, B, and C. However, switching between devices takes time. Switching from the mouse to a button box and then to the keyboard may take longer to perform the sequence (A -> B -> C) than if only one device was used.
Output Devices

Many of the problems with existing systems are a result of the limited amount of 3D visual feedback. A designer cannot clearly pick a 3D point or move in 3-space. Part of the difficulty relates to the 2D nature of the computer screen. Several 3D output devices are available, but they have not been readily accepted by customers (Forrest 1986). One of these devices is the head-mounted display developed by Ivan Sutherland in the 1960's. More recent devices include special 3D screens, but the user must be viewing from a specific position or wear special goggles to see the depth effect. However, the most popular device continues to be the traditional CRT screen. The resolution, size, and color have steadily improved making current screens more successful in presenting 3D graphical information. A traditional screen that presents enough depth information would be acceptable for most 3D applications.

VISUAL PRESENTATION

Visual cues are an important component of a 3D system. Effective visual cues are used to enhance interaction techniques. However, these cues are often ignored or not used to their full potential. An understanding of effective visual cues for clarifying a 3D scene is necessary when designing a 3D system.

Visual Cues

Depth perception can be greatly improved by duplicating the visual cues used by the human visual system. Accommodation, convergence and binocular stereopsis are dependent on the physical characteristics of the eye such as muscle movement and relative positions between the two eyes. Pictorial cues, which are not dependent on the physical characteristics of the eye, provide strong depth cues and have been used for centuries by artists to achieve the illusion of depth in their paintings. These cues include interposition, light, shade, shadows, perspective and brightness.
Interposition

One of the strongest cues is interposition, in which solid objects obscure the objects behind them. This also includes the back side of objects not visible to an observer. These situations are easily handled by a computer with a variety of solutions to the hidden line and hidden surface problem. Wireframe is a presentation style where hidden lines are not removed (Figure 1a). A presentation style that removes hidden lines and surfaces is called a solid representation (Figure 1b).

Light, Shade and Shadows

Light, shade and shadows give clues to the relative size and positions of objects. Shade is found on object surfaces that do not receive direct light. They appear darker than the other surfaces (Figure 1c). Shadows are a result of an object blocking light from a different surface (Figure 1d). Shadows are calculated by projecting rays from imaginary light sources. Several light sources can be placed throughout the scene to provide helpful depth cues. One of the limitations of using shadows is that they are computationally expensive to generate. However, determining shade for appropriate surfaces is not as expensive.

Perspective

Linear perspective makes two objects separated by constant distance ‘appear’ closer together the farther away they are from the observer. Perspective is easily imitated by a
computer using mathematical relationships known about perspective. This cue is especially helpful when objects have many planar and parallel faces. Perspective does not seem to be helpful at all when dealing with objects like spheres (Brooks 1988).

**Brightness**

Relatively brighter objects appear closer than other objects. This cue was used in earlier computer systems with wireframe images. These systems used brighter lines to indicate an edge was close to the observer. Lines that were farther away from the observer were dimmer. This is incorporated into current lighting models by limiting the range of a particular light source. The farther away an object is from the light source the less light it will receive and the dimmer it will appear.

**Presentation Style**

Presentation style is a description of how objects are displayed and includes two components: format and look (Forrest 1986). The presentation format determines how the objects are viewed. Two formats are the most common. 1) Perspective is used to represent a realistic view of a scene with vanishing points. 2) Orthographic projections are typically used in a multiple window format. Typical views include the top, side and front of the scene. The designer must visualize the 3D scene by mentally combining the three views. The presentation look refers to the rendering technique used to display objects. Two formats are based on the previous visual cues. 1) Wireframe representations appear as stick figures and include all lines and surfaces of an object. 2) Shaded objects have hidden surfaces removed and visible faces are shaded by a simulated light source.

**Monocular Movement Parallax**

Movement parallax refers to the apparent relative motion of objects as our eyes move with respect to the objects or the objects move with respect to our eyes. Itelson (1960) states, "If one looks at a near object and moves his head left and right, the near object and some fixed point beyond it seem to move, relative to each other, in opposite directions, while two points at the same distance do not move relative to each other."
This cue seems particularly suited to the dynamic motion capabilities of computer graphics. Braunstein's (1976) conclusion from several experiments is that accurate projections of rigid motions in 3-space usually elicit reports of perceived depth. The cue is intensified when the user has complete control of an object. This allows the observers to visually 'walk around' the object to develop their own perception of the scene. A natural method to manipulate the scene could greatly improve depth perception. However, the scene must move smoothly and as expected, or the illusion is easily destroyed. The kinesthetic experience of moving an object with a physical device such as a mouse or dataglove seems to enhance the depth effect (Brooks 1988).

INTERACTION STYLES

Shneiderman (1987) describes five interaction styles that have been developed. These styles can be used independently of each other or can be combined for some applications. 1) *Menu selection* allows a user to select a command from a limited set of alternatives. This style is especially effective for novices because a variety of tasks can be initiated with little learning. 2) *Form fill-in* is most appropriate when data entry is required. Users move a cursor within a set of related fields and enter data as required. 3) *Command languages* are efficient for expert users because they can express a complex task without accessing a series of menus. Users typically type the commands on a keyboard. However, error rates are usually higher than menu selection. 4) *Natural language* is similar to command languages but there are fewer restrictions on the syntax of an expression. The computer is given the responsibility of deciphering the user's phrase that is often expressed by voice or text. Natural language interaction has had limited success so far. 5) *Direct manipulation* is effective when a visual representation of a task is available. Users perform tasks by pointing at objects and manipulating them. This an appropriate style for most 3D graphic tasks.

PREVIOUS 3D INTERACTION TECHNIQUES

An *interaction task* is the entry of a single piece of information by the user. An *interaction technique* is the method used to enter that piece of information. The next chapter
discusses 3D interaction tasks such as translation, rotation, and sizing. This section reviews previous techniques for performing these common 3D tasks. Early techniques were primarily used for 1D or 2D tasks (Newman 1968; Evans, Tanner, and Wein 1981; Buxton 1982). Additional techniques have been proposed for 3D tasks (Thornton 1979; Badler et al. 1986; Kaufman and Yagel 1989; Galyean and Hughes 1991; LeBlanc et al. 1991; Phillips 1991; Shoemake 1992).

Cutplane

The Cutplane provides effective 3D feedback when selecting 3D points (Edwards, Kessler, and Leifer 1988). Movement of the Cutplane is constrained to remain within a bounding reference room and parallel to the front of the room (Figure 2). This allows intuitive control of the Cutplane with a mouse or other locator device. A perspective projection and shadows from an overhead light source enhance depth perception.

Intersection of the plane and any object in the scene is highlighted, and only points in the highlighted area can be selected for manipulation. Point selection is performed with a 2D crosshair that moves within the Cutplane. Once the point has been selected it can be manipulated in a variety of ways to control scaling and translation.

Comment: Although this method provides improved visual feedback, selecting a 3D point is somewhat awkward because it takes two separate steps. In addition, a natural method to manipulate the point of view is not provided.

Triad Mouse

Nielsen and Olsen (1986) use a technique called a triad mouse to specify a 3D point. A 2D locator device is used to manipulate the 3D position of the triad within a frame of reference. They suggest several possibilities for the size and shape of the triad and conclude that it is best to keep the pointer simple to allow quick refresh time during dynamic interaction sequences.
The triad mouse movement is determined by repeatedly sampling the locator and using the direction of its most recent 2D movement as a basis for the 3D movement of the cursor (Figure 3). For example, forward and backward locator movements would cause cursor movement along the X axis. Horizontal locator movement will move the cursor along the Y axis and diagonal movements would cause a change along the z-axis.

They originally tried a weighted movement along all three axes but found this to be too difficult to control movement along a single axis. They modified this version to constrain movement along a single axis. Although this limits the ability to move freely through space, it improved control.

Translation of an object is constrained to move along a single axis determined by an edge of the object or within a plane that corresponds with a face of the object. In either method, a starting point and ending point are selected and the object is translated along the vector. Although direct manipulation versions of this method can be created, object movement is still unnecessarily constrained.

Similar methods are used for rotation and scaling. A point is selected on the object to determine the axis of rotation and the vector obtained from a second and third point determine the angle of rotation. Their intent was to provide direct manipulation techniques for 3D objects but they admit that there are many cases when the techniques are awkward to apply.
Comment: These interaction methods do not seem convenient and little attention is given to providing enhanced depth cues. They do not address point of view manipulation.

Skitters and Jacks

Bier (1986) refers to his 3D cursor as a skitter, which has a position and an orientation in 3-space. Specific 3D points can be referenced with the use of jacks, individual coordinate frames that act as 3D place holders. Jacks are used as start and end conditions for object transformations.

Skitters are moved through the scene in one of several modes. They can be moved along the face of an object by pointing at a perspective or parallel projection of the 3D scene using a 2D cursor controlled by a standard locator device (Figure 4). The desired face is determined by casting a ray from the viewer's eye through the image point of the 2D cursor. Since the ray can intersect with more than one face, separate modes are provided to indicate whether the intended face is the front face, back face, or perhaps on an obscured object. Another mode is used to move skitters along a line or plane in open space based on the orientation of a selected jack.

Comment: Bier's original intent was to decrease the number of tasks required to perform a simple transformation, but having to switch between several modes is a hindrance. It would be better to allow unrestricted movement of the skitter in all situations resulting in the need for only one mode.
Handle Box

A novel approach to 3D manipulation was applied to a space planning application (Houde 1992). Emphasis was placed on developing a design that would make 3D interaction more natural by drawing upon users' experiences with moving objects in the real world. The 3D tasks were limited to sliding along the floor, lifting, and turning about a vertical axis. This method was developed to allow smooth switching between tasks. A bounding box appears around a selected object. Narrative handles are iconic buttons placed directly on the box to provide control for each task (Figure 5). Handles are positioned on the box to indicate the task. For example, a handle is placed on top of the box to be used for lifting the box. Houde found that providing a composite mode for performing all three tasks allowed users to make repetitive adjustments more smoothly.

Comment: This technique appears intuitive for novice users to learn. However, the composite mode should allow a wider range of tasks before the merits of this system can be sufficiently evaluated.
3-Draw

While most systems currently use 2D input devices, some designers are experimenting with 3D input devices (Zimmerman et al. 1987; Ware and Jessome 1988; Ware and Osborne 1990). 3-Draw is a computer aided design tool targeted at the early concept-forming stages of design (Sachs, Stoops and Roberts 1989). Designers sketch their initial ideas in 3D while the computer displays the objects in perspective and with dynamic motion. The interface consists of two six degrees-of-freedom sensing devices. Designers use both hands to manipulate the two devices. One of the devices is used to control the position and orientation of the scene. The second device is in the form of a stylus and is used like a pencil to draw lines and curves in 3-space. Together, these tools enable a designer to draw and edit freeform curves used to create wireframe structures. While an object is being created, the designer can rotate and move the scene for dynamic visual feedback.

Comment: Interesting features can be provided with 6D devices but they are not a solution by themselves. Users have difficulty controlling all six degrees of freedom at the same time (Badler et al. 1986). Improved interaction techniques must be developed to take advantage of the additional degrees of input and still maintain control.

DESIGN GUIDELINES FOR USABILITY

Several strategies have been identified for designing successful user interfaces. These can be used as guidelines when developing new systems. Several guidelines have been proposed for general software development (Bewley et al. 1983; Card et al. 1983; Gould and Lewis 1983; Norman 1983; Hewett and Meadow 1986; Shneiderman 1986; Smith and Mosier 1986; Apple Computer 1987). However, there are relatively few guidelines associated with graphical interaction (Foley and Wallace 1974; Buxton 1985; Smith and Mosier 1986; Brooks 1988). As guidelines evolve, user interfaces will become more standard and easier to use.
General Guidelines

Gould and Lewis (1983) proposed four principles to design software that is easy to learn, easy to use, and pleasant to use. 1) Designers must clearly understand who the intended user is. This is done by studying their cognitive and behavioral characteristics as well as understanding the expected work to be accomplished. 2) A group of ‘typical’ users should be included in the early design stages of the product. 3) Users should use prototypes to perform realistic work and performance characteristics and subjective ratings should be measured. 4) Any problems that are discovered during testing should be fixed and the testing should be repeated as often as necessary.

Bewley et al. (1983) described several principles, derived from cognitive psychology, which were used during the design of Xerox’s 8010 “Star” office workstation. 1) There should be an explicit user model of the system that is both familiar and consistent. 2) Pointing to something is generally easier than remembering a name and typing it. 3) Similar actions should be performed consistently across domains. For example, deleting a word from text should be similar to deleting information from a database. 4) The screen should accurately show the state of the object the user is working on, “What you see is what you get.”

Guidelines for 3D Interaction

Concerning future 3D systems, Fred Brooks (1988) states, “Our interfaces must be direct-manipulation, not command-string; interactive, not batch; 3-D not 2-D; multisensory, not just visual.” Most of these goals require faster hardware, increased memory, and new input/output devices. Current workstations with dedicated graphics engines can be used to achieve some of these goals, but improved interaction techniques must be developed to take advantage of the hardware.

Dynamic actions should be specified with direct manipulation and interactive decisions should be expressed through a menu (Brooks 1988). Many systems require the user to select an operand followed by an action or vice versa. This sequence interrupts visual and tactile continuity. The solution is to have two cursors: an object cursor that moves continuously throughout the scene and a command cursor that moves discretely among the
commands. One implementation would be to use command keys for each potential command and use a locator device within the environment strictly for direct manipulation.

Another requirement of an effective 3D user interface is the ability to alter the point of view dynamically. Viewpoint specification requires six degrees-of-freedom (df): camera orientation (three df) and camera position (an additional three df). However, camera orientation is often constrained to be pointing towards a particular point such as the origin. This leaves only the position of the camera to be specified (three df).

Interaction Metaphors

Some successful interaction techniques are designed to be used as if real objects were being manipulated. These techniques are considered metaphors because they act 'like' more familiar objects or tasks. A successful metaphor allows the user to relate previous experiences to a new task. This speeds the learning process and makes the user more confident while performing a new task. The selection of a metaphor for each interface helps a designer to define appropriate issues and make consistent decisions about the look, feel and capabilities of the interface (Brooks 1988).

The desktop metaphor allows the user to be only concerned with moving objects (files) on the screen rather than using cumbersome commands to perform the same functions. The shopping cart metaphor (Brooks 1988) allows a client to 'walk through' a building proposed by an architect. This is accomplished using a treadmill with handlebars and a head mounted display. Clients walk on the treadmill and turn the handlebars as if they were pushing a shopping cart. The computer senses the movement of the treadmill and adjusts the point of view accordingly. Chen, Mountford, and Sellen (1988) used a virtual sphere metaphor to control object rotation. The user is to consider the object enclosed in a clear sphere. A locator device is used to rotate the object as if the sphere were a trackball. A stroke across the sphere, a circle in 2D, determines the angle and degree of rotation.
TEETERTABLE: A NEW METAPHOR FOR 3D INTERACTION

Consider a table top attached to a pedestal with a ball joint. The table top can swivel about its central axis as well as pivot about the ball joint depending on which edge of the table is being held. Lifting up or pushing down on a table edge causes the table to tilt (or teeter). The table top can be moved into virtually any orientation. Therefore, objects on the table can be viewed from any position by adjusting the table and keeping the eye position stationary.

Initial Design Goals

TeeterTable was developed for use during the evaluation of SEP(3D). It is a 3D system designed around four goals. 1) It should use direct manipulation techniques for all object manipulations. This includes a natural method to alter the point of view dynamically. 2) It should be easy to learn and allow quick and informal designs to evolve. This means that there is little need for exact dimensioning. 3) The 3D illusion must always be maintained within the virtual environment. This includes the use of shaded, solid objects in a single perspective image. 4) It should be a flexible research tool and allow different techniques to be implemented quickly for formal and informal comparisons.

3D Pointing

A 3D pointer is used to indicate a position in 3-space. It looks like two crosshairs connected by a vertical line (Figure 6). The lower crosshairs are the ‘feet’ of the pointer and the upper crosshairs are the pointer ‘hands’. It looks similar to Nielson and Olsen’s (1986) triad mouse. The cursor feet always remain attached to the table to provide a frame of reference while the cursor hands can be adjusted to different heights. Therefore, the pointer hands indicate the actual 3D position being referenced, while the pointer feet provide a constant reference to the table, or floor of the environment.

The mouse is used to control the 3D pointer feet just as an ordinary 2D pointer. This means that a left and right movement of the mouse translates to a left and right movement of the pointer on the screen, regardless of the orientation of the table. This maintains kinesthetic correspondence between the 3D pointer and the mouse. The height of the
pointer is controlled by a valuator. A knob is a natural device to use in the non-dominant hand and it allows continuous control of the pointer height with minimal hand movement. Therefore, the mouse is used strictly to position the pointer in the XZ plane while the knob is used to adjust the pointer's height.

Most systems use a 2D pointer to select an object. This simply requires placing the pointer anywhere on the object image. However, a 2D position on the screen can refer to an unlimited number of points in 3-space. This type of 2D pointing breaks the illusion of 3D because the designer is pointing to a 2D image.

TeeterTable objects are selected by positioning the pointer inside the volume of the object. This can be done because, unlike a 2D pointer, the TeeterTable 3D pointer always exists in a unique 3D position. This method strongly reinforces the 3D nature of the environment. However, this technique requires effective visual cues to indicate where the pointer is in 3-space.

![Figure 6 3D Pointer.](image1.jpg)  ![Figure 7 TeeterTable Modeling Environment.](image2.jpg)

**Mode Selection**

An iconic representation of the button box appears in the lower left corner of the screen. The collection of virtual buttons is used as a menu. Each icon has a label to indicate the task. Pressing the appropriate key on the button box highlights the associated menu item to indicate the current task. Each mouse button controls a separate action depending on which menu item is selected. A set of menu items is reserved for the right mouse button and a
A button box is used for menu selection to maintain the 3D illusion of the environment. If the mouse was used to select menu items, the 3D pointer must leave the virtual 3D environment to make a selection from a 2D menu. This breaks the 3D illusion of the environment. Using the button box to make menu selections allows the 3D pointer to remain in the virtual environment.

**Visual Cues**

The combination of a 3D pointer and a grid on the TeeterTable provides effective visual cues for selecting points in 3-space. Objects are drawn solid with shaded faces in a single perspective view. Drop shadows are also used to indicate when objects are above the table. The shadows are projected from an imaginary light source directly above the scene. Therefore, shadows only appear on the table when an object is above the table. All of these cues add to the realism of the scene (Figure 7).

TeeterTable allows for simple manipulation of these visual cues. For example, object colors can be altered, the view can be switched from perspective to orthographic, and objects can be displayed in wireframe or solid.
Figure 8  Rotating the table counterclockwise (a) before, (b) during, (c) after.

Figure 9  Tilting the table upwards (a) before, (b) during, (c) after.

View Manipulation

View manipulation is controlled with the left mouse button. A point on the table is selected by positioning the pointer and pressing the left mouse button. A left or right movement of the mouse maps to a clockwise or counterclockwise rotation around a vertical axis passing through the table's center (Figure 8). A forward movement tilts the table upward and a backward movement tilts the table downward (Figure 9). The tilt axis is perpendicular to the line between the table center and the selected point.
Translation

Move mode is selected from the menu and controlled with the right mouse button. Objects are moved by selecting an object and holding down the right button. While selected, pointer movement directly corresponds to object movement. This is equivalent to dragging the object in 3D. Objects are moved along the XY grid with the mouse and along the Z axis by turning the knob (Figure 10).

Sizing

Size mode is selected from the menu and controlled with the right mouse button. Objects are sized by selecting a point or face on the object. While a face is selected, it can be dragged in 3-space just like any other object. This allows sizing along a single axis if desired or shearing an object if the face is moved along a different axis (Figure 11). While a point (or vertex) is selected, it can also be dragged about in 3-space. All faces that share the vertex are sized. However, face movement is constrained to maintain the integrity of the object (Figure 12). Selecting a point allows multiple axis sizing. These techniques can also be used with curved surfaces by manipulating points on a control grid.

Figure 10 Moving an object (a) selecting object, (b) moving along XY grid, (c) moving in vertical direction above the grid.
Figure 11  Sizing an object by using a face (a) selecting face, (b) stretching along a perpendicular axis, (c) shear object by stretching along a different axis.

Figure 12  Sizing an object by using a vertex (a) selecting the vertex, (b) increasing width and depth, (c) increasing height.

Rotation

*Rotate mode* is selected from the menu and controlled with the right mouse button. Object rotation can be done about any axis that passes through the object centroid. This is convenient for most rotations because it relieves the user from having to select a point of rotation. If necessary, an additional method could be provided to select a different rotation point.
Object rotation was designed to behave similarly to view manipulation. The position of the pointer with respect to the object's centroid determines the horizontal axis of rotation. The vertical axis is always perpendicular to the XY plane. These two axes allow an object to be rotated into any orientation (Figure 13).

Object Creation

Create mode is selected from the menu and performed with the right mouse button. The 3D pointer is used to define volumes. For example, the corner of a cube is positioned with a button press and then the opposite corner of the cube is positioned while the button remains pressed. The image is continuously updated as the pointer is moved to give the appearance of a rubber band cube. Releasing the button completes the 3D volume (Figure 14). This allows creation of any primitive that can be enclosed by a bounding box such as a sphere, cylinder, torus, or pyramid. It also provides visual feedback. A similar technique could be used to extrude a 2D object along a 3D path to create complex objects.

Figure 13 Object rotation (a) selecting object, (b) rotating about vertical axis, (c) rotating about horizontal axis.
Versatility of TeeterTable

One of TeeterTable's strengths as a research tool is that a variety of techniques can be implemented or modified for comparison. Commercial systems do not allow this flexibility without having access to the source code. This flexibility allows specific questions or theories to be evaluated. For example, is a wireframe representation of a scene as effective as a shaded one? Is a mouse more effective than a tablet for a particular task? Would a different background color effect the overall performance of a designer? Is a single mode that allows several tasks to be performed better than a collection of separate modes?
CHAPTER III
EVALUATION PLAN

An evaluation plan describes what to evaluate and why. It also provides the step-by-step procedures necessary to accomplish the plan (Spencer 1985). A standardized plan is designed to evaluate or compare a wide variety of products that share certain capabilities. A standardized evaluation plan for the usability of 3D interfaces, SEP(3D), was developed to combine traditional empirical design with a user satisfaction questionnaire. It is designed to be simple enough that anyone can make comparisons of 3D interfaces without special equipment or experience.

This chapter presents the details of SEP(3D). The following sections describe the suggested experimental methods for evaluating interfaces with SEP(3D): objective, subjects, task analysis, stimuli, environment, usability measures, covariate measures, procedure, and confounding factors.

OBJECTIVE

Many 3D systems are used exclusively to prepare precise engineering drawings. Designers often use paper and pencil to capture their initial ideas because existing computer aided design systems do not offer the freedom and flexibility of a pencil (Sachs, Stoops and Roberts 1989). The objective of SEP(3D) is to evaluate the usability of interfaces for performing quick and informal manipulations early in the design process. A standardized evaluation plan would help improve interfaces by identifying successful techniques for further study and by identifying poor techniques to be discarded. SEP(3D) can be used in an iterative design process. Problems can be identified in early evaluations and after corrections have been made, additional evaluations can confirm or reject the effects of the
changes. Comparisons can also be made among commercial systems to identify their strengths and weaknesses.

SEP(3D) evaluates the interface of a 3D system and all other aspects of the system are ignored. Factors such as functionality, cost, help facilities, and system support are important when making a comprehensive comparison of systems, but are beyond the scope of this study.

Intended users of these interfaces are people interested in manipulating 3D objects. This includes professionals such as engineers, architects and other designers. However, new applications would evolve for a wider scope of users if a 3D interface was easier to use.

SUBJECTS

Between-subjects experimental designs assign different subjects to each treatment. This insures there is no confounding influence due to a subject performing the same task under different circumstances. However, a significant cost is incurred because more subjects are required. Within-subjects designs assign the same group of subjects to each of the treatments. This allows direct comparisons of scores for each subject. Fewer subjects are required but they are needed for longer times and results can be confounded due to order and carry-over effects. For example, if subjects perform a task under three different situations, the first performance may affect the second or third performance. One way to protect against confounded results is to have subjects perform the tasks in counter-balanced orders.

The minimum number of subjects that can be used for a single treatment and still obtain reliable data is four (Roberts 1980). Therefore, four expert subjects for each system are recommended for SEP(3D) evaluations. A between-subjects experimental design is used to reduce potential transfer effects for subjects who have experience with more than one system. Although this requires a larger number of subjects, validity of the results will be higher because subject performance is not influenced by previous experience with other 3D systems.
Expert use is often defined by several months experience with a system. However, this is not always a reliable indication of expertise. An additional indication of expert performance is that subjects are able to perform the tasks at a steady rate with few significant errors. Subjects who do not meet these criteria by a third trial should not be included in a study of expert performance.

TASK ANALYSIS

A set of tasks common to many 3D systems and appropriate for evaluation must be developed. A tradeoff exists between making the collection of tasks too narrow, which limits the scope of an evaluation, and having too many tasks, which eliminates some systems from comparison that do not allow all the tasks.

Definitions

An interaction task is the entry of a single piece of information by the user. There are five basic interaction tasks (BIT) for 3D data (Foley et al. 1990). 1) The position task results in an X, Y, Z position in 3-space. 2) The selection task yields an object identification. 3) The text task yields a character string. 4) The quantify task provides a numerical value. 5) The orient task yields an orientation in 3-space.

An interaction technique is a method to enter information into the computer. Many different interaction techniques can be used for a specific interaction task. For example, a mouse can be used to point to a 2D position or a keyboard can be used to enter coordinates of a 2D position.

Axes

In traditional computer graphics, the X and Y axes refer to the horizontal and vertical directions, respectively. The Z axis is used to indicate a depth into the screen. However, this orientation is not natural to most users (Mountford et al. 1986). Many 3D systems, including the ones evaluated in this study, use the Y axis to indicate depth into the screen and the Z axis as the vertical axis. This allows the user to consider the XY plane as the floor of the environment and the Z axis as a reference to the height above the floor. These
distinctions are made only for the purposes of this discussion. Subjects are not aware of the differences during task performance because specific axes are not mentioned in SEP(3D) tasks. Therefore, any system can be evaluated regardless of its axes orientations.

Core Tasks

A variety of applications make use of interactive 3D graphics. Animators manipulate objects in a scene, architects and engineers design structures, and scientists visualize multidimensional data. However, all of these applications share a core set of tasks that are performed during quick and informal 3D design.

A composite interaction task (CIT) is a collection, or series, of basic interaction tasks (Foley et al. 1990). There are several CITs that form the core tasks in a 3D graphics system: translation, rotation, sizing, point of view manipulation, and object construction. Several studies, attempting to improve 3D design systems, have recognized the importance of these tasks (Mountford et al. 1986; Nielson and Olsen 1986; Bier 1986; Chen et al. 1988; Ware and Osborne 1990). Complex tasks are a collection of these simple core tasks. For example, designing an engine is merely a collection of translation, rotation, and creation tasks performed on primitive objects.

SEP(3D) Tasks

Some of the core tasks can occur along any combination of axes. Action along each axis can be considered a separate sub-task. Seven unique sub-tasks can be performed along each combination of axes: X, Y, Z, XY, XZ, YZ, and XYZ. Although it is important to represent all sub-tasks, a compact set of tasks makes an evaluation more manageable. Many of the sub-tasks are performed similarly to each other and can be represented by an appropriately selected sub-task. A summary of sub-tasks selected to represent each of the core tasks in SEP(3D) is provided in Figure 15.
Possible Task | SEP(3D) Task
--- | ---
Move X | Move XY
Move Y | Move XY
Move XY | Move XYZ
Move Z | Move XYZ
Move XZ | Move XYZ
Move YZ | Move XYZ
Move XYZ | Move XYZ
Rotate X | Rotate H
Rotate Y | Rotate H
Rotate Z | Rotate H
Rotate XY | Rotate V
Rotate XZ | Rotate V
Rotate YZ | Rotate V
Rotate XYZ | Rotate V

Figure 15 SEP(3D) tasks selected to represent all possible tasks.

Translation

Translation is the task of moving an object from one position to another position. It is a sequence of selection (of the object) and position (the new position of the object). Translation along the XY plane is performed similarly in many systems since it is a 2D task. Sub-task Move XY was selected to represent movement along X, Y, and XY. Translation along the Z axis is handled quite differently among systems. For this reason, sub-tasks that include movement along Z are grouped together. Sub-task Move XYZ was selected to represent movement along Z, XZ, YZ, and XYZ.

Sizing

Sizing is the task of changing the size of an object along one or more dimensions. It is a sequence of selection (of the object or face) and position (the position of the faces). Sizing can occur along any combination of axes. It is similar to translation concerning movement along each axis. Therefore, sub-task Size XY was selected to represent sizing along X, Y, and XY. Sizing along Z is often handled differently than XY. Therefore, sub-task Size XYZ was selected to represent sizing along Z, XZ, YZ, and XYZ.
Rotation

Rotation involves changing the orientation of an object about an axis of rotation. It is a sequence of selection (of the object), position (of the center of rotation), and orientation (with respect to a coordinate frame). Rotation is one of the most complex tasks because it requires selecting a center of rotation, an axis of rotation, and an angle of rotation. As with other tasks, some rotations are typically easier to perform than others. Mountford et al. (1986) found that most rotation tasks are performed about independent axes (X, Y, or Z) because the behavior of combined axes is difficult to understand visually and is not familiar to the user. Therefore, SEP(3D) does not include rotation tasks about arbitrary axes. This also expands the scope of SEP(3D) by allowing systems to be included in evaluations that may not allow rotations about arbitrary axes.

Rotation about the vertical axis (Z) is probably the most intuitive for users. Sub-task Rotate V was selected to represent vertical rotations about Z. Rotations about horizontal axes (X or Y) are usually more involved but are performed similarly to each other. Sub-task Rotate H was selected to represent horizontal rotations about X and Y.

View Manipulation

Point of view manipulation determines the current view of the scene. It can also be considered as mapping the 3D scene to the 2D screen. It is a sequence of selection (of the virtual camera), position (of the virtual camera), orientation (of the virtual camera), and an additional task of specifying the field of view. SEP(3D) places two limitations on view manipulation to allow for systems that do not allow unrestricted view manipulation. The origin of the scene is always the center of interest and the field of view is held constant. These assumptions reduce the otherwise complex task of view manipulation to positioning the virtual camera in 3-space.

Object Creation

SEP(3D) includes four Create tasks for cubes. A cube is used because it is a primitive object that consists of only a few vertices and faces. This is important when comparing computers with different processing power. Tasks are expected to be performed in real
time and a more complex shape, such as a sphere, would give an advantage to high-end workstations over smaller personal computers because of their faster rendering capability.

**STIMULI**

An instruction booklet (Appendix A) is used as the stimulus for SEP(3D). The first three pages describe the overall evaluation process and what is expected of subjects. A series of pictures guides subjects through twenty 3D interaction tasks. The booklet can be used for any system because instructions are intentionally generic.

**Task Descriptions**

Roberts and Moran (1983) suggest that the overall consistency of how well different techniques work together may be more important than any individual techniques in determining the quality of an interface. Subjects perform a series of twenty tasks: four Create, two Move XY, two Move XYZ, two Size XY, two Size XYZ, two Rotate H, two Rotate V, and four View. The task sequence is random and designed to represent a typical design session. Each task builds upon the previous task.

All subjects use the same task sequence. A single sequence allows fewer subjects to be used because with multiple sequences, a minimum number of subjects would be needed for each unique sequence. Since each task is presented at least twice in a random sequence, order effects for any particular task are reduced.

Each task is described on a separate page. Two pictures appear below each instruction to indicate how the scene should appear before and after completing the task. The instruction describes the task and the appropriate object is highlighted to indicate which object should be manipulated (Figure 16). However, highlighting is not reproduced in the figure or Appendix.

**Evolution of the Instructions**

The instruction booklet went through several changes during the design of SEP(3D). Initial task descriptions used only one picture to indicate how the scene should appear after the task. However, subjects would often turn to the previous page to orient themselves to
the new picture. This was an annoyance to the subjects and increased their performance time. Providing the before and after pictures on the same page eliminated this problem.

Wording of the instructions was also changed. The first version of instructions attempted to describe each object. Even in a simple scene of only four cubes, it quickly became confusing to the subject. Instructions like 'move the tall, thin cube to the left of the medium sized cube' were typical. Highlighting the appropriate object in both pictures allowed the instruction to be rewritten, 'move the object.' The subject can clearly tell from the picture which object to move and to what new position. References to individual axes were also eliminated due to the confusion. Instructions such as, 'rotate the object about the Z-axis' were changed to, 'rotate the object about the vertical axis.'

ENVIRONMENT

Evaluation sessions are conducted where subjects typically use their system. One reason for this is the difficulty of obtaining different computers in a common room. Also, this best replicates the environment a user would be exposed to when using the software. A video camera is used throughout each session to record all trials and an evaluator is present to answer questions and guide subjects through the process.
Task #6

Move the object.

Before

![Before diagram]

After

![After diagram]

Figure 16 Sample task instruction.
**USABILITY MEASURES**

Two measures are taken to evaluate the usability of a technique: performance time and user satisfaction. These measures provide both objective and subjective results. They are common measures found in the literature to estimate usability.

**Performance Time**

Since evaluations are made across different platforms, a simple method is not available for automatically recording task completion times. Therefore, time to complete a task is measured with a stopwatch by the evaluator during review of the video tape. During a pilot study, subjects were told to read the instruction aloud before beginning each task and say 'stop' when they completed the task. Timing began after the instruction was read and ended when the subject said 'stop'.

It became clear that this was not a reliable method for recording true performance time. After reading the instruction, subjects would occasionally pause to form a plan of action. This initial pause should not be included in performance time because it varies between subjects. A larger error for estimating performance time occurred when determining the appropriate ending time. Occasionally a subject would be too concerned about making a perfect match and spend extra time making fine adjustments to an object, termed *matching delay*. Since the intent of these tasks is to simulate a typical design session, and not test ‘matching’ ability, matching delay should not be included in performance time.

**Suggested Timing Method**

Evaluators should begin timing for each task after the subject has read the instruction aloud and begins computer activity. Signs of activity include moving the mouse, typing at the keyboard, or clicking the mouse button. A directional microphone is useful for recording keyboard or mouse button clicks. Timing ends when the task has been completed. An experienced evaluator can determine when a task has been completed and when matching delay begins. The reliability and validity of this timing method are thoroughly discussed in the final chapter.

Matching delay can be minimized by reminding subjects during practice trials that a perfect match is not necessary and that the intent is to imagine they are performing an
impromptu design. Subjects become familiar with what is expected and when they are making unnecessary adjustments. By the third trial, very few of the task times need adjustment for matching delay.

User Satisfaction

User satisfaction is evaluated with a subject questionnaire (Appendix B). Forty-five items are used to determine the overall satisfaction with the system, individual features of the system, and the computer experience of the subject. Responses are given on a scale of 0 through 7. Each end of the scale is anchored with appropriate bipolar adjectives. Eight items are included for each task category (Figure 17).

Item one is intended to evaluate the overall usability of the interface for that particular task. Items two and three are intended to reveal the user’s perception of how easy the system was to learn. These results should obviously be reviewed cautiously because an individual’s memory of the learning process could be biased. However, it does provide a worthwhile insight at a minimum cost.

An important aspect of usability is the ease with which errors can be corrected. Item four is designed to reveal this aspect of the interface. Items five and six evaluate the input device and visual presentation, respectively. Item seven is included to confirm that the SEP(3D) task is appropriate for the system. A low rating by all subjects for this item may indicate that a system was not designed to perform this particular task and results may not be comparable to other systems that were designed for the task. Finally, an open-ended question encourages subjects to report any additional information that they feel relevant.
Translation

Translation is the task of moving an object from one position to another. The following statements apply only to the task of translation with this system.

1. Moving an object
   - difficult
   - easy
   - 0 1 2 3 4 5 6 7

2. When you first learned the system, objects moved in the direction you expected
   - never
   - always
   - 0 1 2 3 4 5 6 7

3. Now that you are familiar with the system, objects move in the direction you expect
   - never
   - always
   - 0 1 2 3 4 5 6 7

4. Correcting your mistakes
   - difficult
   - easy
   - 0 1 2 3 4 5 6 7

5. Input device is appropriate for the task
   - never
   - always
   - 0 1 2 3 4 5 6 7

6. Visual feedback
   - confusing
   - very clear
   - 0 1 2 3 4 5 6 7

7. Moving an object is an appropriate test (common task) of the system
   - inappropriate
   - very appropriate
   - 0 1 2 3 4 5 6 7

8. Are there any comments (good or bad) you wish to add regarding the task of translation? Continue on back if necessary.
COVARIATE MEASURES

One way to increase reliability with a limited number of subjects is to reduce the between-user variability. This could be done by giving subjects a pretest that effectively measures ability associated with the tasks. Subjects who score within a certain range would be used for the experiment. An alternate method is to use all subjects, but use the pretest scores to ‘adjust’ the overall results.

A covariate measure is used to adjust the results of the measure of interest. This is done to (1) increase the precision of the experiment and (2) to remove potential sources of bias in the experiment (Winer 1971). For example, suppose that the purpose of an experiment is to determine the effect of a particular training method upon the ability to high jump. The height of the athlete may play an important role in the ability to jump high. If the athletes were of a wide range of heights, the results would be confounded. If the athletes’ heights were ‘taken into account’, the results would be more reliable and subject to less variation.

Three covariate measures were recorded during the SEP(3D) evaluations. 1) Spatial visualization skills were measured with a test. 2) Simple reaction time to a stimulus on the screen was recorded by a computer. 3) Pointing time using a mouse was also recorded by a computer. Procedures for collecting each covariate measure are outlined in the following sections.

Spatial Visualization Abilities

Spatial visualization is an ability to manipulate 3D images in one’s mind (McGee 1982). A pencil-and-paper test was developed to measure spatial visualization abilities (Vandenberg and Kuse 1979). The Mental Rotation Test requires a subject to match the picture of an object with one from a set of dissimilar objects. The only difference between the original object and the selected object is that they are presented at different orientations about a vertical axis. The test has been used in many studies and there have been clear indications of internal consistency and test-retest reliability (McGee 1982). The test consists of two parts with a maximum score of twenty for each part. Subjects are limited to three minutes for completing each part.
Vicente and Williges (1988) discovered that subjects with low spatial ability often got lost in a hierarchical file structure and therefore performed worse than subjects with higher spatial ability. It is expected that the mental rotation skills of an individual could strongly influence performance time of 3D tasks. If the mental skills of the subjects vary greatly, the performance times may also vary. This reduces the reliability of the data. Factoring out the variance between subjects due to different spatial skills will better indicate performance time.

Reaction Times

Simple reaction time is also considered a potential covariate measure and is taken using a simple computer program. Subjects are instructed to press the mouse button as soon as they see a black dot appear on the center of the screen. The reaction time is recorded by the computer. Three practice trials are completed to familiarize subjects with the process followed by ten test trials. The mean of all test trials is used as a covariate measure.

Pointing Times

Pointing time, or target acquisition time, is also considered a potential covariate measure and is also taken with the use of a simple computer program. Before each trial, subjects place the pointer in the center of the screen. Subjects then move the pointer to a black dot as soon as it appears and press the mouse button. Distance from the pointer to the target is constant but the target position is random. Pointing time is recorded by the computer. Times are not recorded when the subject does not correctly position the pointer. Three practice trials are completed to familiarize the subject with the process followed by ten test trials. The mean of all test trials is used as a covariate measure.

PROCEDURE

Subjects are briefed on what is expected of them and why they are being asked to evaluate the system. The session begins with administering a reaction time test and pointing time test. The Mental Rotation Test is also completed with pencil and paper.
No instructions regarding the system are necessary because subjects are already familiar with the system. Several trials are recommended. The number of trials depends on the experience of the subjects. If novices are learning a system, the number of recommended trials depends on how difficult the system is to learn. Two experiments are discussed in the following chapters that evaluate the costs and benefits of the number of trials.

The questionnaire is completed after finishing all trials. Subjects are encouraged to provide any additional information that they feel is useful. Total time required of each subject is approximately one and a half hours.

CONFOUNDING FACTORS

There are several potentially confounding factors that were considered when developing SEP(3D). Attempts were made to either control these factors or at least minimize their effects.

Different Computers

Comparing software on different computers can provide misleading results. The most obvious difference is the computer's processing speed. If one computer is significantly faster than another, most tasks will be performed more quickly on the faster system. However, if the task is simple enough that processing power does not greatly affect completion time, then the difference between the computers is minimized.

Objects used in SEP(3D) are kept simple so that all computers can easily display them in real time. For personal computers, objects must be displayed in wireframe. However, more advanced workstations can often provide solid, shaded objects. Although advantages due to one aspect of the interface, superior visual cues, are identified, different processor speeds do not otherwise affect performance times.

Different Subjects

Subject expertise on a system is difficult to evaluate. Measures such as the number of months of experience can often be misleading. Keeping the tasks simple helps insure all
subjects can adequately perform the tasks. Informal observation of a subject’s capabilities is often sufficient to determine if the subject had adequate experience.

**Different Techniques**

Commercial systems sometimes provide several different ways to perform the same task. Some of these redundant methods are short cuts that only advanced users take advantage of while some are provided merely to give the user choices. Subjects on the same system should use the same techniques. Using significantly different techniques within the same system would confound the results. For example, subjects may have the option to type each command on a keyboard or select each command from a menu. These different interaction styles could significantly affect performance time.

Evaluators are responsible for recognizing potential differences and informing subjects which technique they should use during the evaluation session. If subjects are unfamiliar with a technique, or strongly prefer not to use it, then their results should be discarded.
CHAPTER IV
METHOD

Two experiments were performed to compare 3D interfaces and to show the reliability and validity of SEP(3D). Experiment One compared eight interfaces including commercial systems and several versions of TeeterTable. Experiment Two compared two versions of TeeterTable. The methods for both experiments were similar and are described in this chapter.

SUBJECTS
Thirty-two unpaid volunteer subjects were used for Experiment One (four expert users for each system). Four unpaid volunteer subjects were used for Experiment Two. Expert users were identified for the commercial systems who had at least three months experience with the application. However, experience was not a reliable indication of expertise. For example, several volunteers claimed to be familiar with the 3D aspects of a particular system but were unable to perform the tasks adequately. These results were discarded and additional volunteers were found.

Expert users, based on the requirement of several months experience, were not available for TeeterTable because it is an experimental environment. However, it has a limited number of features and a novice user can become proficient with the core tasks within a few minutes of practice. Therefore, they are considered experts for this study.

STIMULI
The SEP(3D) instruction booklet was used as stimuli for both experiments to guide subjects through the tasks. The same booklet was used for all evaluations because the instructions were system independent.
GENERAL DESCRIPTIONS OF THE INTERFACES

Three commercial applications were evaluated in Experiment One: Chalk, AutoCAD® Release 10 (Autodesk 1989) and Form*Z (autodesk 1990). Five versions of TeeterTable were also evaluated to determine the effects of small changes to an interface: Original, MultiMode, Triad, Wireframe, and Modified. Two additional versions of TeeterTable were evaluated in Experiment Two to evaluate the effect of constraints: NewRotate and NewSize. Each interface is briefly described for interaction style, presentation style, pointer type and input devices.

Form*Z

Form*Z version 1.04 features a highly interactive graphic interface that allows the user to work directly in 3-space, through parallel or perspective views (autodesk 1990). It is available for the Macintosh computer. Objects are created, edited and manipulated relative to one of the XYZ planes or to any user defined plane position in 3-space. Transformations can be applied to points, lines, faces, objects or groups of objects. A variety of display options allow the designer to visualize scenes in wireframe, solid or shaded formats. For this study, subjects used the wireframe format because solid or shaded formats are not possible for real time manipulation. Graphic control of perspective views allows the viewing position to be changed quickly by interactively rotating the scene. Several modes and options are provided for each task. A mouse is used to manipulate objects and make menu selections.

AutoCAD

Subjects used an IBM PC for this system. AutoCAD is a general purpose Computer-Aided Design/Drafting application (Autodesk 1989). It is largely command based but also provides menus to select commands. Most commands have several parameters that must be specified on a keyboard before the task is performed. These parameters typically include a location, size, or rotation angle. Objects are displayed in wireframe and in a multiple window format. Several views can be displayed at the same time such as the top, side, and front view. For this study, subjects used two windows. One window was for a
perspective view of the scene. A second window was a parallel projection. Few of the
tasks can be performed well in a 3D scene. Object location, size, and orientation can easily
be altered in a parallel projection. A camera and target metaphor is used to alter the
perspective view dynamically.

Chalk

Chalk is an extensive animation system developed at The Ohio State University. It
allows creation of elaborate scripts to control an animation. A small portion of the system
is used for scene composition and object manipulation. It is implemented on a Sun
workstation.

On-screen buttons are used to map vertical mouse movement to each of the axes. Any
combination of buttons can be selected. For example, selecting X and Y axes causes
vertical mouse motion to control movement along the X and Y axes. An additional set of
three buttons is used to map horizontal mouse movement to each of the axes. For example,
movement along the X axis can be mapped to horizontal mouse motion while movement
along the Y axis is mapped to vertical mouse motion. Therefore, diagonal mouse motion
moves objects along the XY plane. Motion of the object is controlled by motion of the
mouse in a separate window, called the chalkboard. View manipulation is controlled by
typing the XYZ eye coordinates. The point of view is assumed to be the origin.

TeeterTable

Several versions of TeeterTable were evaluated in this study to determine if SEP(3D)
can be used to identify highly controlled differences among interfaces. Common features
are described in this section and individual features are described separately. The first
version of TeeterTable was called Original. Three versions evolved from Original to
evaluate specific interface questions: Wireframe, Triad, and MultiMode. After evaluating
the first four versions, three additional versions were created to evaluate the effect of
constraints for performing certain tasks: Modified, NewRotate, and NewSize. The
TeeterTable genealogy is presented in Figure 18.
Figure 18 TeeterTable Genealogy.

TeeterTable is implemented on a Hewlett Packard color workstation. A mouse is used to control movement of the 3D pointer along the XY plane and a knob is used to adjust the height (Z axis) of the pointer. Visual feedback is provided with solid, shaded objects and drop shadows are used to indicate an object is above the XY grid. Color is only used for the pointer and highlighted objects, which are outlined in red. Menu items are displayed on the screen and a button box is used to switch between active modes. Two modes, one for each mouse button, are simultaneously active.

Cubes are created by dragging out the base while the mouse button is held down and applying a height by adjusting the knob. View manipulation is controlled with the left mouse button. With the button pressed, a horizontal mouse movement causes the scene to spin about a vertical axis. Vertical mouse movement causes the scene to tilt about an axis determined by the current pointer position.

Original

Original refers to the first version of TeeterTable. All tasks are performed in one of three modes with direct manipulation techniques. 1) View manipulation is performed while holding down the left mouse button. 2) In create/modify mode, objects are created, moved and sized while holding down the right mouse button. This composite mode was provided
to ease transitions between the most common tasks. 3) In rotate mode, objects are rotated while holding down the right mouse button.

**Wireframe**

Wireframe shares all the techniques and modes of Original but has fewer visual cues. Objects are displayed in wireframe format and drop shadows are not provided. Most commercial packages for personal computers use wireframe representations for real time manipulations due to limited processing power.

Comment: *Wireframe was evaluated to determine the effects of limited visual cues.*

**Triad**

Triad shares the techniques and modes of Original but uses a different technique to control the 3D pointer. The technique allows a mouse to be used exclusively to control the 3D position of the pointer and was implemented similarly to Nielson and Olsen's triad mouse (1986).

Comment: *Triad was evaluated to determine the effect of using a single input device, instead of two, to control the 3D pointer.*

**MultiMode**

MultiMode provides the same capabilities and visual cues of Original but with a larger number of modes. Five modes are used instead of three. 1) View manipulation is performed while holding down the left mouse button. 2) In create mode, objects are created while holding down the right mouse button. 3) In rotate mode, objects are rotated while the right mouse button is pressed. 4) Objects are modified in size mode while holding down the right mouse button. 5) In move mode, objects are translated while holding down the right mouse button.

Comment: *MultiMode was evaluated to determine the effects of separate modes for each task instead of a single composite mode.*
Modified

Five changes were made to TeeterTable based on the initial evaluation of Original, Wireframe, Triad and MultiMode. 1) The pointer was constrained to the table top. Previous versions allowed the pointer to be moved anywhere in 3-space and subjects sometimes 'lost' the pointer. 2) The MultiMode use of individual modes for each task was used because subjects made fewer errors and performance times were slightly faster than for Original. 3) Color was added to the scene to improve visual contrast. A blue background, white table and green objects were used. 4) The ability to tilt the table was removed to test the usability effect of constrained movement. Horizontal mouse movement caused the table to rotate about the vertical axis. Vertical mouse movement had no effect. However, all SEP(3D) View tasks could still be performed correctly. 5) Visual cues for rotating an object were modified. Previous subjects were not able to clearly 'see' which axes were selected. The previous table image used to indicate the rotation axes were replaced with the actual vertical and horizontal axes (Figure 19).

Comment: Modified was evaluated to determine if SEP(3D) is useful during an iterative design process.

Figure 19 Visual cues for rotation: (a) original method indicated a plane around the object to be controlled in the same way as the TeeterTable, (b) modified method clearly indicates the vertical and horizontal rotation axes.
NewRotate

Many of the changes developed for Modified were also included in NewRotate. However, the constrained view manipulation was returned to the Original method and a new constraint was added to rotation. SEP(3D) rotation tasks specify an axis to be perpendicular to a face. In previous TeeterTable versions, users had to position the pointer in the center of the face carefully to select a rotation axis exactly perpendicular to that face. NewRotate allows users to select any point on the face and the rotation axis is automatically set to be perpendicular to the face. This method restricts users from selecting an arbitrary axis but increases the convenience of selecting a perpendicular axis.

Comment: NewRotate was evaluated in Experiment Two to determine if constraints would improve object rotation.

NewSize

Many of the changes developed for Modified were also included in NewSize. However, the constrained view manipulation and the visual cues for object rotation were returned to the Original method. In previous versions, subjects could select a single face to manipulate the size or a vertex could be selected to adjust three faces simultaneously. NewSize only provided the second method. Therefore, all sizing operations were performed by selecting a vertex. This change forced subjects to perform Size XYZ in a single action rather than three separate actions.

Comment: NewSize was evaluated in Experiment Two to determine if constraints would improve sizing operations.

Summary of Differences

Figure 20 summarizes the differences between the interfaces for several dimensions: presentation style, pointer, interaction style, input devices, and hardware.
<table>
<thead>
<tr>
<th>System</th>
<th>Presentation Style</th>
<th>Pointer Interaction Style</th>
<th>Input Devices</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCAD</td>
<td>Wireframe</td>
<td>2D</td>
<td>Mouse</td>
<td>IBM PC</td>
</tr>
<tr>
<td></td>
<td>Multiple Orthographic Perspective</td>
<td>Command Direct Menu</td>
<td>Keyboard</td>
<td></td>
</tr>
<tr>
<td>Chalk</td>
<td>Wireframe</td>
<td>2D</td>
<td>Mouse</td>
<td>Sun Workstation</td>
</tr>
<tr>
<td></td>
<td>Single Orthographic</td>
<td>Command Direct Menu</td>
<td>Keyboard</td>
<td></td>
</tr>
<tr>
<td>Form*Z</td>
<td>Wireframe</td>
<td>2D</td>
<td>Direct</td>
<td>Macintosh</td>
</tr>
<tr>
<td></td>
<td>Perspective</td>
<td>Menu</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2D Orthographic</td>
<td></td>
<td>Keyboard</td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>Shaded</td>
<td>3D</td>
<td>Direct</td>
<td>HP Workstation</td>
</tr>
<tr>
<td></td>
<td>Single Perspective</td>
<td>Menu</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Knob Box</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Button Box</td>
<td></td>
</tr>
<tr>
<td>Wireframe</td>
<td>Wireframe</td>
<td>3D</td>
<td>Direct</td>
<td>HP Workstation</td>
</tr>
<tr>
<td></td>
<td>Single Perspective</td>
<td>Menu</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Knob Box</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Button Box</td>
<td></td>
</tr>
<tr>
<td>Triad</td>
<td>Shaded</td>
<td>3D</td>
<td>Direct</td>
<td>HP Workstation</td>
</tr>
<tr>
<td></td>
<td>Single Perspective</td>
<td>Menu</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Button Box</td>
<td></td>
</tr>
<tr>
<td>MultiMode</td>
<td>Shaded</td>
<td>3D</td>
<td>Direct</td>
<td>HP Workstation</td>
</tr>
<tr>
<td></td>
<td>Single Perspective</td>
<td>Menu</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Knob Box</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Button Box</td>
<td></td>
</tr>
<tr>
<td>Modified</td>
<td>Shaded</td>
<td>3D</td>
<td>Direct</td>
<td>HP Workstation</td>
</tr>
<tr>
<td></td>
<td>Color</td>
<td>Menu</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single Perspective</td>
<td></td>
<td>Knob Box</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Button Box</td>
<td></td>
</tr>
<tr>
<td>NewRotate</td>
<td>Shaded</td>
<td>3D</td>
<td>Direct</td>
<td>HP Workstation</td>
</tr>
<tr>
<td></td>
<td>Color</td>
<td>Menu</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single Perspective</td>
<td></td>
<td>Knob Box</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Button Box</td>
<td></td>
</tr>
<tr>
<td>NewSize</td>
<td>Shaded</td>
<td>3D</td>
<td>Direct</td>
<td>HP Workstation</td>
</tr>
<tr>
<td></td>
<td>Color</td>
<td>Menu</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single Perspective</td>
<td></td>
<td>Knob Box</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Button Box</td>
<td></td>
</tr>
</tbody>
</table>

Figure 20  Interface comparisons across several dimensions.
MEASURES

Suggested SEP(3D) usability measures of performance time and user satisfaction were collected for both experiments. Performance time was estimated by a single evaluator, the designer of TeeterTable. User satisfaction was estimated using the SEP(3D) questionnaire. Covariate measures of reaction time, pointing time, and spatial visualization skills were collected for Experiment One only.

ENVIRONMENT

Evaluation sessions were performed where subjects typically use their system with the exception of the TeeterTable systems (subjects had no previous experience). A video camera with a directional microphone was used throughout each session to record all trials. An evaluator was present to answer questions and guide subjects through the process.

EXPERIMENT ONE PROCEDURE

Subjects were told what was expected of them and why they were asked to evaluate the interface. Sessions began with a reaction time test and pointing time test on a Macintosh computer. A paper and pencil Mental Rotation Test was also completed.

With TeeterTable systems, subjects were given a brief description of how to perform each of the tasks, followed by ten minutes of practice. For all other systems, no instructions or practice time were necessary because subjects were already familiar with the system. Three trials were completed by each subject. The first two trials were intended to familiarize subjects with the specific performance tasks and the language of the instruction booklet. Subjects were encouraged to ask questions during practice trials.

The questionnaire was completed after finishing all trials. Subjects were encouraged to provide any additional information they felt useful. Total time required of each subject was approximately one and a half hours.

EXPERIMENT TWO PROCEDURE

Experiment Two used a within-subjects design. Four subjects performed tasks on two TeeterTable versions: NewRotate and NewSize. Two subjects learned NewRotate first and
two subjects learned NewSize first. After completing the evaluation for one version, subjects returned after approximately six days to learn the second version. The procedure was the same for both sessions.

Subjects were told what was expected of them and why they were asked to evaluate the interface. They were given a brief description of how to perform each of the tasks, followed by ten minutes of practice. Five trials were completed by each subject. The first two trials were intended to familiarize subjects with the specific performance tasks and the language of the instruction booklet. Subjects were encouraged to ask questions during practice trials.

The questionnaire was completed after finishing all trials. Subjects were encouraged to provide any additional information they felt useful. Total time required of each subject was approximately one and a half hours.
CHAPTER V
RESULTS

This chapter presents results from the previously described experiments. Data include performance times and questionnaire ratings. Results of several statistical analyses are presented.

RESULTS OF EXPERIMENT ONE

MISSING AND EXCLUDED DATA

A few data were not included in these analyses for various reasons. This section identifies the missing data and explains why some other data were not included.

Chalk

Chalk users can perform tasks in two distinct ways. Objects can be directly manipulated with a mouse or they can be manipulated by typing commands on a keyboard. Three subjects chose to perform tasks using the mouse and one subject preferred to use the keyboard. These techniques can produce different results and should not be considered equivalent. For example, typing a long command usually takes longer than selecting a menu item with a mouse. Combining times for each technique would not be an accurate measurement. Comparing these techniques would be worthwhile if each method was represented by enough subjects. However, the keyboard technique was only used by one subject. This would not provide enough data for a valid comparison. Therefore, the keyboard subject's data was not included in this study.
Triad

Triad subjects had difficulty learning to control the pointer. Three of the subjects only completed the first trial before their scheduled time expired. Performance times were so slow on trial one and unavailable for trials two and three that Triad data was excluded from these analyses. Although questionnaire data were available, they were also excluded because the subjects did not have an equal opportunity to evaluate Triad.

Chalk, AutoCAD, and Form*Z

Trial three was assumed to provide the most stable performance times. The first two trials were considered practice and allowed subjects to become familiar with the tasks and instructions. Trials one and two were not recorded for Chalk, AutoCAD, and Form*Z. However, all three trials were recorded for the TeeterTable versions and a significant effect was observed for trials ($F(2,18)=28.44$, $p<.001$). Average performance time for the first trial was 24.8 seconds, the second trial was 18.34 seconds, and the third trial was 15.9 seconds. Further analyses concerning the appropriate number of trials are presented in Experiment Two.

Covariate Measures

Covariate measures were collected but not used for these analyses because they had different levels of effect across each interface. This violates the key assumption of homogeneity of slopes and would make results from an analysis of covariance questionable. Some of the data satisfied the assumption, but a preliminary analysis did not increase the significance of the results. A few correlations between performance times and covariate measures were significant ($p < .05$) but they did not increase the significance of the results (Figure 21). This indicates that these covariate measures are not useful in SEP(3D).
<table>
<thead>
<tr>
<th>Task</th>
<th>Reaction Time</th>
<th>Pointing Time</th>
<th>Visualization Test (part 1)</th>
<th>Visualization Test (part 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move XY</td>
<td>0.18</td>
<td>0.41 *</td>
<td>-0.20</td>
<td>-0.03</td>
</tr>
<tr>
<td>Move XYZ</td>
<td>0.20</td>
<td>0.38 *</td>
<td>-0.17</td>
<td>0.08</td>
</tr>
<tr>
<td>Size XY</td>
<td>0.13</td>
<td>0.16</td>
<td>-0.08</td>
<td>0.04</td>
</tr>
<tr>
<td>Size XYZ</td>
<td>-0.18</td>
<td>-0.01</td>
<td>-0.37 *</td>
<td>-0.35 *</td>
</tr>
<tr>
<td>Rotate V</td>
<td>0.21</td>
<td>0.17</td>
<td>-0.15</td>
<td>-0.37 *</td>
</tr>
<tr>
<td>Rotate H</td>
<td>0.20</td>
<td>0.39 *</td>
<td>0.12</td>
<td>0.04</td>
</tr>
<tr>
<td>View</td>
<td>0.09</td>
<td>0.22</td>
<td>-0.16</td>
<td>-0.17</td>
</tr>
<tr>
<td>Create</td>
<td>0.37 *</td>
<td>0.46 *</td>
<td>-0.02</td>
<td>-0.11</td>
</tr>
</tbody>
</table>

Figure 21  Pearson's correlation between performance times and covariate measures (n = 27). An asterisk indicates a significant correlation (p < .05).

DATA ANALYSES

Statistical methods used for each type of data are described in this section.

Performance Times

The most common statistical test to determine significant differences between several samples is the analysis of variance (ANOVA). The basic principle is to determine whether the sample means vary further from the population mean than would be expected in view of the variations of single cases from their means (Guilford and Fruchter 1978). Performance times for this study did not allow an ANOVA to be performed because several key assumptions about distribution were not met. The existence of extreme outliers, a wide range of variances between cells, unequal cell sizes, and cells that were not normally distributed would make analysis of variance results questionable.

Kruskal-Wallis

The Kruskal-Wallis test for k-independent-samples was used to compare times. It has good power and makes few assumptions about data distribution. The test replaces actual observations by their ranks. This method is more robust than ANOVA when outliers are present in the data (Neave and Worthington 1988).
Dunn's Multiple Comparison Procedure

Whenever evidence exists to suggest that the $k$ populations are different ($p < 0.05$), Dunn's multiple comparisons procedure was used to discover which populations were better or worse than others. It is typical to use one overall level of significance for all comparisons. Dunn argues for levels greater than the usual maximum value of 5% to be taken so that any differences can easily be detected (Wilkinson 1989). The rationale for using a higher level of significance is that this technique is used only when a significant result has already been detected by a significance test. Higher significance levels can be tolerated as the number of populations being compared increases. This is because there are larger numbers of pairwise comparisons to share the risk of making an error. For this analysis, there were seven populations and a significance level of 30% was used. The 30% level was equally divided among the twenty-one comparisons being made.

Questionnaire Ratings

Ratings were averaged for each task category. For example, the six items associated with translation were averaged for an overall rating of translation. However, this could be misleading if the questions do not measure the same characteristic. A principal components analysis with a varimax rotation was performed on each section of the questionnaire to determine if the group of items measured the same characteristic. Single components were identified for each of the Move, Size, View, Create, and General sections. The Rotate and Visual sections each had two components identified. However, combining all items into a single component for each section did not alter the significant results. Results were subjected to an analysis of variance (these data satisfy the distribution assumptions of ANOVA). Tukey HSD multiple comparisons were performed when an overall significant result was observed.

Correlation

A correlation coefficient measures how closely two variables are related. The value lies between negative one and one. If two variables are not related at all, the correlation coefficient will be close to zero. If two variables change in the same direction, the
coefficient will be close to one. Likewise, if two variables change in opposite directions but at a similar rate, the coefficient will be close to negative one.

Although performance times and questionnaire responses are different types of data, it would be useful to confirm that they are both a valid measure of usability. If the two measures are highly correlated, one measure can be used to predict the other measure accurately. If the correlation is low, predictions would be less accurate.

Averages were calculated for each task performance time and rating. Pearson's correlation was used to measure the degree of linear relationship between these two variables. Using a group average for each interface reduced the effects of outliers and increased the correlations. The average correlation between tasks and ratings for each group was -0.711.

Subject Comments

Comments in response to the open-ended items of the questionnaire can provide useful information early in system development but are not useful in empirical comparisons because they cannot be categorized easily for structured comparisons. Comments are included in Appendix C.

PRESENTATION OF RESULTS

Box plots are used to show the distribution of the original data (Figure 22). The median of the data is marked by a line passing through the box. The median splits the ordered numbers in half and the hinges split the remaining halves in half again. The lower and upper hinges form the top and bottom of the box and indicate the interquartile range. \( H_{spread} \) is the absolute value of the difference between the hinges. Whiskers protrude from both ends of the box and indicate the lower and upper inner fences. Values outside the inner fences are plotted with asterisks and values outside the outer fences are plotted with circles (Wilkinson 1989).
Inner and outer fences are calculated with the following formulas:

\[
\begin{align*}
\text{lower inner fence} &= \text{lower hinge} - (1.5 \times \text{Hspread}) & (1) \\
\text{upper inner fence} &= \text{upper hinge} + (1.5 \times \text{Hspread}) & (2) \\
\text{lower outer fence} &= \text{lower hinge} - (3.0 \times \text{Hspread}) & (3) \\
\text{upper outer fence} &= \text{upper hinge} + (3.0 \times \text{Hspread}) & (4)
\end{align*}
\]

Results for multiple comparisons are often presented in a diagram. The populations are presented in order of their mean ranks, and lines are drawn to join any populations that the procedure has failed to separate (Neave and Worthington 1988). Figure 23 shows the pairwise comparisons of five hypothetical systems. Systems A, B, and C could not be significantly separated. The same is true for systems B, C, and D. In addition, systems D and E could not be separated.

Figure 22 Description of box plot.  
Figure 23 Paired comparison figure.
RESULTS OF TRANSLATION

Four translation tasks were performed. For each subject, two times are included for Move XY and two times are included for Move XYZ.

Performance Times

Performance times for Move XY are presented in Figure 24. An overall effect was observed for Move XY ($\chi^2(6) = 17.11$, $p < .01$). Paired comparisons revealed AutoCAD was significantly slower than Wireframe and MultiMode (Figure 25). In addition, Chalk was significantly slower than MultiMode.

Performance times for Move XYZ are presented in Figure 26. An overall effect was observed for Move XYZ ($\chi^2(6) = 31.26$, $p < .001$). Paired comparisons revealed AutoCAD was significantly slower than Wireframe, Original, Modified, and MultiMode (Figure 27). In addition, Chalk and Form*Z were both significantly slower than Modified and MultiMode.

Questionnaire Ratings

Items 1-6 referred to translation. Responses were averaged for an overall rating of translation (Figure 28). A significant effect was observed ($F(6,21) = 4.47$, $p < .01$). Paired comparisons revealed that Modified and MultiMode were preferred over AutoCAD and Chalk ($p < .05$).

Correlation

Pearson's correlation between Move ratings and Move XY performance times was significant ($r = -0.799$, $n = 7$, $p < .05$). Pearson's correlation between Move ratings and Move XYZ performance times was significant ($r = -0.869$, $n = 7$, $p < .05$).
Figure 24 Performance times for Move XY.

Figure 25 Paired comparisons of performance times for Move XY.
Figure 26 Performance times for Move XYZ.

Figure 27 Paired comparisons of performance times for Move XYZ.

Figure 28 Average questionnaire ratings for Move.
RESULTS OF SIZING

Four sizing tasks were performed. For each subject, two times were included for Size XY and two times were included for Size XYZ.

Performance Times

Performance times for Size XY are presented in Figure 29. An overall effect was observed for Size XY ($\chi^2(6)= 20.12, p < .01$). Paired comparisons revealed AutoCAD was significantly slower than Chalk, MultiMode, Original, Modified, and Wireframe (Figure 30).

Performance times for Size XYZ are presented in Figure 31. An overall effect was not observed for Size XYZ ($\chi^2(6)= 10.91, p = .09$). Therefore, paired comparisons were not performed.

Questionnaire Ratings

Items 17-22 referred to sizing. Responses were averaged for an overall rating of sizing (Figure 32). A significant effect was not observed ($F(6,21) = 1.99, p = .113$). Therefore, paired comparisons were not performed.

Correlation

Pearson's correlation between Size ratings and Size XY performance times was significant ($r = -0.95, n = 7, p < .01$). Pearson's correlation between Size ratings and Size XYZ performance times was significant ($r = -0.912, n = 7, p < .01$).
Figure 29  Performance times for Size XY.

Figure 30  Paired comparisons of performance times for Size XY.
Figure 31 Performance times for Size XYZ.

Figure 32 Average questionnaire ratings for Size.
RESULTS OF ROTATION

Four rotation tasks were performed. For each subject, two times were included for Rotate V and two times were included for Rotate H.

Performance Times

Performance times for Rotate V are presented in Figure 33. An overall effect was not observed for Rotate V ($\chi^2(6)= 10.32, p = .1$). Paired comparisons were not made.

Performance times for Rotate H are presented in Figure 34. An overall effect was observed for Rotate H ($\chi^2(6)= 24.89, p < .001$). Paired comparisons revealed AutoCAD was significantly slower than Original, MultiMode and Modified (Figure 35). In addition, Form*Z was significantly slower than Modified and MultiMode.

Questionnaire Ratings

Items 9-14 referred to rotation. Responses were averaged for an overall rating of rotation (Figure 36). A significant effect was not observed ($F(6,21) < 1$). Therefore, paired comparisons were not performed.

Correlation

Pearson's correlation between Rotate ratings and Rotate V performance times was not significant ($r = -0.237, n = 7, p > .1$). Pearson's correlation between Rotate ratings and Rotate H performance times was not significant ($r = 0.332, n = 7, p > .1$).
Figure 33  Performance times for Rotate V.
Figure 34 Performance times for Rotate H.

Figure 35 Paired comparisons of performance times for Rotate H.

Figure 36 Average questionnaire ratings for Rotate.
RESULTS OF VIEW MANIPULATION

Four view manipulation tasks were performed. For each subject, four times were included for View.

Performance Times

Performance times for View are presented in Figure 37. An overall effect was observed for View ($\chi^2(6) = 72.45, p < .001$). Paired comparisons revealed Modified was significantly faster than AutoCAD, Chalk, Form*Z, Original and Wireframe (Figure 38). In addition, AutoCAD and Chalk were significantly slower than Original, Wireframe and MultiMode.

Questionnaire Ratings

Items 25-30 referred to view manipulation. Responses were averaged for an overall rating of view manipulation (Figure 39). A significant effect was observed ($F(6,21) = 3.76, p < .05$). Paired comparisons revealed that Modified was preferred over AutoCAD and Chalk ($p < .05$).

Correlation

Pearson’s correlation between View ratings and performance times was significant ($r = -0.955, n = 7, p < .01$).
Figure 37 Performance times for view manipulation.

Figure 38 Paired comparisons of performance times for view manipulation.

Figure 39 Average questionnaire ratings for view manipulation.
RESULTS OF CREATE

Four object creation tasks were performed. All four times were included for Create.

Performance Times

Performance times for Create are presented in Figure 40. An overall effect was observed for Create ($\chi^2(6) = 22.66, p < .01$). Paired comparisons revealed AutoCAD was significantly slower than MultiMode, Modified, and Form*Z (Figure 41). In addition, Chalk was significantly slower than Wireframe, MultiMode, Modified, and Form*Z.

Questionnaire Ratings

Items 33-38 referred to object creation. Responses were averaged for an overall rating of object creation (Figure 42). A significant effect was observed ($F(6, 21) = 4.76, p < .01$). Paired comparisons revealed that Modified and MultiMode were preferred over AutoCAD and Chalk ($p < .05$).

Correlation

Pearson’s correlation between Create ratings and performance times was significant ($r = -0.93, n = 7, p < .01$).
Figure 40  Performance times for Create.

Figure 41  Paired comparisons of performance times for Create.

Figure 42  Average questionnaire ratings for Create.
Figure 43 Average questionnaire ratings for Visual Feedback.

VISUAL FEEDBACK

Items 6, 14, 22, 30, 38, 45, and 46 referred to visual feedback. Responses were averaged for an overall rating of visual feedback (Figure 43). A significant effect was observed ($F(6,21) = 7.12, p < .001$). Paired comparisons revealed that Modified, MultiMode, Original, and Wireframe were preferred over AutoCAD and Chalk ($p < .05$).

Correlation

Pearson's correlation between Visual ratings and average performance times for all tasks was significant ($r = -0.898, n = 7, p < .01$).
RESULTS OF OVERALL COMPARISONS

Although the primary goal of SEP(3D) is to evaluate individual techniques within a 3D system, comparisons of the overall systems can also be useful.

Performance Times

Average performance times for all tasks are presented in Figure 44. An overall effect was observed for average task completion time ($F(6, 21) = 8.24, p < .001$). Paired comparisons revealed AutoCAD was significantly slower than Form*Z, Wireframe, Original, MultiMode, and Modified (Figure 45). In addition, Chalk was significantly slower than MultiMode and Modified.

Questionnaire Responses

Items 6, 14, 22, 30, 38, 45, and 46 referred to general characteristics of the interfaces. Ratings were averaged for an overall rating of the interface (Figure 46). A significant effect was observed ($F(6, 21) = 9.53, p < .01$). Paired comparisons revealed that Modified, MultiMode, Original, and Wireframe were preferred over AutoCAD and Chalk ($p < .01$).

Correlation

Pearson’s correlation between General ratings and average performance times for all tasks was significant ($r = -0.892, n = 7, p < .01$).
Figure 44  Average performance times for all tasks.

Figure 45  Paired comparisons of average performance for all tasks.

Figure 46  Average questionnaire ratings for general characteristics.
Correlation of Average Ranks

Although few significant differences were observed for performance times and questionnaire ratings, the rank order for each task indicates meaningful trends. If an interface consistently ranks high for each task, the interface is relatively strong in all areas. Ranking high in both performance times and responses suggests the two measures are estimating the same factor. Performance results were ranked from 1 (lowest time) to 7 (highest time) for each of the tasks and an average rank was computed for each interface. Questionnaire ratings were also ranked from 1 (most preferred) to 7 (least preferred) and an average rank was computed for each interface. Both averages are presented for each interface (Figure 47). Spearman's correlation between average performance time rankings and average questionnaire rankings was significant ($r = 0.929$, $n = 7$, $p < .01$).
DATA ANALYSES

Descriptions of the statistical methods used for each type of data are presented in this section.

Performance Times

Performance data violated some of the distribution assumptions of ANOVA. Therefore, a non-parametric test was used to analyze the data. Wilcoxon's signed-rank test for matched pairs was used because of the experiment's within-subjects design. This test compares the differences between a set of matched pairs to find significant differences. For this study, each subject has a matched pair of performance times for each task (one time for each interface). A pair exists for each of the five trials. At least two trials (eight data points) were necessary to provide enough data for the Wilcoxon test. Results from a single trial (four data points) would not provide enough data to allow the Wilcoxon test to indicate a significant difference (p < .05). The fourth and fifth trials were used instead of earlier trials to reduce any learning effect.

Significant differences were not observed for Move, View, or Create. This was expected because both interfaces were the same for these tasks. Therefore, these performance results are not presented.

Questionnaire Ratings

Ratings were averaged for each task category. For example, the six items associated with translation were averaged for an overall rating of translation. Results were subjected to an analysis of variance. No significant differences were observed for any of the tasks. Therefore, questionnaire results are not presented. Subject responses to the open-ended items are provided in Appendix C.
RESULTS OF SIZING

Wilcoxon's signed-rank test did not detect a significant difference for Size XY (n = 8, p = .53). However, a significant difference was observed for Size XYZ (n = 8, p < .05). Performance times for Size XYZ are presented in Figure 48.

Figure 48 Performance Times for Size XYZ.
RESULTS OF ROTATION

Wilcoxon's signed-rank test did not detect a significant difference for Rotate V (n = 8, p = .39). However, a significant difference was observed for Rotate H (n = 8, p < .05). Performance times for Rotate H are presented in Figure 49.

PERFORMANCE TRENDS

One of the benefits of obtaining data from five trials instead of three is that performance trends can be identified. In addition, the effects of a within-subject design can be analyzed. Significant effects due to order would indicate that within-subject data may be confounded. Figure 50 shows the combined averages of NewRotate and NewSize for each trial. Trials one through five represent each subject's first session. Trials six through ten represent each subject's second session that was completed approximately six days later on a different system than was used during the first session. Performance times began to stabilize during the fourth trial of each session. Correlations between each pair of trials are provided in Figure 51. All reported correlations are significant (P < .01).
Figure 50  Average performance times for each trial. The first session includes trials one through five. The second session includes trials six through ten.

<table>
<thead>
<tr>
<th></th>
<th>Trial 1</th>
<th>Trial 2</th>
<th>Trial 3</th>
<th>Trial 4</th>
<th>Trial 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial 1</td>
<td>1.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Trial 2</td>
<td>0.68</td>
<td>1.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Trial 3</td>
<td>0.49</td>
<td>0.59</td>
<td>1.00</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Trial 4</td>
<td>0.54</td>
<td>0.63</td>
<td>0.80</td>
<td>1.00</td>
<td>—</td>
</tr>
<tr>
<td>Trial 5</td>
<td>0.51</td>
<td>0.59</td>
<td>0.70</td>
<td>0.81</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Figure 51  Pearson's correlation between trials (n = 64).

**Overall Effects**

An ANOVA was performed on the entire set of data to identify significant interactions related to the within-subjects design. To reduce the effect of several outliers, a log transformation was applied to all data for this analysis only. (The previous analyses using Wilcoxon's test were robust to outliers. Therefore, a log transformation was not necessary.) Main effects were observed for task \(F(7, 14) = 11.85, p < .001\) and trial \(F(4, 8) = 11.60, p < .01\). These effects were expected because there should be a difference in performance times from one trial to the next. In addition, performance times
should vary between tasks because they represent a range of unique tasks. There would have been no task interaction if all SEP(3D) tasks were similar.

There was a significant interaction between interface and task \((F(7, 14) = 3.49, p < .05)\). This interaction confirms that SEP(3D) can detect a range of differences between interfaces. While an interface can excel with one task, it can be worse for another task. This shows a high level of sensitivity for SEP(3D).

There was a significant three-way interaction between interface, trial and order \((F(4,8) = 9.65, p < .01)\). This may indicate that the within-subjects design provides confounded results due to order effects. Perhaps subjects benefit from using one system before another. If so, the results should be discarded. The following section considers this problem.

**Performance Profiles**

Figures 52-60 represent performance profiles for each task. Average performance times are presented for each trial and system. Trials one through five are from the first session and trials six through ten are from the second session. Two subjects used NewRotate in the first session followed by NewSize in the second session. Two other subjects used each system in the opposite order. Therefore, any conclusions drawn from these profiles must be qualified by the fact that only two subjects were available for each session/system combination.

**Differences Between Techniques**

Significant differences for a task are identified when one system clearly performs better than the other during both sessions. This occurred with Size XYZ (Figure 56) and to a lesser extent with Move XYZ (Figure 54). This pattern also occurred with Rotate H (Figure 58) when only considering the fourth and fifth trials. These observations match the findings of the Wilcoxon analyses.
Differences Between Subjects

Profiles that show a system did better during the first session and worse during the second session, or vice versa, may indicate an order effect. Skills acquired from using the first system may affect performance with the second system.

A second interpretation is that subject groups do not have equal skill levels. Large skill differences between groups would make one system appear faster in the first session and make the other system appear faster in the second session. Any observed differences would be due to different skill levels rather than the order the systems were used. This interpretation is the most appropriate for this study.

Techniques for performing view manipulation and object creation were identical for both systems. However, Figures 59 and 60 show that subjects who used NewRotate followed by NewSize were faster during both sessions. Differences should not appear between identical techniques. Therefore, the observed differences must be due to skill levels. This pattern is also shown for the overall average (Figure 52), Size XY (Figure 55), and Rotate V (Figure 57).

Many of the profiles indicate that subjects were slower to learn NewRotate compared to NewSize in the first session. However, NewRotate performance improved quickly and surpassed NewSize by the third trial for most tasks. This apparent interaction did not occur because of different learning curves related the systems but because one of the two NewRotate subjects was a particularly slow starter. In this situation, the profiles are misleading because only two subjects are included for each system/session combination.

No Observed Differences

A third category of profiles show that task performance is virtually the same with either system. This was clearly true for Move XY (Figure 53). The different skill levels that appeared for most other tasks did not affect performance of Move XY.
Figure 52 Overall Average performance times for the first and second sessions.

Figure 53 Performance times for Move XY during the first and second sessions.

Figure 54 Performance times for Move XYZ during the first and second sessions.
Figure 55  Performance times for Size XY during the first and second sessions.

Figure 56  Performance times for Size XYZ during the first and second sessions.

Figure 57  Performance times for Rotate V during the first and second sessions.
Figure 58 Performance times for Rotate H during the first and second sessions.

Figure 59 Performance times for View during the first and second sessions.

Figure 60 Performance times for Create during the first and second sessions.
CHAPTER VI
OBSERVATIONS

SEP(3D) is used to detect significant differences between interfaces and it is the evaluator's responsibility to determine why the interfaces are different. This is a simple task in highly controlled comparisons because all but one or two features are identical between interfaces. The feature that is not identical will likely be the cause of the observed difference. However, comparing commercial software products that vary across several features is more difficult. An evaluator must identify features that are likely responsible for the differences and then additional comparisons must be performed to isolate the suspected differences. In addition, an evaluator can identify common performance errors and aspects of the interface that may affect performance time.

This chapter includes my observations to explain the significant differences detected in the previous experiments.

OBSERVED DIFFERENCES

Evaluator observations can be used to explain significant differences as well as recognize important differences. Roberts and Moran (1983) make a distinction between reliability and importance. Any observed differences can be shown to be reliable by collecting enough data. The objective is to determine if the differences are important, which is a substantive, not a statistical issue. The benefit of a simple evaluation plan is that it can be used to reveal potentially important differences quickly. Once an important difference has been identified, then a determination can be made of how reliable the difference needs to be. Additional experiments can be performed to confirm the observations.
Brooks (1988) also recognizes a distinction between narrow truths proved convincingly with reliable results, "and broad 'truths,' generally applicable, but supported only by possibly unrepresented observations." He suggests there is a need for both types of results because more information is known than is reported, due to a reluctance to share unproven results.

Phases of Performance Time

Performance time occurred in three phases: 1) task selection, 2) item selection, and 3) task performance. *Task selection* involves selecting the desired mode or initiating the desired action. This is most often performed with a menu selection or typing a command. *Item selection* involves selecting the item to manipulate. An object could be a vertex, face, collection of faces, object, or collection of objects. This is most often performed by pointing to the item or specifying the item's name. *Task performance* involves manipulating the selected item and is performed by dragging the item or specifying a command.

Interface designs can affect one or more of these phases. Although it may be difficult to time each phase with a stopwatch reliably, intuition and observation indicate which phases take relatively longer for each interface. Specific changes can be made to an interface to affect any combination of these phases.

TRANSLATION

Two Move XY and two Move XYZ tasks were performed. Move XY tasks involved moving an object across the floor of the scene. Move XYZ tasks involved placing an object on top of another object or removing an object and placing it back on the floor.

Significant Results During Translation

AutoCAD and Chalk were slower than MultiMode when performing Move XY. AutoCAD, Chalk, and From*Z were slower than Modified and MultiMode when performing Move XYZ. Questionnaire ratings indicated that Modified and MultiMode were preferred over AutoCAD and Chalk.
Observations About Translation

AutoCAD was slower than the other interfaces because there were many more commands and options to specify with AutoCAD. This increased the task selection phase of performance.

Chalk users were not able to manipulate objects directly. The pointer was constrained in a separate window, called the chalkboard. Pointer movement on the chalkboard controlled object movement in the scene. However, there was no direct correspondence between mouse and object movement. This sometimes caused the user to be disoriented and increased task performance time.

Move XYZ tasks required objects to be placed on top of other objects. A 3D wireframe representation made it difficult to determine if objects were positioned correctly. Form*Z and AutoCAD users changed views to a 2D elevation to confirm that an object was correctly positioned on top of another object. The additional action of manipulating the view increased task performance time.

A composite mode was provided in the Original version of TeeterTable to perform translation, creation, and sizing. The relative position of the 3D pointer within an object determined which action was performed. Some users made errors during the item selection phase by picking a face instead of the entire object. MultiMode and Modified provided separate modes for each task to eliminate this problem. Transition from one task to another was not as smooth but performance was improved because fewer errors were made during item selection.

Recommendations for Translation

• Provide adequate visual feedback in a single view to indicate object relationships clearly.

• If a composite mode is used to perform several different tasks, provide cues to indicate which task is currently active. In addition, protect against inadvertently selecting a task.
SIZING

Two Size XY and two Size XYZ tasks were performed. Size XY tasks involved proportionally adjusting the width and depth of an object. Size XYZ tasks involved adjusting the width, depth, and height of an object.

Significant Results During Sizing

AutoCAD was slower than Chalk and all TeeterTable versions when performing Size XY. Significant differences were not observed for Size XYZ. Wide ranges of times occurred for all interfaces indicating that sizing is a relatively difficult task. Questionnaire ratings did not reveal any significant differences.

In Experiment Two, NewSize was faster than NewRotate for performing Size XYZ. Questionnaire ratings did not indicate significant differences.

Observations About Sizing

AutoCAD was slowest because subjects could not size along more than one dimension at a time. Performing three separate actions to size along XYZ increased performance time.

An unexpected problem was observed when subjects adjusted the height of an object. TeeterTable and Form*Z subjects performed this task by selecting the top face of an object and dragging it in a vertical direction. Occasionally, subjects would inadvertently move the face in a horizontal direction rather than vertical. Although moving the face away from the viewpoint gave the illusion of increasing the height, the object was being sheared (Figure 61). Some objects became extremely distorted before subjects noticed the problem. This problem can be prevented by constraining the top face to move in a vertical direction.

Form*Z allowed this option but subjects did not always remember to use it.

Recommendations for Sizing

- Provide user-defined constraints for sizing along one or more axes. The selection technique should be well integrated into the task so that users remember the option exists but are not forced to use it.
ROTATION

Two Rotate V and two Rotate H tasks were performed. Rotate V tasks involved rotating an object forty-five degrees about the vertical axis passing through the object's centroid. Rotate H tasks involved rotating an object forty-five degrees about a horizontal axis passing through the object's centroid.

Significant Results During Rotation

Significant differences were not observed for Rotate V. AutoCAD and Form*Z were slower than Modified and MultiMode when performing Rotate H. Questionnaire ratings did not reveal any significant differences. In Experiment Two, NewRotate was faster than NewSize for performing Rotate H. Questionnaire ratings did not reveal significant differences.

Observations About Rotation

Some Chalk subjects had difficulty selecting the correct rotation axis from the menu. The X, Y and Z labels for world axes were not intuitive for the user after the scene was reoriented. The labels appeared to have different meanings depending on how the scene was oriented. This inconsistency caused errors and increased task selection time.
AutoCAD and Form*Z users typically had to change to a 2D projected view to perform a horizontal rotation. They were not able to perform the task accurately in a 3D wireframe view. The additional view manipulation increased performance time.

Some TeeterTable users had difficulty recognizing the relationship between the point they selected on an object and the resulting axis of rotation. This confusion increased task selection time. In addition, it was difficult to select a precise axis. Subjects were often annoyed that the object would not rotate 'exactly' as they desired. NewRotate in Experiment Two addressed this problem by providing constraints. The constraints improved task selection time.

Recommendations for Rotation

• Allow axis selection without reference to a particular X, Y, Z label. For example, axes should be selected in the 3D scene and not from a menu.
• Provide constraints so that a particular axis is selected when desired. Constraints allow new users to control rotations precisely.
• Provide visual cues to indicate which axis is currently selected.

VIEW MANIPULATION

Four View tasks were performed. They all involved spinning the scene around a vertical axis to alter the point of view.

Significant Results During View Manipulation

AutoCAD and Chalk were slower than all of the TeeterTable versions. In addition, Modified was faster than Form*Z, Original, and Wireframe. Questionnaire ratings revealed that Modified was preferred over AutoCAD and Chalk.

Observations About View Manipulation

TeeterTable, AutoCAD and Form*Z provided direct manipulation techniques for view manipulation. Modified was most effective because the scene was constrained to rotate around a vertical axis. All other systems required the user to also adjust the tilt of the scene about the horizontal axis. This was especially time consuming for AutoCAD users because
the two actions were performed separately. The user first adjusted the tilt of the scene and then adjusted the rotation. TeeterTable and Form*Z users were faster with this task because they could combine these actions into a single motion.

Form*Z visual cues were confusing for view manipulation. As the scene was rotated, the original objects and their new positions were both displayed in wireframe. Therefore, there were two representations displayed while the scene was rotated. This was often confusing to the user and increased performance time.

Low questionnaire ratings indicate that all Chalk subjects found view manipulation difficult. Subjects typed XYZ coordinates to indicate an eye position. Since the coordinates were in world space, subjects had to remember the current eye position before a new eye position could be determined. For example, if a user is currently looking at the scene from the South, the eye position might be (0.0, 10.0, 0.0) in world coordinates. Changing the view to look from the East would require a new eye position (10.0, 0.0, 0.0). Subjects often made several attempts of specifying coordinates until the view was correct.

Recommendations for View Manipulation
- Provide direct manipulation techniques with as many visual cues as possible while still allowing real time motion.
- Provide user-defined constraints for performing the manipulation.

OBJECT CREATION

Four Create tasks were performed that involved creating cubes of similar sizes.

Significant Results During Object Creation

AutoCAD and Chalk were slower than MultiMode, Modified, and Form*Z. Questionnaire ratings indicated that Modified and MultiMode were preferred over AutoCAD and Chalk.
Observations About Object Creation

AutoCAD users were required to type several options to create a simple cube. These options included height, width, depth, orientation, and position. The lack of direct manipulation increased performance time.

Recommendation for Object Creation

- Allow object construction in a single direct manipulation action.

VISUAL FEEDBACK

The commercial interfaces provided wireframe representations. Form*Z and Chalk generally provided a single window and AutoCAD provided multiple windows for orthographic and perspective projections. The TeeterTable versions provided a single perspective view.

Significant Results

Questionnaire ratings indicated AutoCAD and Chalk were preferred less than all of the TeeterTable versions.

Observations About Visual Feedback

AutoCAD subjects used two windows during their design session. One window provided a 3D view of the scene and the other window was a 2D view from above the scene, called a plan view. Objects were selected and manipulated in the plan view. However, the orientation of the plan view did not change when the 3D view was manipulated. Subjects often made errors during item selection because they had to rotate the 3D scene mentally to match the orientation of the plan view.

The lack of effective visual feedback can be confusing or misleading. For example, a Chalk subject inadvertently moved an object in the vertical direction instead of back into the scene as required. However, she did not recognize the error had occurred because perspective was not used. For one of the Move XYZ tasks, it was difficult to determine if an object was correctly positioned on top of another object. TeeterTable's solid visual cues
were helpful in understanding the relationship between objects. However, the wireframe representations in Wireframe, Form*Z, AutoCAD and Chalk were ambiguous.

Several TeeterTable subjects had difficulty selecting an object that was above the XY grid. TeeterTable’s 3D pointer was especially ambiguous with the wireframe representation. However, Form*Z, AutoCAD, and Chalk subjects had few problems with this task because objects were selected with a 2D pointer.

Recommendation for Visual Feedback

- Allow all tasks to be easily performed in a single perspective view with solid objects.
CHAPTER VII
EVALUATION OF SEP(3D)
AND CONCLUSIONS

This chapter considers four criteria to evaluate the effectiveness of SEP(3D). 1) Are the results reliable? Are they consistent, and if an experiment were repeated, would similar results be found? 2) Are the results valid? Are they a true measure of usability for 3D manipulation? 3) Are the results useful? Can different systems be compared with respect to individual techniques? Can new design guidelines be discovered or can existing guidelines be confirmed? 4) Is SEP(3D) easy to use? Can people with different backgrounds effectively use SEP(3D) for useful purposes? This chapter also includes recommendations to evaluators who choose to use SEP(3D), limitations of SEP(3D), and concluding remarks about this study.

RELIABILITY

An evaluation must produce reliable results. This includes selecting appropriate subjects and recording useful measures. Methods to collect these measures should be well-structured so that any researcher can obtain consistent results. The reliability coefficient is a ratio of the variance of true scores to the variance of obtained scores. True scores are the results that would be obtained if the measurement were perfectly accurate and obtained scores are the actual results. Since the true scores are never known, the reliability of an evaluation must be estimated using statistical techniques on the obtained scores.

Subjects

Both the number and quality of subjects are important factors when designing an experiment. SEP(3D) recommends four subjects for each interface. Experts are often
defined by the number of months experience a user has with an interface. However, the best indication of expertise is consistent performance. Subjects should be able to perform the relatively simple SEP(3D) tasks at a steady pace with few errors. Subjects who do not meet this standard after a few practice trials should not be included in an evaluation of 'expert' performance.

Instruction Booklet

Clear and consistent instructions are a vital part of an evaluation plan. Subjects must be able to follow instructions without help from the observer. All tasks must be performed consistently to insure valid comparisons across subjects, interfaces, and experiments. The SEP(3D) instruction booklet was successful in this respect. Subjects found the instructions easy to follow and no errors were observed due to misunderstanding the instructions.

Performance Times

A common procedure for estimating the reliability of an evaluation is the test-retest method. It is obtained by determining the correlation between scores on one administration of the evaluation and scores on a second administration. A high correlation indicates that any one administration of an evaluation will likely produce consistent results. A low correlation reduces the confidence a researcher can have in a single administration.

The most appropriate measures for the test-retest method from this study are performance times in Experiment Two. The Pearson correlation between trials three and four was significant ($r = 0.80$, $n = 64$, $p < .01$). The Pearson correlation between trials four and five was also significant ($r = 0.81$, $n = 64$, $p < .01$). Figure 51 in Chapter V shows that the correlation increased between each successive pair of trials.

Performance Time Estimation

Ideally, two or three evaluators would make independent observations and results would be pooled. However, it may be impractical to have several evaluators for a study when dozens of subjects participate. A post hoc study was conducted to determine the reliability of a single evaluator. The times estimated by the author were compared with
times by two independent evaluators, Rater B and Rater C. After reviewing one tape for practice, Raters B and C evaluated the performance times for three subjects. The random sample of subjects represented three interfaces.

The intraclass correlation for sixty observations was 0.876. This indicates that the reliability of a single evaluator can be considered 87.6% (Guilford and Fruchter 1978). Raters B and C provided statistically identical results but their times were significantly higher than my times (F(3,57) = 6.96, p < .001). However, there was no interaction between rater and interface. This indicates that differences observed by each rater were consistent between interfaces.

These results suggests that SEP(3D) performance times should not be used for comparison by different evaluators because the timing method may not create consistent results.

**Test-Retest Reliability**

The test-retest reliability of performance time estimation for a single evaluator was determined by comparing the author’s times from the initial evaluation with a second evaluation of the same subjects. Several weeks separated the first evaluation from the second. The Pearson correlation between both evaluations was significant (n = 60, r = 0.998, p < .001). This indicates that the timing method provides consistent results for a single evaluator over time.

**Reliability versus Validity**

A tradeoff occurs between obtaining reliable results and valid results. Removing ‘matching delay’ from the end of each task makes the time a more valid measure of expert performance because designers would not normally be concerned about ‘matching’ during an informal design. However, this timing method introduces the possibility of evaluator bias. Different evaluators might record different times because individual judgment is used to determine when the task is completed.

If matching delay is not removed, independent evaluators can rate performance time with a high degree of reliability. Another post hoc study was performed to determine the
reliability of an alternative method for estimating performance time. Performance time began after the subject read an instruction aloud and ended when the subject turned the page to start the new task. An additional independent evaluator, Rater D, and the author evaluated performance times for three subjects using this method. The Pearson correlation between our times was significant \((n = 60, r = 0.998, p < .001)\). My average time (23.8 s) was similar to Rater D's time (24.1 s). Although this timing method provided more reliable results, the times were less valid because they included matching delay.

Covariate Measures

Covariate measures of reaction time, pointing time, and spatial visualization skills did not provide useful results. Most of the data violated the key assumption of homogeneity of slopes making an analysis of covariance questionable. The few data that could be used in an analysis did not increase the significance of the results. Correlations between all covariate measures and performance times are presented in Chapter V (Figure 21). The search for appropriate and reliable covariate measures for 3D interaction should continue, but these preliminary results are not encouraging.

VALIDITY

An evaluation must also produce valid results. It must be clear that the plan is evaluating what is intended to be evaluated. SEP(3D) was designed to evaluate the usability of interfaces for performing quick and informal 3D design. There are many ways to determine the validity of an evaluation plan. The three principle types of validity are criterion-related validity, content validity, and construct validity. In addition, external and internal validity relate to the quality of an experimental design.

Criterion-related Validity

Criterion-related validity is an indication of how well the selected measures of a plan accurately estimate the desired traits. For this study, usability was considered the ease with which an interface can be used to perform 3D manipulation tasks. The literature confirms that performance times and user satisfaction are valid measures of usability.
Content Validity

*Content validity* is concerned with the subjective determination of whether the evaluation plan measures what is intended. A review of the literature clearly indicates that translation, rotation, sizing, and view manipulation are a common set of tasks for performing 3D design. The order and collection of SEP(3D) tasks accurately represent a typical design session.

Sensitivity

SEP(3D)'s sensitivity to detect differences can be divided into two categories. *Low sensitivity* is needed to detect differences between interfaces that may differ on several factors. For example, Experiment One compared TeeterTable versions with three commercial systems. These interfaces had different presentation styles, pointer techniques, and input devices. Unfortunately, there is not enough control to identify which factors contribute most to observed differences. This is not a weakness of SEP(3D) but an indication of the difficulty in comparing interfaces that differ on several factors that affect usability. For this study, low sensitivity was demonstrated when SEP(3D) detected differences between a TeeterTable version and one of the commercial systems.

*High sensitivity* is needed to detect small differences between interfaces that share all but one factor. This is most relevant to the iterative design of an interface. Highly controlled changes to an interface can be evaluated by repeated uses of SEP(3D). Any detected differences can be reliably attributed to the minor change. The TeeterTable versions are a good example of controlled changes during an iterative design. In addition, the three commercial systems evaluated in Experiment One happen to share many factors. The biggest differences between the three are interaction style. For this study, high sensitivity was demonstrated when SEP(3D) detected differences between any TeeterTable versions or between any of the commercial systems. The commercial systems shared many of the same features despite appearing different.
Task Validation

Ideally, each SEP(3D) task should be validated by demonstrating it can detect significant differences between at least two interfaces for both sensitivity levels. Differences can be detected by either performance times or user satisfaction. Figure 62 summarizes the task validation. All tasks were validated by at least one measure except for Rotate V. Rather than invalidating the inclusion of Rotate V, these results indicate that the interfaces used in this study were similar for performing Rotate V. This task could be validated by evaluating an interface that performed Rotate V differently than the interfaces in this study.

Performance times were the most useful measure. Questionnaire ratings did not indicate any differences in the high sensitivity category. However, questionnaires should still be included in an evaluation because they provide potentially useful information at a low cost.

<table>
<thead>
<tr>
<th>Task</th>
<th>High Sensitivity</th>
<th>Low Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Times</td>
<td>Ratings</td>
</tr>
<tr>
<td>Move XY</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Move XYZ</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Size XY</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Size XYZ</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Rotate V</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Rotate H</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>View</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Create</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Figure 62 Validation of each task. A check mark indicates a significant difference was detected.
Construct Validity

An evaluation plan has a high degree of construct validity if results correlate highly with other tests designed to measure the same traits. In addition, results must have a low correlation with other tests designed to measure a different trait. Convergent validity is a measure of how well similar tests measure the same trait and discriminant validity is a measure of how divergently other tests measure different traits.

Convergent Validity

The only measure available to evaluate the convergent validity of SEP(3D) performance times is SEP(3D) questionnaire ratings. If these results are highly correlated, it can be assumed that they are both measuring the same trait, usability. Excluding rotation, the correlations between performance times and ratings for all tasks ranged from -0.955 to -0.799 in Experiment One (an average of -0.902). These results indicate that SEP(3D) performance times and questionnaire ratings are measuring the same trait.

Discriminant Validity

Experiment One demonstrated that SEP(3D) results can identify differences between interfaces. However, the rank order of evaluated interfaces was similar for each task. For example, Modified was always faster than Form*Z which was always faster than AutoCAD. This may indicate that all SEP(3D) tasks are not necessary because they appear to measure the same trait. For example, it might be inferred that performing a single task would provide as much information as performing all twenty tasks. The rank order of the single task would be assumed to be the same for all other tasks.

SEP(3D) was designed to detect individual differences for each task. To justify including all tasks, conflicting results must be obtained between two systems. Experiment Two demonstrated the discriminant validity of SEP(3D). NewRotate was superior to NewSize for rotation and NewSize was superior to NewRotate for sizing. Unlike Experiment One, a single task was not sufficient for comparing interfaces.

NewRotate and NewSize were intentionally designed to demonstrate the discriminant capabilities of rotation and sizing. Ideally, several interfaces would be designed to further
demonstrate the discriminant capabilities of all tasks. Since each SEP(3D) task is unique, additional interfaces could be specifically designed to demonstrate discrimination between the remaining tasks.

**External Validity**

External validity of a research investigation is an indication of how well results can be generalized to other subjects, treatments, and experiments. It is a measure of how well the entire experiment is designed. By definition, a standardized evaluation plan should have a high degree of external validity. This study demonstrated the external validity of SEP(3D) by comparing interfaces on several different platforms.

**Internal Validity**

Internal validity is an indication of how an experiment is designed to minimize the effect of confounding factors. This indicates that the dependent variable is a direct effect of the independent variables. SEP(3D) was designed to insure performance time and user satisfaction are a direct result of the user interface.

**USEFULNESS OF RESULTS**

Chapter VI discusses the usefulness of the results. Statistically reliable differences, as well as common performance errors, were observed. Observations were presented to explain the differences. Recommendations to improve future techniques were made based on the results and observations.

**Iterative Design**

SEP(3D) can be used early in system development to identify weaknesses and strengths. The weaknesses can be removed or improved and a second evaluation can be performed to determine if the modifications were improvements. Several changes were made to TeeterTable after an initial evaluation of Original and MultiMode. Modified was designed to reduce errors, enhance visual feedback, and improve performance times.
Although only one significant difference between Modified and Original was identified for an individual task, Modified performed faster and received higher ratings for almost all tasks. A Wilcoxon Signed Ranks Test comparing average performance times for all tasks indicates that Modified was significantly faster than Original (n = 8, p < .05). The same test comparing average ratings for seven categories indicates that Modified was significantly more preferred than Original (n = 7, p < .05).

The significant difference for an individual task was for view manipulation. Modified’s view manipulation was more constrained than Original. The benefit of constrained movement was also demonstrated in Experiment Two. NewRotate’s constrained rotation was faster than NewSize for rotating objects about a horizontal axis. NewSize’s constrained sizing operation was faster than NewRotate for sizing objects along all three axes.

Evaluating Theories

Many personal computers are limited to wireframe representation but more powerful workstations are able to provide solid representation in real time. This is often considered better 3D visual feedback. The Wireframe version of TeeterTable was evaluated to determine if this is true.

Questionnaire items relating to visual feedback were averaged for each interface. There were no significant differences between the TeeterTable versions. Wireframe may have been rated higher and performed better than expected because the scene was simple. The relationships of four cubes could still be seen clearly in wireframe. A more complex scene with spheres and other objects would likely indicate differences between wireframe and solid representations.

Evaluator Observations

Informal but useful information was obtained by watching subjects perform tasks. This information can be used early in the development of an interface when changes can still be easily made.
Subject Comments

Comments in response to the open-ended questionnaire items can also be useful when performing an iterative design. However, subject comments are not recommended for empirical comparisons because they are difficult to categorize for a meaningful analysis.

EASE OF USE

One of the objectives of this study was to make SEP(3D) easy to use for anyone desiring information related to 3D interaction. This objective was met because SEP(3D) instruction booklets and questionnaires can be used by anyone. In addition, a video camera and stopwatch are the only equipment needed to evaluate performance data.

RECOMMENDATIONS TO EVALUATORS

SEP(3D) not only provides a collection of standardized stimuli but also includes validations and recommendations for experimental designs. Evaluators unable to use SEP(3D) for their situation are encouraged to design their own evaluation plan based on the recommendations in this section.

Tasks, Instructions, and Questionnaires

- Develop new instruction booklets that follow the SEP(3D) format to include additional tasks or unusual stimuli.
- Require subjects to perform a series of tasks that represent the breadth of your domain.
- Resist the temptation to perform a quick evaluation by comparing techniques for a single task. It is important to evaluate techniques in a realistic situation which means continually switching between several tasks during a design.
- Use the existing SEP(3D) questionnaire or follow the format when including additional questions for tasks you add.
Experimental Design

Experiment One demonstrated that a between-subject experimental design with only four subjects for each interface was sufficient for identifying significant differences between interfaces.

Experiment Two demonstrated that a within-subjects experimental design does not necessarily cause harmful order effects. However, only two subjects were used for each system/order combination. A difference in subject skills made it appear that an order effect had occurred. Combining times from both sessions reduced the skill level effect and provided a valid result for each system.

Number of Trials

Experiment One included results from only the third trial which limited the power of the analyses. The additional trials performed in Experiment Two increased the power of the statistical analyses at a low cost. Experiment Two also indicates that more than five trials may not be necessary. By the fourth trial, performance time approached asymptotic levels. This pattern would probably be similar for experts of any system. However, the number of trials necessary for novice users to approach asymptotic levels will vary from one system to another depending on the learnability of the system.

Performance Time Estimation

- Use the suggested timing method to remove 'matching delay' and pool times from as many evaluators as possible to reduce the effect of individual bias.

- Resist the temptation to be the only evaluator of a system that you have designed because you may be unintentionally biased when estimating performance time. However, results from this study indicate that it can be done when there are no alternatives. The author was able to evaluate TeeterTable subjects without any detected bias despite being the designer of TeeterTable.
SEP(3D) LIMITATIONS

Two limitations of SEP(3D) were observed while evaluating interfaces during this study.

Simple Scenes

A tradeoff was made between simple and complex scenes for stimuli. Simple scenes were chosen to allow interfaces on computer systems with different processing capabilities to be compared. Simple scenes allow all computer systems to display the image in real time. However, simple scenes did not allow a significant difference to be observed between wireframe and solid representations. Similar visual cues that would be expected to enhance complex scenes cannot be evaluated with the current SEP(3D) stimuli. An additional instruction booklet that uses complex scenes could be developed for evaluating computer systems that have enough processing speed to display the scene in real time.

View Manipulation

View manipulation is an important task in 3D design. Direct manipulation techniques provide highly effective visual feedback. Until recently, these direct manipulation techniques have been neglected. View manipulation was included in SEP(3D) to acknowledge the importance of this task. Tasks such as zooming and panning were not included to keep view manipulation simple and comparable across a variety of interfaces. However, it is not clear that these limitations were necessary. Future instruction booklets should include a wider range of viewing tasks.

FUTURE RESEARCH

More commercial 3D systems should be evaluated using SEP(3D). A collection of design guidelines will evolve for improving interfaces. In addition, similar usability studies should be performed during the design of new software. Changes can be made early in the design phase at a low cost.

Reliable covariate measures would be useful in reducing the effect of subject differences during evaluations. However, this study was not able to identify any reliable
measures despite trying several measures which were presumed to be useful. Additional covariate measures should be considered and evaluated.

CONCLUSION

SEP(3D) successfully provides an objective and reliable picture of 3D interaction techniques. Results from two experiments indicate that SEP(3D) results are reliable and valid. Results were used in a successful iterative design and evaluator observations were used to identify why significant differences may have occurred.

SEP(3D) tasks represent atomic actions performed in a variety of 3D applications. These atomic tasks are components of more complex tasks. Performance of the atomic tasks indicate how effective the interaction techniques will be when performing more realistic tasks. SEP(3D) differs from more traditional experimental designs that require subjects to perform the same task many times to determine peak performance. SEP(3D) subjects perform a series of tasks that build upon previous tasks. This provides an accurate indication of how techniques work with respect to the entire system. Performing a single task over and over again will not provide the same understanding of the overall system.

The user interface discipline is only at the beginning of understanding 3D interaction. Formal methods such as SEP(3D) allow researchers to increase the body of factual knowledge regarding 3D interaction. Future systems can be designed on proven theories and guidelines rather than on uninformed intuition.
APPENDIX A

SEP(3D) INSTRUCTION BOOKLET
Evaluation of 3D Interaction Techniques

Instruction Booklet

Scott Grissom

The Ohio State University
Department of Computer and Information Science
January 30, 1992
Instructions

Introduction
You are about to perform a series of tasks to help evaluate the user interface of this 3D graphics system. This is not a test of your skills but an evaluation of the computer system. If something is particularly difficult to do then there is something wrong with the system and not with you!

The first session will be practice so that you can become familiar with the instructions and the tasks which you will be performing. Feel free to ask any questions along the way. After the practice session you will complete the tasks three more times. The sessions will be recorded on videotape and it is therefore important that you work as quickly as you can without sacrificing accuracy.

Tasks
These are the tasks that are being evaluated:

- **create:** create an object of certain size in a particular location.
- **translate:** move an object to a new 3D position.
- **rotate:** change the orientation of an object about its centroid.
- **size:** change the size of an object along one or more dimensions.
- **view:** change the view of the scene.

The instructions lead you through a sequence of tasks which is intended to represent a freeform design session. You will be creating objects, moving them about and changing their relative sizes. You will also be moving the view of the scene to better understand your 'design' as it evolves.

Since you will be performing a limited number of tasks, you can disregard any of the other features of your system. Feel free to use any short cuts to complete each task. The idea is for you to complete the tasks as easily and quickly as possible.
Instruction Sheets

Each task instruction is on a separate page. At the top of each page is the task number. There are a total of twenty tasks in each session. Each page has a short description of the task to perform and two pictures of how the scene should appear before and after completing the task. The object being referred to in each instruction is highlighted in both pictures for easy reference. Use the pictures merely as a guide. Do not be overly concerned that the computer screen look exactly like the instruction sheet.

Starting Position

Each session should begin with a blank scene with no existing objects. The initial view should be adjusted so that you are looking down at the origin at approximately a forty-five degree angle. The origin of the scene should be in the center of the screen with approximately twenty ‘units’ of space on the left and right. Use your best judgment for the size of one ‘unit’.

Instruction Booklet Position

Position this instruction booklet so that you can easily read the instructions and does not interfere with using the system.

Default Settings

If possible, you should preselect any default settings for your system which will make the completion of tasks easier. For example, you will only be creating cube-like objects. As another example, all rotations will be performed about the object’s centroid. Other settings may occur to you after completing the practice session. Remember, the idea is for you to complete the tasks as easily and quickly as possible.
Instructions

Follow the instructions carefully and be sure to complete each task before turning the page to read the next one. Please read each instruction out loud before starting the task. This indicates when you have started the task and turning the page indicates when you have completed the task.

The instructions are intentionally nonspecific. You should use the pictures as a general guide to determine object sizes and relative positions. There are no right or wrong interpretations. Just remember that these tasks are intended to represent a typical freeform design session in which you are ‘playing’ with the objects.

Questions

Please refrain from asking questions while performing the tasks (except during the practice session).

Do you have any questions before beginning?
Task #1

Create an object of size 3x3x3 in the position as shown.

Before:

After:
Task #2

Create a second object of size 3x3x3, in the position as shown.

Before:

After:
Task #3

Reduce the width and depth of the object but maintain the current height.

Before:

After:
Task #4

Rotate the object forty-five degrees about the vertical axis.

Before:

After:
Task #5

Reduce the width and depth of the object but increase the height.

Before:

After:
Task #6

Move the object.

Before:

After:
Task #7

Create a new object 4x4x1 in the position as shown.

Before:

After:
Task #8

Change the view.

Before:

After:
Task #9

Place the back object on top of the front object.

Before:

After:
Task #10

Change the view.

Before:

After:
Task #11

Create a new object 3x3x3 in the position as shown.

Before:

After:
Task #12

Rotate the object forty-five degrees about the horizontal axis.

Before:

After:
Task #13

Change the view.

Before:

After:
Task #14

Move the object onto the grid.

Before:

![Before Image]

After:

![After Image]
Task #15

Rotate the object forty-five degrees about the vertical axis.

Before:

After:
Task #16

Rotate the object forty-five degrees about the horizontal axis.

Before:

After:
Task #17

Increase the width and depth of the object but maintain the current height.

Before:

After:
Task #18

Change the view.

Before:

After:
Task #19

Increase the width, depth and height of the object.

Before:

After:
Task #20

Move the object.

Before:

After:
Stop !
APPENDIX B

SEP (3D) USER SATISFACTION QUESTIONNAIRE
SEP(3D) Questionnaire

Instructions

Subject # ______

1) Each of the following sections are designed to evaluate a single task. The section begins with a description of the task and all questions within the section refer to that task.

2) Each question has a series of numbers from 0 to 7. This is a sample question.

Chocolate chip cookies

bad
0 1 2 3 4 5 6 7
good

3) Circle the number which best describes your reaction to the question. For example, if chocolate chip cookies are your favorite food of all time then you would circle the number 7. However, if chocolate chip cookies are just okay but nothing special, you might circle 3. Circle any one of the numbers which best describes your reaction to the question.

4) If you feel that a question is not relevant to the system you are evaluating then write N/A to the side of the question and do not circle any of the numbers.

5) At the end of each section is a question for you to add any comments (good or bad) which you feel would be useful. For example, you may wish to describe a particular problem or you may feel that the questions have not covered a particular aspect which you feel should be mentioned.
Translation

Translation is the task of moving an object from one position to another. The following statements apply only to the task of translation with this system.

1. Moving an object
   
   | difficult | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
   | easy |

2. When you first learned the system, objects moved in the direction you expected
   
   | never | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
   | always |

3. Now that you are familiar with the system, objects move in the direction you expect
   
   | never | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
   | always |

4. Correcting your mistakes
   
   | difficult | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
   | easy |

5. Input device is appropriate for the task
   
   | never | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
   | always |

6. Visual feedback
   
   | confusing | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
   | very clear |

7. Moving an object is an appropriate test (common task) of the system
   
   | inappropriate | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
   | very appropriate |

8. Are there any comments (good or bad) you wish to add regarding the task of translation? Continue on back if necessary.
Rotation

Rotation is the task of changing the orientation of an object about a specific axis. The following statements apply only to the task of rotation with this system.

9. Rotating an object
difficult  easy
0 1 2 3 4 5 6 7

10. When you first learned the system, objects rotated in the direction you expected
never  always
0 1 2 3 4 5 6 7

11. Now that you are familiar with the system, objects rotate in the direction you expect
never  always
0 1 2 3 4 5 6 7

12. Correcting your mistakes
difficult  easy
0 1 2 3 4 5 6 7

13. Input device is appropriate for the task
never  always
0 1 2 3 4 5 6 7

14. Visual feedback
confusing  very clear
0 1 2 3 4 5 6 7

15. Rotating an object about its centroid is an appropriate test (common task) of the system
inappropriate  very appropriate
0 1 2 3 4 5 6 7

16. Are there any comments (good or bad) you wish to add regarding the task of rotation? Continue on back if necessary.
Sizing

Sizing is the task of changing the length of an object along one or more dimensions. The following statements apply only to the task of sizing with this system.

17. Sizing an object
   difficult 0 1 2 3 4 5 6 7
   easy

18. When you first learned the system, object faces moved in the direction you expected
   never 0 1 2 3 4 5 6 7
   always

19. Now that you are familiar with the system, object faces move in the direction you expect
   never 0 1 2 3 4 5 6 7
   always

20. Correcting your mistakes
   difficult 0 1 2 3 4 5 6 7
   easy

21. Input device is appropriate for the task
   never 0 1 2 3 4 5 6 7
   always

22. Visual feedback
   confusing 0 1 2 3 4 5 6 7
   very clear

23. Sizing an object is an appropriate test (common task) of the system
   inappropriate 0 1 2 3 4 5 6 7
   very appropriate

24. Are there any comments (good or bad) you wish to add regarding the task of sizing? Continue on back if necessary.
View Manipulation

View manipulation is the task of moving your ‘eye’ to obtain a particular view of the scene. The following statements apply only to the task of view manipulation with this system.

25. View manipulation
difficult  easy
0 1 2 3 4 5 6 7

26. When you first learned the system, your ‘eye’ moved in the direction you expected
never  always
0 1 2 3 4 5 6 7

27. Now that you are familiar with the system, your ‘eye’ moves in the direction you expect
never  always
0 1 2 3 4 5 6 7

28. Correcting your mistakes
difficult  easy
0 1 2 3 4 5 6 7

29. Input device is appropriate for the task
never  always
0 1 2 3 4 5 6 7

30. Visual feedback
confusing  very clear
0 1 2 3 4 5 6 7

31. View manipulation is an appropriate test (common task) of the system
inappropriate  very appropriate
0 1 2 3 4 5 6 7

32. Are there any comments (good or bad) you wish to add regarding the task of view manipulation? Continue on back if necessary.
Creation

Creation is the task of specifying the size and position of an object. The following statements apply only to the task of creating cube-like objects with this system.

33. Creating an object

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

34. When you first learned the system, objects appeared in the position and of the size you expected

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>never</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>always</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

35. Now that you are familiar with the system, objects appear in the position and of the size you expect

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>never</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>always</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

36. Correcting your mistakes

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

37. Input device is appropriate for the task

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>never</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>always</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

38. Visual feedback

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>confusing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>very clear</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

39. Creating an object is an appropriate test (common task) of the system

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>inappropriate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>very appropriate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

40. Are there any comments (good or bad) you wish to add regarding the task of cube creation? Continue on back if necessary.
General Characteristics

41 - 43. Your overall reaction to the system for the tasks being evaluated

- terrible: 0, 1, 2, 3, 4, 5, 6, 7
- wonderful: 0, 1, 2, 3, 4, 5, 6, 7

- difficult: 0, 1, 2, 3, 4, 5, 6, 7
- easy: 0, 1, 2, 3, 4, 5, 6, 7

- dull: 0, 1, 2, 3, 4, 5, 6, 7
- stimulating: 0, 1, 2, 3, 4, 5, 6, 7

44. First learning to operate the system

- difficult: 0, 1, 2, 3, 4, 5, 6, 7
- easy: 0, 1, 2, 3, 4, 5, 6, 7

45. Object position within the scene

- confusing: 0, 1, 2, 3, 4, 5, 6, 7
- very clear: 0, 1, 2, 3, 4, 5, 6, 7

46. Object position in relation to other objects

- confusing: 0, 1, 2, 3, 4, 5, 6, 7
- very clear: 0, 1, 2, 3, 4, 5, 6, 7

47. System response time

- too slow: 0, 1, 2, 3, 4, 5, 6, 7
- fast enough: 0, 1, 2, 3, 4, 5, 6, 7

Your Computer Experience

48. Number of months experience with computers

0, 3, 6, 12, 24, 24+

49. Number of months experience with this 3D graphics systems

0, 3, 6, 12, 24, 24+

50. Number of months experience with any other 3D graphics systems

0, 3, 6, 12, 24, 24+
APPENDIX C

SUBJECT RESPONSES TO OPEN-ENDED ITEMS
All subject comments from the open-ended items of the questionnaire are presented for each of core task. Although some interesting suggestions are made, this data is not reliable for any meaningful comparisons. However, this type of anecdotal information can be quite useful early in the development stages of a user interface.

SUBJECT COMMENTS ABOUT TRANSLATION

AutoCAD
No comments were made about translation.

Chalk
"There are different ways to move an object." "It is sometimes confusing." "It would be more useful to describe the desired position of an object in relation to other object instead of an XYZ coordinate."

Form*Z
"Straightforward once the concept of moving perpendicular to the reference plane is understood." "It would be useful to indicate a height while in a 3D view without having to switch to an orthographic view."

Modified
"Hard to gauge whether the object was actually resting on another object." "Would be nice to be able to select more than one object at a time." "Would be nice to have objects automatically align themselves with each other when necessary." "Sound feedback would be useful to indicate that an object has been lowered to the ground level." "Moving was easy and clear." "It was difficult to move an object 'exactly' where I wanted it."

MultiMode
"Visual feedback was unclear when an object was being lowered onto another object because the drop shadow does not appear on the lower object."
NewRotate

"Highlighting selected objects in red may be a poor choice for color blind users."
"There was no feedback to indicate that I had not selected when object when i thought I had."

NewSize

"It is easier to move an object with NewSize than with NewRotate."

Original

"Very straight forward." "Easy enough as long as the entire object was selected as opposed to a single face." "Difficult to determine if the 3D pointer was far enough inside the object to select the entire object rather than a single face."

Triad

"Difficult to adjust the height of the pointer." "Difficult to align objects with their surroundings." "Moving an object was difficult."

Wireframe

"Translation is well implemented."

SUBJECT COMMENTS ABOUT SIZING

AutoCAD

No comments were made about sizing.

Chalk

"There are different ways to size and object." "Sometimes it is confusing."
Form*Z

"It is difficult to perform proportional sizing along XY. "This is difficult because actions seem to behave ‘differently’ while in perspective instead of orthographic." “Sizing along three dimensions at once is difficult.”

Modified

"It would be nice to constrain sizing to a single axis." “Highlighting the entire face that is currently selected rather than just the outline would be better visual feedback." “It was sometimes difficult to select the desired face or vertex.” “When sizing a single face, it would have been helpful to constrain motion to maintain the shape of the cube instead of allowing sheering to create a trapezoidal shape.”

MultiMode

"It is difficult to correct an object after a face has unintentionally be sheared.” "It is difficult to judge the height of an object.”

NewRotate

“There was no feedback to indicate that I had not selected an object when I thought I had.” “Selecting a vertex does not always work.” “There is no need to provide the face selection technique because all sizing operations can be done by selecting a vertex.”

NewSize

“Selecting a face or a vertex should still be an option. Selecting the vertex did not always work very well.” “Rotated objects become distorted when sized.”

Original

“There was only one instance of sizing when the mistake was not easy to correct.” “Sizing along more than one dimension at a time was sometimes difficult.”
Triad

"This was difficult because objects moved below and above the XY plane." "It took practice to correctly select a single face of point." "It is very easy to perform unintended shearing when adjusting the height."

Wireframe

"An undo command would be useful for correcting accidental sheer." "It is difficult to adjust the height of an object without sheering the top face." "Easy to use." "Sometimes the visual cues were confusing when one object was on top of another object."

SUBJECT COMMENTS ABOUT ROTATION

AutoCAD

No comments were made about rotation.

Chalk

"There are different ways to rotate an object." "Sometimes it is confusing." "It becomes difficult to rotate an object when you are not looking down the y-axis." "Relative rotation should be provided instead of absolute. I would prefer to indicate a rotation of forty-five degrees from the existing orientation rather than have to remember that the object is currently at 210 degrees and the new position should be 255."

Form*Z

"Picking an exact center of rotation is difficult because it is done by hand." "Difficult to grab the correct handle to perform a desired rotation." "It is especially difficult in perspective." It would be useful to enter a specific degree angle rather than adjust by hand.
Modified

"I would like type the amount of rotation instead of using the mouse.” “It was difficult to limit rotation to a single axis.” “It would be nice to allow rotation about a single axis.” “It was difficult to do a single axis rotation with the mouse. Separate knobs for each axis would be better.”

MultiMode

"It is difficult to perform the desired rotation.” “The mouse was difficult to use for rotation.”

NewRotate

"Using a knob would be more natural than the mouse to rotate an object.” “It is sometimes difficult to select the correct axis.”

NewSize

"It is difficult to understand the axis of rotation with respect to mouse movement.”

Original

"Difficult to accurately rotate about a single axis. “Difficult to select correct point for the desired rotation.” “Difficult to tell where to place the pointer in order to rotate in the desired direction.

Triad

"Difficult to restrict rotation to a single axis.” “A little more practice would make this relatively easy.” “Difficult to predict the direction the object would rotate.”

Wireframe

"Using a knob to control rotation would be less confusing than the mouse.” “Rotation was the most difficult part of the system.” “Hand motion and desired rotation direction do not agree.” “Takes a while to get used to but eventually becomes straightforward.”
SUBJECT COMMENTS ABOUT VIEW MANIPULATION

AutoCAD

No comments were made about view manipulation.

Chalk

"Real time view manipulation is needed that is driven by the current aspects of the view rather than global coordinates." "Very difficult and confusing to determine the necessary position of the eye for a desired view."

Form*Z

"Selecting preset views is easier to use than the direct manipulation technique."

Modified

"Direction of rotation was easy to control but if zooming were added it would be much more difficult to control." "The knob used for controlling the cursor height should also be used to tilt the table and the mouse would continue to rotate the table." "There may be times when other view angles would be necessary but for these tasks, the views seemed adequate."

MultiMode

"It would be nice to have the adjustments of rotation and tilt be controlled with separate input devices." "For a new user, using three knobs to control movement might be easier than using a mouse."

NewRotate

"Large movements were difficult to control. I felt as if I lost control." "I often used the left mouse button to select an object which caused an unintended view manipulation. I would use the right button for view manipulation and the left button for everything else." "This is the hardest task to get used to." "A knob would be more natural than the mouse."
NewSize

“I sometimes moved the mouse in a circular direction to rotate the table but only a horizontal mouse movement was necessary.”

Original

“It might be more useful if thumbwheels were used for manipulation instead of the mouse.” “Very touchy, needs to be either slowed down or performed in increments.” “Forced to keep picking new points to correct a bad rotation.” “Sometimes the pointer would get lost in 3-space and it was difficult to find it.”

Triad

“The direction of the mouse movement was too touchy and made it difficult to make adjustments.” “View manipulation was quite intuitive but the pointer was ‘lost’ once and it was difficult to locate.” “A very nice feature but it can sometimes be difficult to keep all directions in order.”

Wireframe

“Using a knob to control view manipulation would be less confusing than the mouse.” “Unintended adjustments often occurred because it is difficult to move the mouse along a single axis.” “Straight forward.”

SUBJECT COMMENTS ABOUT OBJECT CREATION

AutoCAD

No comments were made about object creation.

Chalk

“It is difficult to create objects other than simple cubes. It is also difficult to change the initial size of the object.” “A mouse is not a convenient input device for expressing a 3D thought.”
Form*Z

"This is one of the easiest and most intuitive tasks." "It is difficult to judge where the object would be placed in relation to the grid or other objects."

Modified

"The cursor height could not be precisely determined." "It would be nice to be able to copy an existing cube and already be in move mode to reposition the new cube." "I would like a scale of the 3D cursor to indicate the current height."

MultiMode

"It is difficult to determine the correct height of the object."

NewRotate

No comments were made about creation.

NewSize

"Provide a scale on the 3D pointer to indicate the current height"

Original

"I would like to see some feedback on the height of an object." "Creation was very simple."

Triad

"It is difficult to adjust the pointer height." "The pointer height was difficult to control which made most tasks more difficult."

Wireframe

"Visual cues were confusing if the object was initially created above the XY plane." "The grid made this task easy to perform."
LIST OF REFERENCES


