

2015

# Natural user interfaces for interdisciplinary design review using the Microsoft Kinect

Anastacia Maria MacAllister  
*Iowa State University*

Follow this and additional works at: <http://lib.dr.iastate.edu/etd>

 Part of the [Mechanical Engineering Commons](#)

---

## Recommended Citation

MacAllister, Anastacia Maria, "Natural user interfaces for interdisciplinary design review using the Microsoft Kinect" (2015). *Graduate Theses and Dissertations*. 14482.  
<http://lib.dr.iastate.edu/etd/14482>

This Thesis is brought to you for free and open access by the Graduate College at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

**Natural user interfaces for interdisciplinary design review using the Microsoft Kinect**

by

**Anastacia Maria MacAllister**

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Mechanical Engineering; Human Computer Interaction

Program of Study Committee:

Eliot Winer, Major Professor

Matthew Frank

Adarsh Krishnamurthy

Iowa State University

Ames, Iowa

2015

Copyright © Anastacia Maria MacAllister, 2015. All rights reserved.

## TABLE OF CONTENTS

LIST OF FIGURES .....	iii
LIST OF TABLES .....	iv
ABSTRACT .....	v
CHAPTER 1: INTRODUCTION .....	1
Purpose of Work .....	1
Motivation .....	1
Thesis Organization .....	3
References .....	4
CHAPTER 2: BACKGROUND .....	5
Teams and Technology .....	5
Benefits of Virtual Reality .....	7
Natural User Interfaces .....	9
Natural User Interfaces for 3D Manipulation .....	12
Low Cost Interaction Devices .....	14
Cognitive Modeling Using the GOMS Family .....	17
References .....	20
CHAPTER 3: A NATURAL USER INTERFACE FOR IMMERSIVE DESIGN REVIEW .....	24
Forward .....	24
Abstract .....	24
Introduction .....	25
Background .....	27
Teams and Technology .....	28
Benefits of Virtual Reality .....	29
Natural User Interfaces .....	32
Methodology – Program Overview .....	34
Teamcenter Visualization Mockup .....	37
Mockup VisController .....	39
Command Structure .....	39
Motion Data .....	40
VisMockup Communicator .....	41
Kinect .....	43
Conclusion .....	43
Future work .....	44
References .....	44
CHAPTER 4: ASSESSING THE FEASIBILITY OF MANIPULATING ENGINEERING GEOMETRY USING A MICROSOFT KINECT AND COMPUTER-AIDED ENGINEERING SOFTWARE .....	48
Abstract .....	48
Introduction .....	49
Background .....	50
Teams and Technology .....	51
Benefits of Virtual Reality .....	51
Natural User Interfaces .....	53
Natural User Interfaces for 3D Manipulation .....	54

Low Cost Interaction Devices .....	56
Cognitive Modeling Using the GOMS Family .....	56
Methodology – Program Overview .....	58
Head Tracking .....	64
Methodology – Interface Evaluation .....	66
Results – Interface Evaluation .....	69
Conclusions .....	75
References .....	76
<b>CHAPTER 5: GENERAL CONCLUSIONS.....</b>	<b>81</b>
<b>CHAPTER 6: FUTURE WORK.....</b>	<b>82</b>



## LIST OF FIGURES

Figure 1: Program Component Diagram.....	35
Figure 2: Left Click Gesture.....	35
Figure 3: Left Swipe Gesture .....	36
Figure 4: Immersive Wand Interacting With Immersive Menu .....	36
Figure 5: Head Tracking View Adjustment .....	37
Figure 6: Command Structure .....	40
Figure 7: Command Package Types .....	41
Figure 8: Mockup Interface .....	60
Figure 9: Program Information Flow Diagram .....	61
Figure 10: Left Click Gesture.....	61
Figure 11: Left Swipe Gesture .....	62
Figure 12: Immersive Wand Interacting With Immersive Menu .....	63
Figure 13: Head Tracking View Adjustment .....	65
Figure 14: NGOMSL Task Analysis .....	68

## LIST OF TABLES

Table 1: Activate Immersive Menu Timings by Method.....	69
Table 2: Rotate Model CW Timings By Method .....	71
Table 3: Activate Wand Tracking Projected NGOMSL Task Times by Method .....	72
Table 4: Task Completion Time Comparison .....	73

## ABSTRACT

As markets demand engineered products faster, waiting on the cyclical design processes of the past is not an option. Instead, industry is turning to concurrent design and interdisciplinary teams. When these teams collaborate, engineering CAD tools play a vital role in conceptualizing and validating designs. These tools require significant user investment to master, due to challenging interfaces and an overabundance of features. These challenges often prohibit team members from using these tools for exploring designs. This work presents a method allowing users to interact with a design using intuitive gestures and head tracking, all while keeping the model in a CAD format. Specifically, Siemens' Teamcenter® Lifecycle Visualization Mockup (Mockup) was used to display design geometry while modifications were made through a set of gestures captured by a Microsoft Kinect™ in real time. This proof of concept program allowed a user to rotate the scene, activate Mockup's immersive menu, move the immersive wand, and manipulate the view based on head position.

This work also evaluates gesture usability and task completion time for this proof of concept system. A cognitive model evaluation method was used to evaluate the premise that gesture-based user interfaces are easier to use and learn with regards to time than a traditional mouse and keyboard interface. Using a cognitive model analysis tool allowed the rapid testing of interaction concepts without the significant overhead of user studies and full development cycles. The analysis demonstrated that using the Kinect™ is a feasible interaction mode for CAD/CAE programs. In addition, the analysis pointed out limitations in the gesture interfaces ability to compete time wise with easily accessible customizable menu options.

## CHAPTER 1: INTRODUCTION

### Purpose of Work

The overall goal of the work is to fuse low cost commodity tools and an existing computer-aided design (CAD) package to quickly evaluate new interaction concepts that go beyond the mouse and keyboard for engineering design software. For the work, the prototype system is evaluated for feasibility against traditional interaction modes using the natural goals, operators, methods, and selection language (NGOMSL) to avoid a costly user study on a proof of concept system.

### Motivation

Designing and validating new products like aircraft and machinery rely heavily on engineering tools. Tools like computer aided design (CAD) and computer aided engineering (CAE) often drive the design process and are integral in product development [1]. While these tools are very effective at creating, modifying, and evaluating designs, they can be very cumbersome to use [2]. Low cost commodity interaction devices like the Kinect™ and Leap Motion™ could help improve CAD and CAE tools by introducing user centered interface design principles seen traditionally in virtual reality [3 – 8].

Combining low cost interaction devices with engineering tools has the possibility to simplify interfaces, improve user interaction, increase understanding of designs, and allow increased participation in the design process [9, 10]. However, incorporating low cost interaction devices into CAD and CAE does introduce fundamental changes to the user experience. Change of this magnitude requires improvements in small stages, with continuous

evaluation along the way. Towards this end, creating a proof of concept prototype systems allows identifying, testing, and refining key interface elements for engineering software that go beyond the mouse and keyboard. Evaluation throughout the process in stages saves time and increases the likelihood of a system that is ready for full user testing.

The commodity interaction tools market is extremely fluid with new devices rapidly entering the market. As a result, potential devices need to be evaluated quickly. Timely evaluation is required to match pace with the new ideas continually introduced. Evaluation of devices will aid in generating requirements for creating a system that shifts to a new interaction paradigm for CAD and CAE, beyond the mouse and keyboard.

Along with traditional mouse and keyboard interfaces there is also the traditional mode of evaluation. Evaluation is commonly performed on fully functioning systems. Building and evaluating a fully functioning system, however, requires significant resources to program, test, recruit users, and analyze. By the time any testing or analysis is done changes can become expensive and time consuming. In addition, with commodity devices, any lengthy evaluation becomes obsolete. By the time results are prepared, a new model, often with many new features, is available. The goal of quick evaluation necessitates building and testing programs in a short amount of time. To accomplish this the authors advocate building proof of concept prototype systems with a few key features and evaluating them using existing usability methods. For concept evaluation the authors focused on cognitive modeling methods. Cognitive modeling provides the opportunity to quickly and cheaply evaluate concepts, ensuring the ability to evaluate a wider range of options than with traditional methods [11].

Overall, engineering CAD tools play a central role in designing and evaluating products. However, the ease of use of these tools is generally poor and limited to dated mouse and keyboard interaction. The work presented in this paper evaluates in terms of usability a proof of concept solution that integrates low cost interaction devices and CAD/CAE using cognitive modeling tools using a fast and agile method.

### Thesis Organization

For the work presented in this thesis there are two main contributions. The first involves developing a proof of concept prototype system that integrates the requirements outlined above. The second contribution is the usability analysis of the system compared with the existing engineering software user interface.

Chapter 3 is a paper written on the work and published in the ASME IDETC/CIE 2014 conference in Buffalo, New York. The paper focuses mainly on the technical development of the prototype system and the devices used. In this section, there is also extensive background on the justifications for each component required for the program and how it adds the end goal of a usable prototype.

Chapter 4 is a paper submitted to The Journal of Computing and Information Science in Engineering (JCISE). This section focuses mainly on the task analysis method used to evaluate the prototype system against the current engineering software user interface. This chapter also provides supporting evidence and justification for the gesture to function mapping. Chapter 5 discusses the conclusions of the work and Chapter 6 future work on the prototype.

## References

- [1] Mujber, T.S., Szecsi, T., Hashmi, M.S.J., 2004, "Virtual Reality Applications in Manufacturing Process Simulation," *Journal of Materials Processing Technology*.
- [1] Wang, L., Shen, W., Xie, H., Neelamkavil, J., Pardasani, A., 2002, "Collaborative Conceptual Design—State of the Art and Future Trends", *Computer-Aided Design*, 34(13), pp. 981-996.
- [2] Stacey, M., and Claudia E, 1999, "CAD System Bias in Engineering Design," *Proceedings of the 12th International Conference on Engineering Design*, Vol. 2.
- [3] Yeo, H., Lee, B., Lim, H., 2013, "Hand Tracking and Gesture Recognition System for Human-Computer Interaction Using Low-Cost Hardware," *Multimedia Tools and Applications*, pp. 1-29.
- [4] Hartson, H.R., 1998, "Human-Computer Interaction: Interdisciplinary Roots and Trends," *Journal of Systems and Software*, 43(2), pp. 103-118.
- [5] Shackel, B., 1997, "Human-Computer Interaction—Whence and whither?," *Journal of the American Society for Information Science and Technology*, Vol. 48, pp. 970-986.
- [6] van Beurden, M. H., IJsselsteijn, W. A., & Hopf, K., 2011, "User centered design of gesture-based interaction technology," In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, (pp. 1-4). IEEE.
- [7] Fidopiastis, C. M., Rizzo, A. A., & Rolland, J. P., 2010, "User-centered virtual environment design for virtual rehabilitation," *Journal of Neuroengineering and Rehabilitation*, 7(11).
- [8] Hix, D., Swan, J. E., Gabbard, J. L., McGee, M., Durbin, J., & King, T., 1999, "User-centered design and evaluation of a real-time battlefield visualization virtual environment," In *Virtual Reality, 1999. Proceedings*, (pp. 96-103). IEEE.
- [9] Jaimes, A., Sebe, N., 2007, "Multimodal human-computer interaction: A survey, *Computer Vision and Image Understanding*," Vol. 108, No. 1-2, pp. 116-134.
- [10] Pavlovic, V.I., Sharma, R., Huang, T.S., 1997, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: a Review," *Pattern Analysis and Machine Intelligence*, 19(7), pp. 677-695.
- [11] Li, X., Gunal, M., 2012, "Exploring Cognitive Modeling in Engineering Usability Design," *Journal of Engineering Design*, 23(2), pp. 77-97.

## CHAPTER 2: BACKGROUND

Previous research relating to the work in this paper is distributed into six main areas: 1) research on team use of technology, 2) the benefits of VR, 3) natural user interfaces (NUIs), 4) gestures for 3D manipulation, 5) low cost interaction devices and 6) Goals, Operators, Methods, and Selection cognitive modeling. Work in each of these areas influenced the building and evaluation of the immersive low cost user-friendly design review prototype presented in this paper.

### Teams and Technology

Design teams today are increasingly made up of interdisciplinary backgrounds. Swink et al. points out that in today's world concurrent design is needed to meet customer demands and to keep a company profitable in the fast paced international market [1]. As a result, companies are favoring concurrent design, leading to interdisciplinary teams. These teams need effective tools that allow them to communicate and understand each aspect of the design process [2], [3], helping them review designs and assemble strategies to pinpoint potential roadblocks.

The necessity for tools of this nature has not escaped researchers. There exist numerous examples of research looking for ways to help teams collaborate and communicate. Liveboard is an example of one such tool. It was a device developed to display information on a larger more input friendly screen [4]. Use of the device was targeted at situations where simple desktop interactions fell short. Results point to people viewing these devices as helpful and worthwhile tools. However, the researchers found while people want a device that helps them communicate, if using the device is challenging they are not willing to spend time trouble shooting in a group environment.



Bouchlaghem et al. presents an example of tools helping interdisciplinary teams in Architecture, Engineering and Construction [5]. Their research focused on an integrated visualization tool for project information, such as blue prints, CAD models and design mockups. They found that a common tool helped conceptual designers work collaboratively and communicate ideas. The tool facilitated planning amongst the groups and served as a consulting tool in meetings with customers. In the end, the easy to use tool helped bridge the gap between designers and builders.

Work presented by Zhong et al. details a system that helps collaborators exchange ideas and coordinate their efforts on a project [6]. The use cases involving the system show its value at aiding communication between team members. The system proved helpful at preventing conflicts during design conceptualization phases though product implementation.

Based on previous work, it is evident researchers recognize the need for tools that help facilitate interactions with design information. Research points to the importance of making tools intuitive and accessible. This involves moving beyond mouse and keyboard interaction for engineering tools. Identifying and investigating elements of a program that help users escape the limited traditional interaction approach will lower the barrier of entry to effective design review with CAD and CAE tools. As more research is performed the resultant outcomes will allow recommendations to be made regarding the meshing of low cost tools and engineering design programs.

## Benefits of Virtual Reality

What makes VR attractive to academia and industry is the features and experiences that are difficult or impossible to replicate in the physical world. Examples include, exploring how a new aircraft design will fit on an existing carrier before building a physical prototype, virtually walking through a factory to identify design flaws before breaking ground, or practicing a dangerous or complex assembly process before ever stepping in a work cell. These types of interactions can be invaluable to a team of product designers. It allows them to see and experiment with their design in ways that are time consuming, cost prohibitive or impossible to enact in real life. These abilities, mixed with CAD and CAE, provide enormous potential. Berta [7] takes an in-depth look at the potential benefits to industry if VR and CAD / CAE integration becomes reality. These include: 1) using VR principles to create simpler interfaces for casual users, 2) receiving CAD accuracy along with part information, and 3) the immersion and sense of “presence” that comes with VR. Immersion is defined as the perception that one is completely enveloped by or pulled into the virtual environment and presence is the feeling of physically being in the virtual environment [8, 9]. However, implementing CAD / CAE and VR integration proves difficult. Challenges listed by Berta mainly stem from the transfer between CAD and VR packages. These challenges include loss of geometry, topology, semantics and behaviors.

In addition to the challenges with conversion, there is also the challenge of VR adoption. While VR can be very beneficial to many divisions within a company [10], these systems are traditionally large and costly installations that serve a specific purpose. This is a considerable barrier for VR adoption. Even a large company can only afford a limited number of these

systems, not enough for the numerous interdisciplinary teams that could benefit. One of the main goals of these large expensive systems is immersion. Bowman and McMahan find that large scale VR provides nearly full immersion, but in some cases full immersion is not always necessary to reap the benefits of VR [11].

McMahan et al. further investigates immersion and the link it has to six degree of freedom (6 DOF) manipulation tasks. In a study conducted the level of immersion is controlled by turning stereo on or off and using one or three computer assisted virtual environment or CAVE™ walls. Participants were asked to perform a task using one of three interaction devices while head tracked. They found interaction devices have more influence on participant completion time and errors than immersion level [12]. In their discussion they interpret these results as meaning that one can receive the same benefits of large VR systems using less costly displays combined with head tracking and 6 DOF input devices. While the results conclude that full immersion benefits are not tied to large-scale VR environments, they neglect to pinpoint what degree of immersion is enough. They also do not discuss what components besides stereo and number of screens contributes to increasing user presence in a system.

While a full VR system is not necessary to provide the user with a sense of immersion, certain components of VR do need to be present. Barfield finds that head tracking plays an important role in the presence of users in a system [13].

Gruchalla provides another look at how VR can help with immersion [14]. In the study, participants planned a new well path in a mature oil field. Participants did this using either a stereoscopic desktop display or a head tracked stereoscopic cave automatic virtual environment (CAVE™) display.

Background research on VR suggests lower cost systems can provide users with some of the same immersive benefits seen in large-scale systems. As long as users can manipulate and interact with models, they can build a more sound mental model of complex 3D parts. The guidelines of immersion and low cost VR play a central role in the method development of the proof of concept program presented in this paper.

### Natural User Interfaces

As pointed out by McMahan et al. the interaction mode can be just as important as the display hardware. Some of the biggest detractors to CAD / CAE program usability stems from interface deficiencies. As natural user interfaces (NUIs) have become commonplace, much research has been done to examine their potential.

Kosmadoudi et al. proposes using games to help users better understand software programs [15]. But Ju et al. takes a step closer to addressing the need for improving interaction between the user and the program [16]. They ran participants in a study evaluating a number of physical objects like springs, joysticks, and a gyrobox that were assigned to functions in the modeling software like rotate, zoom, and extrude. After using the tools participants were asked a series of questions about how they perceived the usability of the tools.

Francese et al. published research using the Wiimote and Kinect, both low cost devices, to help people explore Bing maps [17]. Participants in the study navigated maps and their reactions were recorded. Results of the study showed that motion control devices increased the users' sense of immersion and presence. In addition, participants who used the low cost devices adapted to an expert level of control, moving around the map faster than other users.

Shratuddin and Wong studied limitations of standard interaction techniques and identified weaknesses in current design processes [18]. They point out that current graphical user interfaces (GUIs) limit creativity and lock users into a linear design process. To combat this, the researchers developed a program that allowed a user to collaboratively design and manipulate components in a virtual world using intuitive gestures.

Work by Purschke et al. is an example of an engineering application used at Volkswagen to aid in the conceptual design of cars [19]. The program uses gestures, through data glove inputs, to manipulate views of a 3D model as well as change material properties.

Fiorentino et al. 2012 took an in depth look at NUIs. The work highlighted the difficulty and complications associated with CAD interfaces. They set out to create a system that let a user explore a CAD model through augmented reality and technical drawings. The authors performed a user study validating that NUIs help users become proficient and comfortable with a software program more quickly than without [20].

Tumkor et al. recognizes the ability of low cost interaction devices to improve interaction and collaboration in design [21]. For the work they use two Kinect sensors to allow users to explore and dismantle SolidWorks™ CAD models using hand poses. To evaluate the system the researchers conducted a user study measuring task completion time to constrain and assemble a design. From the study they found that performing some tasks through the SolidWorks™ interface were faster using gestures, but for the majority of tasks a mouse was still the most efficient. Based on the study the researchers concluded that hand-gesture recognition systems do hold promise for interacting with CAD systems, however user interface

changes are necessary to free the user from the traditional menu navigation in order to reap the benefits of gestures.

Nanjundaswamy et al. creates a system that incorporates gestures, brain-computer interface, and speech to make interactions with CAD systems more intuitive [22]. They combine these three elements to move interaction with CAD systems beyond a standard mouse and keyboard. To evaluate the system they conduct a preliminary human factors study using their interface in Google Sketchup. For the study they looked at the effort required to create the model, noted the users preferred way to accomplish a task, and recorded limitations encountered by users.

Song et al. developed a system for interacting with CAD models using gaze and finger control [23]. The system is aimed at tackling one of the drawbacks to gesture systems, user fatigue. To build and test their system they identify three primary CAD tasks translation, rotation, and zooming. From here they carry out user testing measuring the time to acquire information and task completion time. At the end of the study they also conduct user interviews to gauge the intuitiveness of the system compared with traditional mouse and keyboard interaction. Results for task completion time were mixed; some tasks were faster using a mouse but others the gestures were more efficient. Results from the interviews suggested that users found the gestures much more intuitive for carrying out the tasks compared to the mouse and keyboard. However, in terms of comfort users on average reported that they found the mouse and keyboard interaction to be more comfortable.

This research illustrates how CAD and CAE programs would benefit from NUIs incorporating low cost, commodity interaction technologies. Towards this goal of accessibility

NUIs are promising, which is why they are the basis for the proof of concept interface presented.

### Natural User Interfaces for 3D Manipulation

With the introduction of devices like the Kinect<sup>TM</sup> and the increase in computing power, the hardware for gesture recognition has become commoditized. The potential of this technology has not gone unnoticed by the research community. Unfortunately though, researchers who create these systems tend to focus on high recognition rates and do not consider the usability of gestures [24]. For the work presented, the authors drew on previous publications in 3D gesture interaction to guide the development of a gesture based user interface (UI). Based on the literature review, gestures were assigned to actions accordingly, taking into account the needs of the user in terms of understanding and usability.

Lee et al. looked at developing gestures to control content displayed on a television screen [25]. They used Wizard-of-Oz studies to develop gesture sets and allow user evaluation of gestures. Wizard-of-Oz studies prompt participants to perform a certain action, in this case a gesture, to complete an assigned task. The Wizard portion of the study involves the facilitator who actually completes the action the user was instructed to using the standard interface. Wizard-of-Oz studies for the work by Lee et al. allowed them to focus on the user interaction component and not be limited by the technology available. Though their studies they found that eliciting user input in this manner provided helpful insight for the creation of more intuitive gestures, by focusing more on how users would complete a task and less about how well a system would capture a certain gesture.

Park et al. investigated using gestures for manipulating models on screen [26]. A custom viewer was built to load geometry and allow a user, through a simple set of gestures, to manipulate models. They did not do a formal user study on the gestures or illicit user feedback. However, the gestures they used for their system were in line with recommendations in research literature, especially for the rotate gesture.

Sabir et al. recognized the need for simpler more intuitive interfaces when viewing 3D models [27]. They developed a system that allowed biologists to explore molecular structures. The researchers conducted a user study that elicited usability feedback on the system. Overall, the users were receptive and enthusiastic. However, from the results the researchers found that switching “modes” in the software was challenging for novice users.

Fikkert et al. researched gestures for use with large displays [28]. Their study found that users consider tapping or clicking to trigger a feature an intuitive gesture. Results also indicated that to keep interfaces simple for users duplicate gestures for activating and deactivating a feature is recommended.

Gallo et al. explored using free hand navigation to control medical imaging data using a Kinect™ [29]. In their work they used gestures for manipulating medical models. Gestures used by Gallo et al. to manipulate the medical models were similar to other gestures seen in literature for manipulation tasks such as rotate and zoom.

The literature review above on interaction modes for this paper focused on free hand 3D object manipulation work. This ensured that gestures used in the researchers program were applicable to the action assigned to the gesture and intuitive for users. Work by the surveyed authors reinforced the decisions made by the researchers on assigning gestures to functions in



the prototype program presented. Elements incorporated in this work from previous research aims to ensure the gestures match what users expect interaction wise from a program for specific functions.

### Low Cost Interaction Devices

Several commodity low cost devices were considered for the prototype developed. These devices were placed under consideration due their helping accomplish the overall goals laid out in the introduction section. Devices considered were the Kinect™, Wimote™, and the Leap Motion™. Discussed below are the device specifications and how they relate to the project. For each device, characteristics that further the project goals are highlighted while the detracting features are also discussed.

The Wiimote™ released with the Nintendo Wii™ gaming system in 2006 [30] is considered a pioneer in the motion gaming market. The device contains an accelerometer, D-pad, gyroscope, and infrared sensor. Feature wise the device shared many of the same characteristics as traditional gaming controllers, but came with the additional infrared (IR) sensor for motion capture gaming. The motion-capturing feature of the device relies on an IR emitter bar and a IR receiver implanted in the controller. With the hand held controller the user can make motions, which are then represented on screen. This is done through the controller receiver relationship. The sensor bar contains a number of infrared lights and based on the distance and rotation of these seen by the controller the position relative to the emitter bar can be calculated. This data allows for the transforming of player body movements into on screen action. While pioneering when first released the Wiimote™ technology compared to others

available is somewhat dated and limited. Holding a physical controller greatly limits interaction options, which is not ideal for the work.

The Microsoft Kinect™ was released in 2010 as a controller less gaming device and heralded as the future of computer based interaction [31]. Version one of the device for Windows was made up of RGB and IR cameras. The Kinect™ version one has a maximum camera resolution of 1280x960 and captures at a maximum of 30 frames per second [32]. This allowed the device to make out user movements in a zone approximately 4 to 8 meters away from the camera. Initially the Kinect™ was only released for the Xbox gaming device, however due to its popularity a Windows version was released along with a developer SDK. Not long after releasing the device developers gained direct access to the information recorded by its RGB and depth cameras. This opened up a number of options. No longer were developers limited to the Microsoft SDK. With this freedom developers set off to utilize the Kinect™ for purposes beyond gaming. The low cost of the device, \$150, made it ideal for research. Developers set to work creating programs that could recognize various gestures and other user actions. These kits were released to the public or were made available for purchase. The ease of developing with the Kinect™ made it popular not only in research but in independent developer circles as well. As a result, the Kinect™ provided considerable developer support with a range of SDKs to select from. The distance and controllerlessness of the Kinect™ lent itself well to large group spaces. In addition, the capture range allowed for a number of gesture options. The device could be used to capture full body or arm gestures. However, one detractor to the Kinect™ is its limited resolution. While the device is proficient at collecting large

sweeping motions, its limited resolution prevents capturing finer hand motions. This can create a problem with prolonged use, since the user can become fatigued.

The third device under consideration for the research presented in this paper was the Leap Motion™ [33]. The device was first released to the public in 2013 and retails for \$79.99. The LEAP Motion is a desktop device that combines optical and IR tracking to capture a users hand and finger movements with surprising precision [34]. Unlike the other two devices mentioned above, the LEAP has the ability to capture fine hand and finger movements, allowing more precise control. However, this increased accuracy comes with tradeoff in tracking area. While devices like the Wiimote and Kinect can capture whole body movements, the LEAP is limited to a desktop size tracking area.

The Microsoft Kinect™ was selected as the low cost immersion device for the proof of concept system after surveying the available devices. Selection of the Kinect™ was made based on the maturity of the technology along with the wide development support base. In addition the ability of the Kinect™ to get designers up and out of the traditional desktop design environment intrigued the researchers. The biggest reservation the researchers had when selecting the Kinect™ was the low resolution. However because Microsoft recognized the market potential of the device and announced plans to continue development in the form of the more accurate second generation Kinect™, the researchers believe that the technology limitations will be short lived. As a result, the researchers determined that the Kinect™ provided the greatest flexibility when trying to design and test a system that branches out from the traditional mouse and keyboard desktop interaction.

### Cognitive Modeling Using the GOMS Family

There are a number of cognitive modeling methods developed by academics and industry alike. One such family of modeling methods adopted by those who work in the engineering field is Goals, Operators, Methods, and Selections (GOMS). The GOMS family of methods is well suited to the type of sequential and hierarchal interfaces found in engineering applications, like CAD / CAE. The GOMS family of models consists of a number of variants. Each variant is geared towards a specific use case.

The GOMS method is popular in engineering circles for evaluating user interfaces. This family of models is fairly extensible. Existing literature uses the model for validating and testing interfaces from text to gestures. The research by Eisert et al. is such an example [35]. For the work, Eisert et al. used the GOMS-KLM method to predict task completion time for different gesture interfaces in an automobile. They used the models to investigate the tradeoffs in completion time based on interface position in the vehicle. Based on the results of the modeling process a hypothesis was developed about where users would prefer the gesture interface. The researchers found that the user study results matched the hypothesis generated by the cognitive model validating the use of GOMS before user testing. This approach allowed designers to test interfaces with minimum time and user resources. They also helped identify deficiencies before users became involved.

The work by Gray et al. illustrates a large case study using GOMS in a business setting [36]. The researchers compared interfaces for telephone company toll and assistance operators as they were investigating claims by operators about a slow and frustrating new interface. After conducting a GOMS analysis they found that the proposed new interface was actually much

slower than the original. The new slow interface would have cost the company two million dollars a year in lost productivity. The study showed the value of cognitive models such as GOMS. If the company had conducted a GOMS analysis they could have prevented the purchase of the software.

While GOMS analysis is a powerful tool, selecting the proper variation is imperative to receive meaningful results. John and Kieras compiled a guide to selecting a GOMS type model based on a modeler's specific needs [37]. This guide aided the work presented in this paper by highlighting the strengths of each model and selection aspects to consider. Traditional GOMS is limited because it only models expert users who are already familiar with the interface, whereas NGOMSL takes into account learning time for novice users. Kieras and the Handbook of Human Computer Interaction detail the steps required when performing a NGOMSL analysis [38, 39].

Parsons et al. uses a variation of NGOMSL to predict task completion time for a gesture user interface between an assistive robot and those with severe spinal cord injuries [40]. The researchers wanted to test out various user interface configurations, and wanted to model the interface to make sure it was as complete as possible before gathering participants. To accomplish this goal they developed a NGOMSL model to test and predict the performance of different UI configurations. They then tested this model against participants. They found their model to be incredibly accurate when compared to the study results. The error between modeled and predicted results was less than ten percent in most cases. From their results, the authors concluded that task analysis models like NGOMSL were accurate predictors of user interface performance when applied in the correct context.

Another example of NGOMSL used for task analysis is by Lee and Koubek [41]. The researchers extended the NGOMSL model to use between multiple users. This work showed the extensibility of the NGOMSL language and its usability beyond standard GUI interfaces.

Mohan et al. uses NGOMSL to model a user interface for soccer playing robots [42]. The researchers took advantage of the learning rate factor in NGOMSL to estimate task completion time for trouble shooting problems with robots. The NGOMSL model helped them review their interface before they put it into use. Modeling user interaction provided an advantage since valuable time evaluating an actual competition did not have to occur. Modeling the interface translated to a competitive edge, since the cognitive model allowed them to pinpoint trouble spots before the competition.

Based on the guide, for this work the Natural Goals, Operators, Methods, and Selections Language (NGOMSL) method was selected. This model builds upon traditional GOMS by allowing a modeler to take learning into account. This learning factor was attractive for the work presented in this paper because the user interface is novel and will require user learning. For the application in this paper, the researchers believed that NGOMSL, with the learning factor, provides the most accurate representation for the use case. The researchers followed these steps and equations outlined in the sources above when constructing the model. Overall, the NGOMSL model allows the authors to evaluate and identify prototype features that are promising interface options without having the costly overhead associated with a full user study.

## References

- [1] Swink, M.L., Sandvig, J.C., Mabert, V.A., 1996, "Customizing concurrent engineering processes: Five case studies," *Journal of Product Innovation Management*, 13(3), pp. 229-244.
- [2] Shen, W., Hao, Q., Li, W., 2008, "Computer Supported Collaborative Design: Retrospective and Perspective," *Computers in Industry*, 59(9), pp. 855-862.
- [3] Belkadi, F., Bonjour, E., Camargo, M., Troussier, N., & Eynard, B., 2013, "A situation model to support awareness in collaborative design," *International Journal of Human-Computer Studies*, 71(1), 110-129.
- [4] Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J., Welch, B., 1992, "Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*, pp. 599-607.
- [5] Bouchlaghem, D., Shang, H., Whyte, J., Ganah, A., 2005, "Visualisation in Architecture, Engineering and Construction (AEC), *Automation in Construction*," 14(3), pp. 287-295.
- [6] Zhong, H., Wachs, J. P., & Nof, S. Y., 2014, "Telerobot-enabled HUB-CI model for collaborative lifecycle management of design and prototyping," *Computers in Industry*.
- [7] Berta, J., 1999, "Integrating VR and CAD," *Computer Graphics and Applications, IEEE* , 9(5), pp.14-19.
- [8] Witmer, B. G., Singer, M. J., 1998, "Measuring presence in virtual environments: A presence questionnaire," *Presence: Teleoperators and virtual environments*, 7(3), 225-240.
- [9] Schuemie, M. J., Van Der Straaten, P., Krijn, M., Van Der Mast, C. A., 2001, "Research on presence in virtual reality: A survey," *CyberPsychology & Behavior*, 4(2), 183-201.
- [10] Mujber, T.S., Szecsi, T., Hashmi, M.S.J., 2004, "Virtual Reality Applications in Manufacturing Process Simulation," *Journal of Materials Processing Technology*, Vol. 155–156, pp. 1834-1838.
- [11] Bowman, D.A., and McMahan, R.P., 2007, "Virtual Reality: How Much Immersion Is Enough?," *Computer* , 40(7), pp.36-43.
- [12] McMahan, R.P., Gorton, D., Gresock, J., McConnell, W., Bowman, D.A., 2006, "Separating the Effects of Level of Immersion and 3D Interaction Techniques," *Proceedings of the ACM symposium on Virtual reality software and technology (VRST '06)*, pp.108-111.
- [13] Barfield, W., Hendrix, C., Bystrom, K., 1997, "Visualizing the Structure of Virtual Objects Using Head Tracked Stereoscopic Displays," *Virtual Reality Annual International Symposium*, pp.114-120.

- [14] Gruchalla, K., 2004, "Immersive Well-path Editing: Investigating the Added Value of Immersion," *Virtual Reality - Proceedings*, pp.157-164.
- [15] Kosmadoudi, Z., Lim, T., Ritchie, J., Louchart, S., Liu, Y., Sung, R., 2013, "Engineering design using game-enhanced CAD: The potential to augment the user experience with game elements," *Computer-Aided Design*, 45(3), pp. 777-795.
- [16] Ju, W., Madsen, S., Fiene, J., Bolas, M., McDowall, I., Faste, R., 2003, "Interaction Devices for Hands-On Desktop Design," *SPIE*, pp. 585-595.
- [17] Francese, R., Passero, I., Tortora, G., 2012, "Wiimote and Kinect: Gestural User Interfaces Add a Natural Third Dimension to HCI," *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*, pp.116-123.
- [18] Shiratuddin, M.F.; Wong, K.W., 2011, "Non-contact Multi-hand Gestures Interaction Techniques for Architectural Design in a Virtual Environment," *Information Technology and Multimedia (ICIM)*, pp.1-6.
- [19] Purschke, F., Schulze, M., Zimmermann, P., 1998, "Virtual Reality-new Methods for Improving and Accelerating the Development Process in Vehicle Styling and Design," *Computer Graphics International*, pp.789-797.
- [20] Fiorentino, M., Uva, A.E., Monno, G., Radkowski, R., 2012, "Augmented Technical Drawings A Novel Technique for Natural Interactive Visualization of Computer-Aided Design Models," *Journal of Computing and Information Science in Engineering*, Vol. 12, pp. 1-8.
- [21] Tumkor, S., Esche, S. K., Chassapis, C., 2013, "Hand Gestures in CAD Systems," *Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition*, pp. 1-9.
- [22] Nanjundaswamy, V.G., Kulkarni, A., Chen, Z., Jaiswal, P., Shankar, S.S., Verma, A., Rai, R., 2013, "Intuitive 3D computer-Aided Design (CAD) System With Multimodal Interfaces," *Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers in Engineering Conference*, pp. 1-11.
- [23] Song, J., Cho, S., Baek, S., Lee, K., Bang, H., 2014, "GaFinC: Gaze and Finger Control Interface for 3D Model Manipulation in CAD Application," *Computer-Aided Design*, Vol. 46, pp. 239-245.
- [24] Wright, M., Lin, C., O'Neill, E., Cosker, D., Johnson, P., 2011, "3D Gesture Recognition: An Evaluation of User and System Performance," *International Conference on Pervasive Computing*, pp. 294-313.
- [25] Lee, S., Chae, J., Kim, H., Lim, Y., Lee, K., 2013, "Towards a More Natural Digital Content Manipulation via User Freehand Gestural Interaction in a Living Room," *Proceedings on the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*, pp. 617-626.
- [26] Park, H., Park, J., Kim, M., 2012, "3D Gesture-Based View Manipulator for Large Scale Entity Model Review," *Asia Simulation Conference*, pp. 524-533.



- [27] Sabir, K., Stolte, C., Tabor, B., O'Donoghue, S., 2013, "The Molecular Control Toolkit: Controlling 3D Molecular Graphics via Gesture and Voice," *Biological Data Visualization*, pp. 49-56.
- [28] Fikkert, W., van der Vet, P., van der Veer, G., Nijholt, A., 2010, "Gesture for Large Display Control," 8<sup>th</sup> International Gesture Workshop, pp. 245-256.
- [29] Gallo, L., Placitelli, A.P., Ciampi, M., 2011, "Controller-Free Exploration of Medical Image Data: Experiencing the Kinect," *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, pp. 27-30.
- [30] Nintendo – Corporate Information, 2014, "Company History", <http://www.nintendo.com/corp/history.jsp>.
- [31] Lowensohn, J., 2011, "Timeline: A Look Back at Kinect's History," CNET.
- [32] Microsoft Developer Network, 2014, "Kinect for Windows Sensor Components and Specifications." <http://msdn.microsoft.com/en-us/library/jj131033.aspx>.
- [33] LEAP Motion, 2014, <https://www.leapmotion.com/>.
- [34] Gorman, M., 2012, "Leap Motion Gesture Control Technology," Engadget, <http://www.engadget.com/2012/05/25/leap-motion-gesture-control-technology-hands-on/>.
- [35] Eisert, J., Gartenberg, D., Thornton, R., Youmans, R., 2012, "Optimal Interface Location and Limits of Gesture Proficiency in an Automobile," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 56(1), pp. 1882 – 1886.
- [36] Gray, W.D., John, B.E., Atwood, M.E., 1993, "Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance," *Human-Computer Interaction*, Vol. 8, pp. 237-309.
- [37] John, B.E., Kieras, D.E., 1996, "Using GOMS for User Interface Design and Evaluation: Which Technique?," *ACM Transactions on Human-Computer Interaction*, 3(4), pp. 287-319.
- [38] Kieras, D., 1996, "A Guide to GOMS Model Usability Evaluation Using NGOMSL."
- [39] Helander, M., Landauer T.K., Prabhu, P.V., 1997, *Handbook of Human-computer Interaction*. Amsterdam: Elsevier, Print.
- [40] Parsons, B., White, A., Warner, P., Gill, R., 2006, "Validation Methods for an Accessible User Interface Rehabilitation Robot," *Universal Access in the Information Society*, 5(3), pp. 306-324.
- [41] Lee, S., Koubeck, R.J., 2011, "Communicative NGOMSL: Development of an Evaluation Method for a Text-Based Communication System," *International Journal of Human-Computer Interaction*, 27(11), pp. 1013-1036.
- [42] Mohan, R.E.; Calderon, C.A.A.; Changjiu Zhou; Pik Kong Yue; Lingyun Hu, 2008, "Modelling human-humanoid robot interaction in soccer robotics domain using

NGOMSL," *Robot and Human Interactive Communication (RO-MAN 2008), The 17th IEEE International Symposium on*, pp. 407-412.

## CHAPTER 3: A NATURAL USER INTERFACE FOR IMMERSIVE DESIGN REVIEW

### Forward

This paper was originally published in the proceedings of the ASME 2014 International Design and Engineering Technical Conferences & Information in Engineering Conference. The paper was titled "*A Natural User Interface for Immersive Design Review.*" The formatting of the paper has been modified to fit with the rest of the thesis. The contents have not been altered.

### Abstract

As markets demand engineered products faster, waiting on the cyclical design processes of the past is not an option. Instead, industry is turning to concurrent design and interdisciplinary teams. When these teams collaborate, engineering CAD tools play a vital role in conceptualizing and validating designs. These tools require significant user investment to master, due to challenging interfaces and an overabundance of features. These challenges often prohibit team members from using these tools for exploring alternatives. This paper presents a method allowing users to interact with a design using intuitive gestures and head tracking, all while keeping the model in a CAD format. Specifically, Siemens' Teamcenter® Lifecycle Visualization Mockup (Mockup) was used to display the design geometry while modifications were made through a set of gestures captured by a Microsoft Kinect™ in real time. This proof of concept program allowed a user to rotate the scene, activate Mockup's immersive menu, move the immersive wand, and manipulate the view based on head position. The result is an immersive user-friendly low cost platform for interdisciplinary design review.

## Introduction

Designing and validating new products like aircraft and machinery rely heavily on engineering tools. While these tools are very effective at creating, modifying, and evaluating designs, they can be very cumbersome to learn for all except practiced experts. CAD packages in general are dominated by a hierarchical approach for creating model representations [1]. This approach is often confusing to inexperienced or casual CAD users.

With the recent emphasis on interdisciplinary design teams [2], those unaccustomed to CAD interfaces are forced to deal with engineering tools to help evaluate designs. The significant learning curve and stigma of difficulty associated with navigating engineering software creates a significant hurdle for reviewing designs thoroughly. In addition, even if non-engineering users wanted to familiarize themselves with CAD technology, they most likely do not have time, resources, or access to these types of software tools. In light of this, a design team often becomes reliant on a subset of its members to drive changes. This limits other group members' ability to explore and review these changes. Limits on the members lead to a sub-optimal functioning team as well as design output.

Due to the lack of user centered design principles in many CAD interfaces, user navigation challenges do not come as a surprise [3, 4]. A way to deal with the deficiencies in CAD interfaces is to incorporate user-centered design principles seen in many VR applications [5 – 7]. In addition, VR aids immersion through head tracking and provides more intuitive interaction modes [8, 9]. This immersion gives the user a more complete mental model of the design, improving understanding and evaluation ability. VR systems though are often expensive and require considerable preparation work to load models and maintain the system, not to

mention the space and setup requirements. Large-scale VR systems are custom solutions tailored to the installations specific physical space and not easily moved or repurposed. The custom nature of each system, the space requirements, and cost make deploying traditional large VR systems challenging. As a result, many companies that invest in VR have limited facilities. These limited resources cannot support numerous dispersed teams that could benefit from more user-friendly evaluation tools.

Even though VR is becoming more affordable and easy to implement, it is still another tool inserted into a crowded well-established workflow. VR programs for design review add additional expensive software/hardware to purchase and maintain to already budget conscientious departments.

In addition, VR does not often use the same file formats as CAD programs. CAD program file types contain boundary representation (b-rep) data. B-rep based files contain information on the topology and exact geometry of the model. Exact dimensions and surface relations allow designers to modify and evaluate models in engineering software packages. Converting from a CAD format to a VR format produces a tessellated model. Tessellated model geometry is specified by a series of vertex points. Therefore the file longer contains the topology information or the exact geometry. Measurements taken from the tessellated format are only as exact as the tessellation granularity. Accuracy of the tessellation is not exact enough for the majority of engineering applications. Also, changes to the model are challenging since all operations must be done on a vertex-by-vertex basis. Converting between file types results in a loss of important engineering data stored in CAD files. Not only does CAD geometry need to be converted into a VR format, but also any changes proposed in the VR package need to be

reimplemented in CAD after a design review, due to the limitations of VR formats. Tools are needed to bridge this gap for design reviews between CAD and VR to allow manipulation of geometry in a user-friendly environment.

Overall, engineering CAD tools play a central role in designing and evaluating products. However, the ease of use of these tools is generally poor for non-everyday users, creating a barrier for truly effective immersive interdisciplinary design review. The work presented in this paper takes steps towards a solution combining the best of both worlds. This research presents a medium for performing design reviews in a program that can display CAD information, eliminating conversion loss and the need for another expensive tool. The work takes steps towards a solution that addresses pitfalls associated with CAD by integrating the VR principles of gesture-based controls and head tracking. These principles allow the user to develop a better mental model of the design, thus helping them understand and identify possible issues early on in the design process.

### Background

Previous research relating to the work in this paper is scattered into three main areas; research on team use of technology, the benefits of VR, and natural user interfaces (NUIs). Work in each of these areas influenced the building of the immersive low cost user-friendly design review prototype presented in this paper.

## Teams and Technology

Design teams today are increasingly made up of interdisciplinary backgrounds. Swink et al. points out in today's world concurrent design is needed to meet customer demands and to keep a company profitable in the fast paced international market [10]. As a result, companies are favoring concurrent design, leading to interdisciplinary teams. These teams need effective tools that allow them to communicate and understand each aspect of the design process [11, 12]; helping them review designs and assemble strategies to pinpoint potential roadblocks.

The necessity for tools of this nature has not escaped researchers. There exist numerous examples of past research looking for ways to help teams collaborate and communicate. For example Liveboard, a device developed to display information on a larger more input friendly screen [13]. Use of the device was targeted at situations where simple desktop interactions fell short. Results point to people viewing these devices as helpful and worthwhile tools. However, the researchers found while people want a device that helps them communicate, if using the device is challenging they are not willing to spend time trouble shooting in a group environment.

Bouchlaghem et al. presents an example of tools helping interdisciplinary teams in Architecture, Engineering and Construction [14]. Their research focused on an integrated visualization tool for project information, such as blue prints, CAD models and design mockups. They found that a common tool helped conceptual designers work collaboratively and communicate ideas. The tool facilitated planning amongst the groups and served as a consulting tool in meetings with customers. In the end, the easy to use tool helped bridge the gap between designers and builders by keeping them on the same page.

Work presented by Zhong et al. details a system that helps collaborators exchange ideas and coordinate their efforts on a project [15]. The use cases involving the system show its value at aiding communication between team members. The system proved helpful at preventing conflicts during the design conceptualization phase all the way through product implementation.

Based on previous work, it is evident researchers recognize the need for tools that help facilitate interactions with design information in a group setting. Research points to the importance of making tools intuitive and accessible for all team members. This involves taking steps towards going beyond regular mouse and keyboard interaction for engineering tools. Identifying and investigating elements of a program that help users escape the limited traditional interaction approach will lower the barrier of entry to effective design review with CAD tools. As the work matures the result will allow all members of a team to work together and contribute during a design review.

### **Benefits of Virtual Reality**

What makes VR attractive to academia and industry is it provides users experiences that are difficult or impossible to replicate in the physical world. This type of interaction can be invaluable to a team of product designers. It allows them to see and experiment with their design in ways that are too dangerous or cost prohibitive in real life. This VR ability mixed with CAD provides great potential. Berta [16] takes an in-depth look at the potential benefits to industry if VR and CAD integration becomes reality. Benefits to integration include: 1) using VR principles to create simpler interfaces for casual users, 2) receiving CAD accuracy along with part information, (e.g., dimensions, kinematics etc.), and 3) the immersion and sense of “presence” that comes with VR. Where immersion is defined as the perception that one is



completely enveloped by or pulled into the virtual environment and presence is the feeling physically being in the virtual environment [17, 18]. However, implementing CAD and VR integration proves difficult. Challenges listed by Berta mainly stem from the transfer between CAD and VR packages. These challenges include loss of geometry, topology, semantics (i.e. object names, dimensions, constraints, etc.), and behaviors.

In addition to the challenges with conversion, there is also the challenge of VR adoption. While VR can be very beneficial to many divisions within a company [19], VR systems are traditionally large and costly installations that serve a specific purpose. These traditional installations provide users with the sense of immersion and presence but at a significant cost. This is a considerable barrier for VR adoption. Even a large company can only afford a limited number of these systems, not enough for the numerous interdisciplinary teams that could benefit. One of the main goals of these large expensive systems is immersion. Bowman and McMahan find that large scale VR provides nearly full immersion, but in some cases full immersion is not always necessary to reap the benefits of VR [20].

McMahan et al. further investigates immersion and the link it has to six degree of freedom (6 DOF) manipulation tasks. In the study they control the level of immersion by turning stereo on or off and using one or three cave walls. The researchers ask the participants to perform a task using one of three interaction devices while head tracked. They find interaction devices have more influence on participant completion time and errors than immersion level [21]. In their discussion they interpret these results as meaning that one can receive the same benefits of large VR systems using less costly displays combined with head tracking and 6 DOF input devices. While the results conclude that full immersion benefits are not tied to large-scale VR

environments, they neglect to pinpoint what degree of immersion is enough. They also do not discuss what components besides stereo and number of screens contributes to increasing user presence in a system.

While a full VR system is not necessary to provide the user with a sense of immersion, certain components of VR do need to be present. Barfield finds that head tracking plays an important role in the presence of users in a system [22]. In the study, participants view a bent wire with some combination of stereo and head tracking on or off and then select the corresponding 2D image on paper. Study results indicate head tracking and stereo does not help the users correctly determine the 2D wire image but it does increase their presence, or feeling of immersion.

Gruchalla provides another look at how VR can help with immersion [23]. In the study, participants plan a new well path in a mature oil field. Participants plan a well using a stereoscopic desktop display or a head tracked stereoscopic CAVE display. Participants using the CAVE found more correct paths and perform faster than those who merely used the stereo display without head tracking. According to the authors the results indicate VR environments can help users develop a more accurate understanding of complex 3D environments.

Background research on VR suggests lower cost systems can provide users with some of the same immersive benefits seen in large-scale systems. As long as users can manipulate and interact with models, they can build a more sound mental model of complex 3D parts. The guidelines of immersion and low cost VR play a central role in the program developed for the work in this paper.

## Natural User Interfaces

As pointed out by McMahan et al. the interaction mode can be just as important as the display hardware. Some of the biggest detractors to CAD program usability stems from interface deficiencies. As natural user interfaces (NUIs) have become commonplace, much research has been done to examine their potential.

Kosmadoudi et al. proposes using games to help users better understand the programs [24]. But Ju et al. takes a step closer to addressing the need for improving interaction between the user and the program [25]. Ju et al. creates physical tools and maps them to the corresponding actions in the software.

Francese et al. published research using low cost tools to help people explore Bing maps [26]. Participants in the study navigate and react to images on screen. Results of the study show that motion control devices help increase the users' sense of immersion and presence in a program. In addition, participants who used the Wiimote and Kinect adapted to an expert level of control, moving around the map faster than other users.

Shratuddin and Wong study limitations of standard interaction techniques. The work does a good job of identifying weaknesses in the current design process [27]. They point out that current GUIs limit creativity and lock users into a linear design process. To combat this, researchers develop a program that allows the user to collaboratively design and manipulate components in a virtual world using intuitive gestures.

Work by Purschke et al. is an example of an engineering application used at Volkswagen to aid in the conceptual design of cars [28]. The program uses gestures and glove input which allows a user to manipulate views and material choices inside the model. They try and solve the

problem of lack of CAD data in VR by using ACIS and integrating it into the VR system for areas that need exact representations.

Fiorentino et al. 2012 takes a more in depth look, using a study to benchmark some of the benefits of NUIs. They highlight the difficulty and complications associated with using CAD interfaces. They set out to create a system that lets the user explore a CAD model through augmented reality and technical drawings. The authors perform a user study validating that NUIs help users become proficient and comfortable with a program. Study participants took only on average 20 minutes with the program before they felt comfortable [29].

Fiorentino et al. 2013 describes a program where users conduct a design review using the STEP files paired with augmented reality information. They use intuitive gestures to let users manipulate the scene and explore models. They found that users were able to get up to speed quickly [30].

So far the background has focused on the building blocks for combining VR and CAD, but no program assembles the pieces and tries to address the road map laid out by Berta [16]. The main takeaway from the background research conducted is that CAD needs to be made more accessible for use in interdisciplinary design reviews. Towards this goal of accessibility NUIs are promising, which is why they are the basis for the work presented. In addition to the accessibility goal, when building a program one needs to make it easy to use, affordable and immersive. This would enable a wide range of disciplinary experts to gain a greater understanding of a current design iteration without feeling pressure in a group setting. Affordability is also important since the program should be available for use by many design teams to see benefits. Lastly, highlighted principles of immersion are required so users can

develop a more complete mental model of the design to make more intelligent decisions. Research presented in this paper uses these requirements as a guide. This work combines and builds upon previous research to create an immersive user-friendly low cost prototype for interdisciplinary design review.

#### Methodology – Program Overview

Work presented in this paper takes steps towards combining the benefits of VR and CAD into a prototype for immersive design reviews. The project uses Siemens' Teamcenter Visualization Mockup to display the models. Control of the package is provided to the user via gestures and tracking.

The project is composed of three major components: the code developed by Siemens in collaboration with the researchers, code developed by the researchers at Iowa State University and the commercial hardware used to capture the users movements. The chain of events leading to a user seeing a change in Mockup starts when the motion hardware, in this case a Kinect, captures a user's movement data. This data is deciphered as either a gesture or joint movement. After the type of data is determined, it is packaged into a command and passed onto Mockup for display through the VisController Application Programming Interface (API). Figure 1 shows the diagram of project components along with information travel.

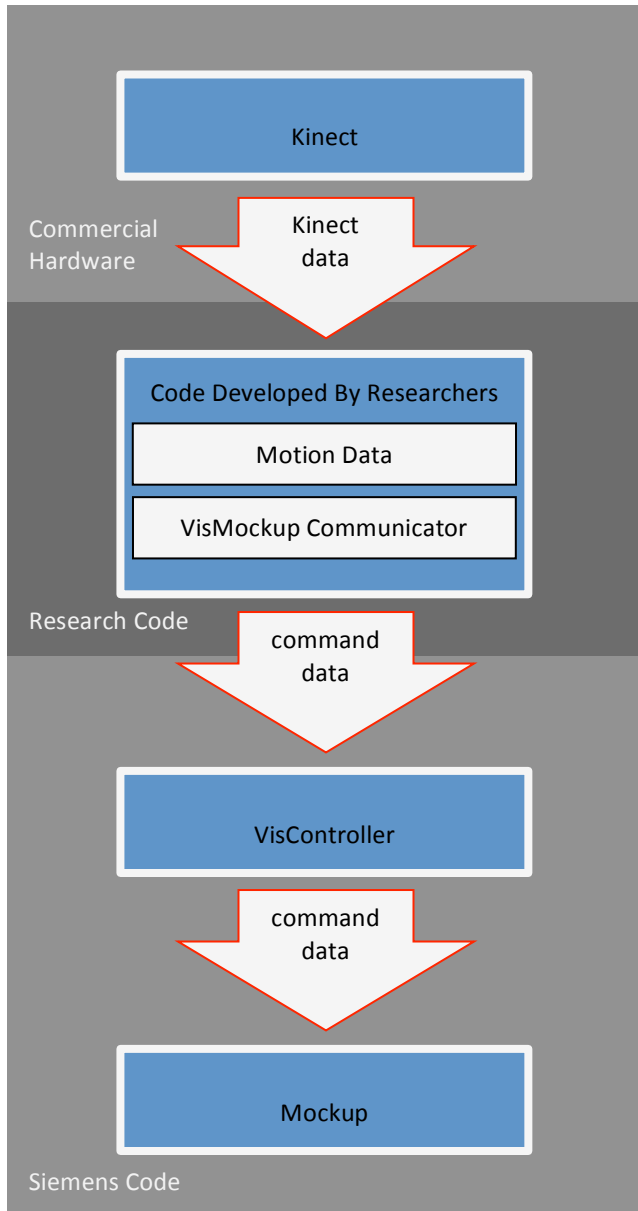


Figure 1: Program Component Diagram on screen model clockwise about the upwards y-axis and right swipe rotates the model counter clockwise about the upwards y-axis. Right push activates and deactivates the immersive virtual wand-tracking mode. When wand-tracking mode

Program functionality revolves around four gestures mapped to Mockup commands and head tracking of the user. The four gestures are left swipe, right swipe, right push and left push. Since the program was an initial proof of concept integrating a number of components as discussed in the background section, only a limited number of gestures were included for testing capability. Diagrams of left hand gesture motions are shown in Figure 2 and Figure 3. The right hand motions are mirror images of those shown below.

Left swipe is mapped to rotate the



Figure 2: Left Click Gesture



Figure 3: Left Swipe Gesture

is engaged, the immersive virtual wand is mapped to the users right hand movements. When the immersive wand-tracking mode is active and the virtual wand comes into contact with part of the model, the piece will turn a solid color. This indicates that the part can be selected or modified. The color changing feedback feature will be useful for

future work in part selection. Left push activates and deactivates the immersive menu, shown in Figure 4. Notice that when the immersive wand comes into contact with one of the menu icons, it turns a solid color indicating that it is available for selection. This feedback allows natural interaction with the Mockup interface elements and user via the immersive virtual wand.

In order to see the benefits associated with past research on NUIs, selected gestures were intended to be intuitive and easy for users to remember. The swipe motions were selected for rotating the model because they parallel the action of rotating a model in the physical world. Clicking motions were selected for triggering the menu and

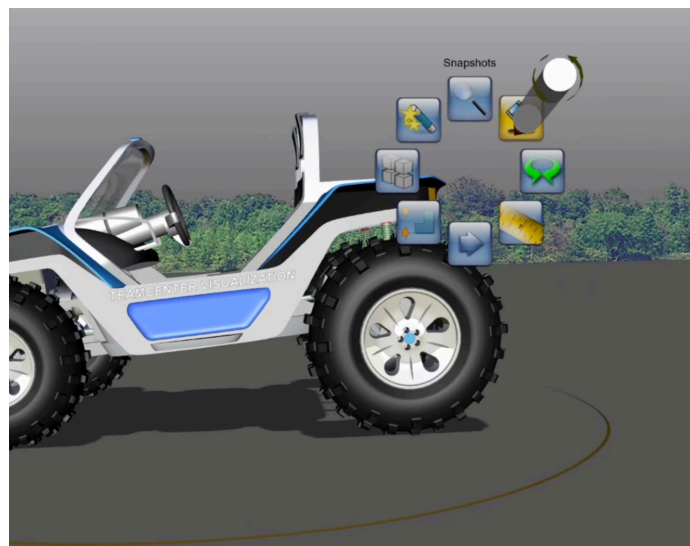


Figure 4: Immersive Wand Interacting With Immersive Menu

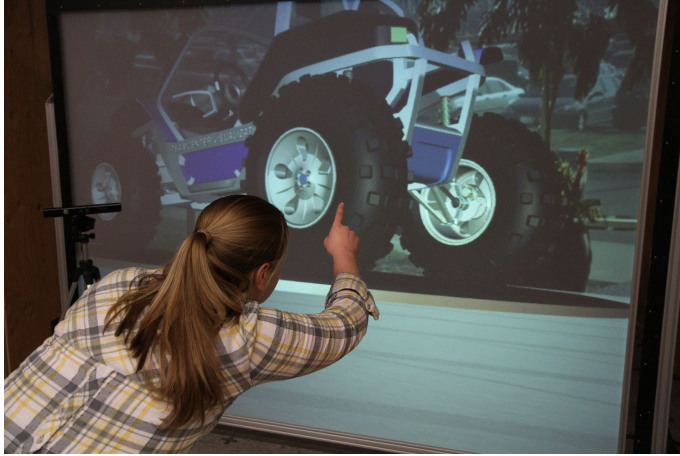


Figure 5: Head Tracking View Adjustment

wand due to the prevalence of clicking for selection, making it easy for users to remember.

Another important piece of the program is head tracking. Background research indicates that head tracking

improves users overall immersion and understanding of the model on the screen [16 - 18]. Head tracking in the program adjusts the view of the model in a 2D plane as the user moves. For example, if the user moves their head down, the view on screen lowers so they can inspect the underside of an aircraft just as one would do in a physical environment. Furthermore, if the user moves right or left, the view adjusts accordingly, as shown in Figure 5.

In order for the onscreen model interactions to scale properly with movement, the user must perform a onetime set up of the Mockup immersive scene before launching the program. Set-up steps ensure the immersive viewing window is centered in a stereo comfort zone. Once the immersive viewing window is set-up, this and the model can be saved in a Mockup VF file and used repeatedly if the model file is unchanged.

## Methodology – Program Components

### Teamcenter Visualization Mockup

Teamcenter Visualization Mockup is the piece of the application that displays all the information to the user. It is an engineering product lifecycle management tool developed by Siemens. It uses its own proprietary JT format to store and display 3D models. Model data can



contain a large amount of information such as product structure, boundary representations and product manufacturing information. The program was selected for the project because of wide industry support for JT files and Siemens products.

A 64 bit version with the 9.1.2 patch was used for project development which included software hooks created by Siemens. These software hooks allow Mockup to receive commands sent to the program through the VisController API.

Mockup communicates with VisController using a TCP socket connection. The TCP connection currently allows one-way communication and ensures commands are received when sent. Future releases of the API will include two-way communication.

Currently the program written by the researchers requires Mockup to be in immersive mode to run the program. Setting up immersive mode requires preparation of the JT model for immersive viewing along with the use of system specific configuration files. The Siemens product specific configuration files required to run the program are VCD, SCD and the XML files. The SCD file sets up the “devices” used during the immersive session. For this project, the “devices” were the head and wand. Once this file is setup, it seldom needs to be changed. The VCD file sets up the immersive viewing window size and stereo viewing properties. A file is required for each display setup to ensure the correct size and stereo is displayed in immersive mode. The XML file initializes the immersive display preferences. For the work presented in this paper, settings for hotkey functions, immersive wand type and initial motion sensitivity were set in this file.

## **Mockup VisController**

VisController (VisController) is an API written by Siemens in collaboration with the researchers to pass “commands” into Mockup using the software hooks mentioned earlier. All accessible actions are specified in this portion.

The goal of VisController was to create an extensible API. The main functionality of this portion of the project was to establish a connection with Siemens’ software, pass commands and send error message feedback to the user. Messages sent to can consist of head position, wand position, mapped hot key commands, navigation mode, select and deselect.

VisController’s complexity and numerous steps to set up a connection prompted creation of a Command Structure (discussed in the next section) to provide an abstract extensible interface for developers; while also insulating code from changes made by Siemens to VisController throughout the development process. All required actions for setting up VisController are handled by VisMockup Communicator code. As a result, developers do not need to interact with VisController directly to manipulate Mockup.

## **Command Structure**

The command structure is extensible code written in C++ that contains variables for all the information VisController can transmit. Command packages are used to pass information about user actions to various parts of the program where they are interpreted. The components of this structure are detailed in Figure 6.

The way commands are used within the program and the command structure set-up make it very extensible. Structure of the commands abstracts them from the functionality of the

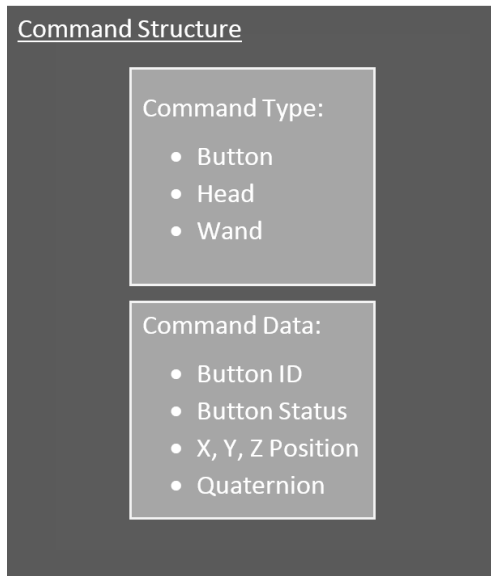


Figure 6: Command Structure

program. Additional features added to VisController require only a handful of functions added in key places. This ensures the program can be reused and extended upon in future work.

### **Motion Data**

The motion data portion of the code runs the Kinect, processes the raw data and handles gestures.

Raw data and gestures from the Kinect are translated into commands available in Mockup.

This section uses a third party API called Omek Beckon [31] to identify gestures and return joint position information from the Kinect. When a user moves or a gesture is detected, the Motion Data section creates a command package using the Command Structure. This package is sent to VisMockup Communicator where the abstract command is then converted into the VisController specific format.

Figure 7 shows the three types of command packages created by the Motion Data portion of the program, along with the data each package contains. In the program, the button commands are mapped to gestures and head/wand movements are from user joint information.

For button commands, when a user performs a gesture it triggers package creation in the Motion Data section. This package contains the ButtonID (mapped to a specific hot key command within Mockup) and button down set to true/false (for actions like show/hide

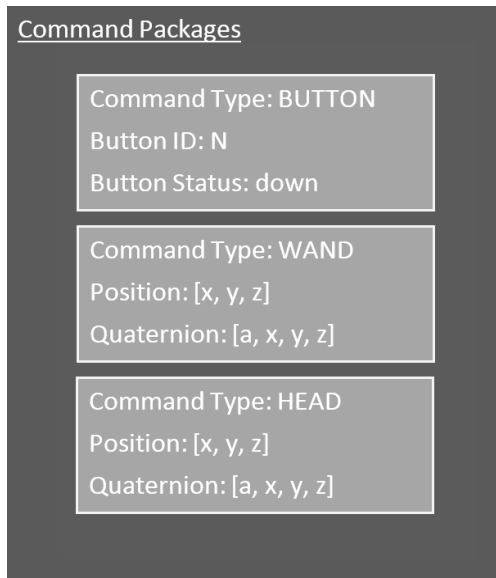


Figure 7: Command Package Types

immersive menu). In contrast to the wand and head packages, where position coordinates and rotation information is packaged into a command based on joint data from the Kinect.

This Motion Data portion of the code is very flexible and can be adapted to use devices other than the Kinect like the LEAP Motion [32] or TrackIR [33]. For example, during program testing, a Qt [34] graphical user interface (GUI) was used to send commands. In

this case instead of using gestures, button presses and numerical scroll boxes were used to trigger events. As long as input data is mapped to a command available in Mockup the program will function with any input or gesture.

The gestures used for the program came from Omek Beckon pre-trained to users movements. Each gesture in the program was given a corresponding action in Mockup. Additional gestures can be added or removed depending on the user's specifications. This requires only one additional step, mapping gestures to functions in the Motion Data portion of the code.

### VisMockup Communicator

The purpose of VisMockup Communicator is to:

- Use VisController to perform setup actions
- Package and unpackage data commands

- Direct command data to VisController

This interface, written in C++, provides a way for developers to quickly begin manipulating an immersive scene while insulating their code from changes to VisController. Overall, the main job of VisMockup Communicator is to abstract interaction between VisController and the rest of the code. VisMockup Communicator ensures data in each command package is sent to the correct function within VisController, to forward on to Mockup.

Vismockup Communicator itself is flexible and easy to modify. Minimal coupling exists between Vismockup Communicator methods and those in VisController. If a change is made in VisController or new functionality is added, VisMockup Communicator requires changes in only a few key locations.

Making a connection requires Mockup to be running. If it is not running and VisMockup Communicator tries to make a connection, it will sit and wait, checking periodically to see if a connection can be made. Once a connection is established VisMockup Communicator sends an initialize message to VisController ensuring that it is ready to receive commands. As soon as VisController is ready to accept commands, they can be passed via VisMockup Communicator.

VisMockup Communicator interprets data via the command structure. It must unpack each command package and pass command data to the correct VisController functions. However, once VisController receives command data from VisMockup Communicator it can be sent directly to Mockup without further modifications.

## **Kinect**

The first generation Kinect hardware tracks users' basic movements and provides that data to Omek Beckon. The Kinect was not modified in any way for the project. It is the standard, commercial hardware that can be purchased by the public.

## **Conclusion**

The resulting project provides a step towards an immersive, user-friendly, low cost platform for interdisciplinary design review. Users of the program receive the best of both VR and CAD. Program benefits include increased understanding, enhanced mental models of complex 3D spaces, and inclusion of CAD engineering/manufacturing data. In addition, user interaction is performed with an easy to use natural user interface (NUI) eliminating the drawbacks related to traditional CAD packages, lowering the barrier of entry for use in design reviews with interdisciplinary teams. This ensures all members of a team can explore and interact with possible designs, increasing the possibility of identifying potential design flaws.

While the platform described above uses Siemens' Teamcenter Lifecycle Visualization Mockup capabilities to display models, the framework developed is extendable beyond the package. Adding support for another program would require minor tweaks to the VisMockup Communicator structure in order to pass the command data in the correct format. Moving forward, the hope is that more companies recognize the value VR principles can add to their design pipeline. Once companies realize this, they can then start pushing CAD software developers to create hooks into their programs like discussed in this work.

### Future work

Future project work will focus on adding features to the program and conducting user studies to refine user interaction and measure benefits to understanding.

Additional features will focus on expanding the capabilities of the program. Any additional features will provide the user more control over the system and allow more manipulation of the environment on screen. Functions that would greatly enhance the user's experience are: selecting parts, the ability to move parts individually, zoom and triggering animations. Selection will be especially challenging due to the limited resolution of the Kinect. Moving forward, selection and part manipulation will be aided by the addition of two-way communication from Mockup. The next version of VisController will be able to send the position and rotation information for each part, in addition to the part selected. This will open the door to more interaction between the user and individual parts, increasing the platforms ability to add value during a design review.

After expanding the number of features, next is a user study to help refine the gesture selection. User feed back will ensure gestures are intuitive and easy to use. After gesture refinement comes a study focusing on users interacting and interpreting designs using the tool. Such a study would focus on time required before users feel comfortable with the program and the extent the tool helps users unfamiliar with a design build a mental representation.

### References

- [1] Stacey, M., and Claudia E, 1999, "CAD System Bias in Engineering Design," Proceedings of the 12th International Conference on Engineering Design, Vol. 2.
- [2] Wang, L., Shen, W., Xie, H., Neelamkavil, J., Pardasani, A., 2002, "Collaborative Conceptual Design—State of the Art and Future Trends", *Computer-Aided Design*, 34(13), pp. 981-996.

- [3] Hartson, H.R., 1998, "Human-Computer Interaction: Interdisciplinary Roots and Trends," *Journal of Systems and Software*, 43(2), pp. 103-118.
- [4] Shackel, B., 1997, "Human-Computer Interaction—Whence and whither?," *Journal of the American Society for Information Science and Technology*, Vol. 48, pp. 970-986.
- [5] van Beurden, M. H., IJsselsteijn, W. A., & Hopf, K., 2011, "User centered design of gesture-based interaction technology," In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, (pp. 1-4). IEEE.
- [6] Fidopiastis, C. M., Rizzo, A. A., & Rolland, J. P., 2010, "User-centered virtual environment design for virtual rehabilitation," *Journal of Neuroengineering and Rehabilitation*, 7(11).
- [7] Hix, D., Swan, J. E., Gabbard, J. L., McGee, M., Durbin, J., & King, T., 1999, "User-centered design and evaluation of a real-time battlefield visualization virtual environment," In *Virtual Reality, 1999. Proceedings*, (pp. 96-103). IEEE.
- [8] Jaimes, A., Sebe, N., 2007, "Multimodal human-computer interaction: A survey," *Computer Vision and Image Understanding*, Vol. 108, No. 1-2, pp. 116-134.
- [9] Pavlovic, V.I., Sharma, R., Huang, T.S., 1997, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: a Review," *Pattern Analysis and Machine Intelligence*, 19(7), pp. 677-695.
- [10] Swink, M.L., Sandvig, J.C., Mabert, V.A., 1996, "Customizing concurrent engineering processes: Five case studies," *Journal of Product Innovation Management*, 13(3), pp. 229-244.
- [11] Shen, W., Hao, Q., Li, W., 2008, "Computer Supported Collaborative Design: Retrospective and Perspective," *Computers in Industry*, 59(9), pp. 855-862.
- [12] Belkadi, F., Bonjour, E., Camargo, M., Troussier, N., & Eynard, B., 2013, "A situation model to support awareness in collaborative design," *International Journal of Human-Computer Studies*, 71(1), 110-129.
- [13] Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J., Welch, B., 1992, "Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*, pp. 599-607.
- [14] Bouchlaghem, D., Shang, H., Whyte, J., Ganah, A., 2005, "Visualisation in Architecture, Engineering and Construction (AEC), *Automation in Construction*," 14(3), pp. 287-295.
- [15] Zhong, H., Wachs, J. P., & Nof, S. Y., 2014, "Telerobot-enabled HUB-CI model for collaborative lifecycle management of design and prototyping," *Computers in Industry*.
- [16] Berta, J., 1999, "Integrating VR and CAD," *Computer Graphics and Applications*, IEEE , 9(5), pp.14-19.
- [17] Witmer, B. G., Singer, M. J., 1998, "Measuring presence in virtual environments: A presence questionnaire," *Presence: Teleoperators and virtual environments*, 7(3), 225-240.



- [18] Schuemie, M. J., Van Der Straaten, P., Krijn, M., Van Der Mast, C. A., 2001, "Research on presence in virtual reality: A survey," *CyberPsychology & Behavior*, 4(2), 183-201.
- [19] Mujber, T.S., Szecsi, T., Hashmi, M.S.J., 2004, "Virtual Reality Applications in Manufacturing Process Simulation," *Journal of Materials Processing Technology*, Vol. 155–156, pp. 1834-1838.
- [20] Bowman, D.A., and McMahan, R.P., 2007, "Virtual Reality: How Much Immersion Is Enough?," *Computer*, 40(7), pp.36-43.
- [21] McMahan, R.P., Gorton, D., Gresock, J., McConnell, W., Bowman, D.A., 2006, "Separating the Effects of Level of Immersion and 3D Interaction Techniques," *Proceedings of the ACM symposium on Virtual reality software and technology (VRST '06)*, pp.108-111.
- [22] Barfield, W., Hendrix, C., Bystrom, K., 1997, "Visualizing the Structure of Virtual Objects Using Head Tracked Stereoscopic Displays," *Virtual Reality Annual International Symposium*, pp.114-120.
- [23] Gruchalla, K., 2004, "Immersive Well-path Editing: Investigating the Added Value of Immersion," *Virtual Reality - Proceedings*, pp.157-164.
- [24] Kosmadoudi, Z., Lim, T., Ritchie, J., Louchart, S., Liu, Y., Sung, R., 2013, "Engineering design using game-enhanced CAD: The potential to augment the user experience with game elements," *Computer-Aided Design*, 45(3), pp. 777-795.
- [25] Ju, W., Madsen, S., Fiene, J., Bolas, M., McDowall, I., Faste, R., 2003, "Interaction Devices for Hands-On Desktop Design," *SPIE*, pp. 585-595.
- [26] Francese, R., Passero, I., Tortora, G., 2012, "Wiimote and Kinect: Gestural User Interfaces Add a Natural Third Dimension to HCI," *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*, pp.116-123.
- [27] Shiratuddin, M.F.; Wong, K.W., 2011, "Non-contact Multi-hand Gestures Interaction Techniques for Architectural Design in a Virtual Environment," *Information Technology and Multimedia (ICIM)*, pp.1-6.
- [28] Purschke, F., Schulze, M., Zimmermann, P., 1998, "Virtual Reality-new Methods for Improving and Accelerating the Development Process in Vehicle Styling and Design," *Computer Graphics International*, pp.789-797.
- [29] Fiorentino, M., Uva, A.E., Monno, G., Radkowski, R., 2012, "Augmented Technical Drawings A Novel Technique for Natural Interactive Visualization of Computer-Aided Design Models," *Journal of Computing and Information Science in Engineering*, Vol. 12, pp. 1-8.
- [30] Fiorentino, M., Radkowski, R., Stritzke, C., Uva, A.E., Monno, G., 2013, "Design Review of CAD Assemblies Using Bimanual Natural Interface," *International Journal on Interactive Design and Manufacturing*, 7(4), pp. 249-260.

- [31] Omek Interactive, Ltd., 2014, Beckon,  
<http://www.omekinteractive.com/products/beckon-3-for-pc/>.
- [32] Leap Motion, Inc., 2014, <https://www.leapmotion.com>.
- [33] NaturalPoint, Inc., 2014, TrackIR, <http://www.naturalpoint.com/trackir/>.
- [34] Qt, 2014, <http://qt-project.org>.

## CHAPTER 4: ASSESSING THE FEASIBILITY OF MANIPULATING ENGINEERING GEOMETRY USING A MICROSOFT KINECT AND COMPUTER-AIDED ENGINEERING SOFTWARE

A paper submitted to *The Journal of Computing and Information Science in Engineering*

Anastacia MacAllister<sup>1,2</sup>, Dr. Eliot Winer<sup>3</sup>, Tsung-Pin Yeh<sup>4</sup>, Daniel Seal<sup>5</sup>, and Grant Degenhardt<sup>6</sup>

### Abstract

This paper evaluates gesture usability and task completion time for a system that enables users to manipulate engineering geometry in a computer-aided design / computer-aided engineering (CAD/CAE) environment. A cognitive model evaluation method was used to evaluate the premise that gesture-based user interfaces are easier to use and learn with regards to time than a traditional mouse and keyboard interface. Using a cognitive model analysis tool allowed the rapid testing of interaction concepts without the significant overhead of user studies and full development cycles.

---

<sup>1</sup> Graduate Student, Department of Mechanical Engineering, Iowa State University

<sup>2</sup> Primary researcher and author

<sup>3</sup> Associate Professor, Department of Mechanical Engineering, Iowa State University

<sup>4</sup> Software Engineer – Principal, Siemens PLM Software

<sup>5</sup> Program Manager, The Boeing Company

<sup>6</sup> Senior Engineer, The Boeing Company

## Introduction

Designing and validating new products like aircraft and machinery rely heavily on engineering tools. Tools like computer aided design (CAD) and computer aided engineering (CAE) often drive the design process and are integral in product development [1]. While these tools are very effective at creating, modifying, and evaluating designs, they can be very cumbersome to use [2]. Low cost commodity interaction devices like the Kinect™ and Leap Motion™ could help improve CAD and CAE tools by introducing user centered interface design principles seen traditionally in virtual reality [3 – 8].

Combining low cost interaction devices with engineering tools has the possibility to simplify interfaces, improve user interaction, increase understanding of designs, and allow increased participation in the design process [9, 10]. However, incorporating low cost interaction devices into CAD and CAE does introduce fundamental changes to the user experience. Change of this magnitude requires improvements in small stages, with continuous evaluation along the way. Towards this end, creating a proof of concept prototype systems allows identifying, testing, and refining key interface elements for engineering software that go beyond the mouse and keyboard. Evaluation throughout the process in stages saves time and increases the likelihood of a system that is ready for full user testing.

The commodity interaction tools market is extremely fluid with new devices rapidly entering the market. As a result, potential devices need to be evaluated quickly. Timely evaluation is required to match pace with the new ideas continually introduced. Evaluation of devices will aid in generating requirements for creating a system that shifts to a new interaction paradigm for CAD and CAE, beyond the mouse and keyboard.

Along with traditional mouse and keyboard interfaces there is also the traditional mode of evaluation. Evaluation is commonly performed on fully functioning systems. Building and evaluating a fully functioning system, however, requires significant resources to program, test, recruit users, and analyze. By the time any testing or analysis is done changes can become expensive and time consuming. In addition, with commodity devices, any lengthy evaluation becomes obsolete. By the time results are prepared, a new model, often with many new features, is available. The goal of quick evaluation necessitates building and testing programs in a short amount of time. To accomplish this the authors advocate building proof of concept prototype systems with a few key features and evaluating them using existing usability methods. For concept evaluation the authors focused on cognitive modeling methods. Cognitive modeling provides the opportunity to quickly and cheaply evaluate concepts, ensuring the ability to evaluate a wider range of options than with traditional methods [11].

Overall, engineering CAD tools play a central role in designing and evaluating products. However, the ease of use of these tools is generally poor and limited to dated mouse and keyboard interaction. The work presented in this paper evaluates in terms of usability a proof of concept solution that integrates low cost interaction devices and CAD/CAE using cognitive modeling tools using a fast and agile method.

### Background

Previous research relating to the work in this paper is distributed into six main areas: 1) research on team use of technology, 2) the benefits of VR, 3) natural user interfaces (NUIs), 4) gestures for 3D manipulation, 5) low cost interaction devices and 6) Goals, Operators, Methods,

and Selection cognitive modeling. Work in each of these areas influenced the building and evaluation of the immersive low cost user-friendly prototype presented in this paper.

### **Teams and Technology**

Design teams today are increasingly made up of interdisciplinary backgrounds. Swink et al. points out that in today's world concurrent design is needed to meet customer demands and to keep a company profitable [12]. As a result, companies are favoring concurrent design, leading to interdisciplinary teams. These teams need effective tools to communicate and understand each aspect of the design process [13, 14], helping them review designs and assemble strategies to pinpoint potential roadblocks.

The necessity for tools of this nature has not escaped researchers. There exist numerous examples of research looking for ways to help teams collaborate and communicate [15 – 17]. Previous research points to the importance of making tools intuitive and accessible. This involves moving beyond mouse and keyboard interaction. Identifying and investigating elements of a program that help users escape the limited traditional interaction approach will lower the barrier of entry associated with CAD and CAE tools. As more research is performed the resultant outcomes will allow recommendations to be made regarding the meshing of low cost tools and engineering design programs.

### **Benefits of Virtual Reality**

What makes VR attractive to academia and industry is the features and experiences provided that are difficult or impossible to replicate in the physical world. Examples include, exploring how a new aircraft design will fit on an existing carrier before building a physical

prototype or practicing a dangerous assembly process before ever stepping in a work cell. These types of interactions can be invaluable to a team of product designers. It allows them to experiment with their design in ways that are time consuming, cost prohibitive or impossible to enact in real life. These abilities, mixed with CAD and CAE, provide enormous potential. Berta [18] takes an in-depth look at the potential benefits to industry if VR and CAD / CAE integration becomes reality. These include: 1) using VR principles to create simpler interfaces, 2) receiving CAD accuracy along with part information, and 3) the immersion and sense of “presence” associated with VR. Immersion is defined as the perception that one is completely enveloped by or pulled into the virtual environment and presence is the feeling of physically being in the virtual environment [19, 20]. However, implementing CAD / CAE and VR integration proves difficult. Challenges listed by Berta mainly stem from the transfer between CAD and VR packages.

In addition to the challenges with conversion, there is also the challenge of VR adoption. While VR can be very beneficial to many divisions within a company [21], these systems are traditionally large and costly installations. This is a considerable barrier for VR adoption. Even a large company can only afford a limited number of these systems, not enough for the numerous interdisciplinary teams that could benefit. One of the main goals of these large expensive systems is immersion. Bowman and McMahan find that large scale VR provides nearly full immersion, but in some cases full immersion is not always necessary to reap the benefits of VR [22]. In other work McMahan et al. found interaction devices have more influence on participant completion time and errors than immersion level [23]. In their discussion they interpret these results as meaning that one can receive the same benefits of

large VR systems using less costly displays combined with head tracking and 6 DOF input devices.

While a full VR system is not necessary to provide the user with a sense of immersion, certain components of VR do need to be present. Barfield finds that head tracking plays an important role in the presence of users in a system [24].

Gruchalla provides another look at how VR can help with immersion [25]. In the study, participants planned a new well path in a mature oil field using either a stereoscopic desktop display or a head tracked stereoscopic cave automatic virtual environment (CAVE™) display.

Background research on VR suggests lower cost systems can provide users with some of the same immersive benefits seen in large-scale systems. As long as users can manipulate and interact with models, they can build a more sound mental model of complex 3D parts. The guidelines of immersion and low cost VR play a central role in the method development of the proof of concept program presented in this paper.

### **Natural User Interfaces**

As pointed out by McMahan et al. the interaction mode can be just as important as the display hardware. Some of the biggest detractors to CAD / CAE program usability stems from interface deficiencies. As natural user interfaces (NUIs) have become commonplace, much research has been done to examine their potential [26 – 32].

Tumkor et al. recognized the ability of low cost interaction devices to improve interaction and collaboration in design [33]. From a user study they found that performing some tasks though the SolidWorks™ interface were faster using gestures, but for the majority of tasks a mouse was still the most efficient. The researchers concluded that hand-gesture



recognition systems hold promise for interacting with CAD systems, however changes are necessary to free the user from the traditional menu navigation in order to reap the benefits of gestures.

Song et al. developed a system for interacting with CAD models using gaze and finger control [34]. The system aimed to tackle one of the drawbacks to gesture systems, user fatigue. When building and testing their system they identified three primary CAD tasks translation, rotation, and zooming. Results from the interviews suggested that users found the gestures much more intuitive for carrying out the tasks compared to the mouse and keyboard. However, in terms of comfort users on average reported that they found the mouse and keyboard interaction to be more comfortable.

This research illustrates how CAD and CAE programs would benefit from NUIs incorporating low cost, commodity interaction technologies. Towards this goal of accessibility NUIs are promising, which is why they are the basis for the proof of concept interface presented in this paper.

### **Natural User Interfaces for 3D Manipulation**

With the introduction of devices like the Kinect™ and the increase in computing power, the hardware for gesture recognition has become commoditized. The potential of this technology has not gone unnoticed by the research community [35 – 37]. Unfortunately though, researchers who create these systems tend to focus on high recognition rates and do not consider gesture usability [38]. For the work presented, the authors drew on previous publications in 3D gesture interaction to guide the development of a gesture based user

interface (UI). Based on the literature review, gestures were assigned to actions accordingly, taking into account the needs of the user in terms of understanding and usability.

Lee et al. looked at developing gestures to control content displayed on a television screen [39]. They used Wizard-of-Oz studies to develop gesture sets and allow user evaluation of gestures. Wizard-of-Oz studies prompt participants to perform a certain action, in this case a gesture, to complete an assigned task. Wizard-of-Oz studies for the work by Lee et al. allowed them to focus on the user interaction component and not be limited by the technology available. Though their studies they found that eliciting user input in this manner provided helpful insight for the creation of more intuitive gestures.

Fikkert et al. researched gestures for use with large displays [40]. Their study found that users consider tapping or clicking to trigger a feature intuitive. Results also indicated that to keep interfaces simple, duplicate gestures for activating and deactivating a feature is recommended.

The literature review above on interaction modes for this paper focused on free hand 3D object manipulation work. This ensured that gestures used in the researcher's program were applicable to the action assigned to the gesture and intuitive. Work surveyed reinforced assigning gestures to specific functions in the prototype program presented. Elements incorporated in this work from previous research aims to ensure the gestures match what users expect interaction wise from a program for specific functions.

### **Low Cost Interaction Devices**

Low cost devices reviewed were the Kinect™, Wimote™, and the Leap Motion™. The Wiimote™ released with the Nintendo Wii™ gaming system in 2006 [41] is considered a pioneer in the motion gaming market. While pioneering when first released the technology is now somewhat dated and limited.

The Microsoft Kinect™ was released in 2010 as a controller less gaming device and heralded as the future of interaction [42]. The low cost and ease of development with the device, \$150, made it ideal for research. While the device is proficient at collecting large sweeping motions, its limited resolution prevents capturing finer hand motions.

The third device considered for the research presented in this paper was the Leap Motion™ [43]. The device was first released to the public in 2013 and retails for \$79.99. While the device provides precision tracking this increased accuracy comes with tradeoff in tracking area.

The Microsoft Kinect™ was selected as the low cost immersion device for the proof of concept system after surveying the available devices. The researchers determined that the Kinect provided the greatest flexibility when trying to design and test a system that branches out from the traditional mouse and keyboard desktop interaction.

### **Cognitive Modeling Using the GOMS Family**

There are a number of cognitive modeling methods developed by academics and industry alike. One such family of modeling methods adopted by those who work in the engineering field is Goals, Operators, Methods, and Selections (GOMS). The GOMS family of methods is well suited to the type of sequential and hierarchal interfaces found in engineering

applications, like CAD / CAE. The GOMS family of models consists of a number of variants. Each variant is geared towards a specific use case.

The GOMS method is popular in engineering circles for evaluating user interfaces. This family of models is fairly extensible. Existing literature uses the model for validating and testing interfaces from text to gestures. Research by Eisert et al. [44] used the GOMS-KLM method to predict task completion time for different gesture interfaces in an automobile. Based on the results of the modeling process a hypothesis was developed about where users would prefer the gesture interface. The researchers found that the user study results matched the hypothesis generated by the cognitive model validating the use of GOMS before user testing. This approach allowed designers to test interfaces with minimum time and user resources. They also helped identify deficiencies before users became involved. The work by Gray et al. compared interfaces for telephone company toll and assistance operators [45] as they were investigating claims by operators about a slow and frustrating new interface. After conducting a GOMS analysis they found that the proposed new interface was actually much slower than the original, potentially costing millions in lost productivity.

While GOMS analysis is a powerful tool, selecting the proper variation is imperative to receive meaningful results. John and Kieras compiled a guide to selecting a GOMS type model based on a modeler's specific needs [46]. This guide aided the work presented in this paper by highlighting the strengths of each model and selection aspects to consider. Traditional GOMS is limited because it only models expert users who are already familiar with the interface, whereas NGOMSL takes into account learning time for novice users. Kieras and the Handbook

of Human Computer Interaction detail the steps required when performing a NGOMSL analysis [47, 58].

Parsons et al. uses a variation of NGOMSL to predict task completion time for a gesture user interface between an assistive robot and those with severe spinal cord injuries [49]. From their results, the authors concluded that task analysis models like NGOMSL were accurate predictors of user interface performance when applied in the correct context. Lee and Koubek [50] extended the NGOMSL model to use between multiple users. This work showed the extensibility of the NGOMSL language and its usability beyond standard GUI interfaces.

Mohan et al. uses NGOMSL to model a user interface for soccer playing robots [51]. The researchers took advantage of the learning rate factor in NGOMSL to estimate task completion time for repairing robots.

Based on the guide, for this work the Natural Goals, Operators, Methods, and Selections Language (NGOMSL) method was selected. This model builds upon traditional GOMS by allowing a modeler to take learning into account. This learning factor was attractive for the work presented in this paper because the user interface is novel and will require user learning. Overall, the NGOMSL model allows the authors to evaluate and identify prototype features that are promising interface options without having the costly overhead associated with a full user study.

#### Methodology – Program Overview

Work presented in this paper takes steps towards evaluating, in terms of usability, a prototype that combines low cost interaction devices and CAD/CAE software. Presented below an overview of the prototype system and its components. The project uses Siemens'

Teamcenter Visualization Mockup™ to display the models and Omek Beckon™ to capture the gestures.

Omeck Beckon is an API built off the OpenNI [52] project for capturing information from depth cameras. The application programming interface (API) provides the ability to capture user joint position information along with a standard set of gestures it recognizes. This program was selected based on its support for a number of depth camera devices and its integration with the Kinect™. Also the existence of a limited gesture library with the possibility of expansion attracted the authors.

Teamcenter Visualization Mockup™ [53] is a product lifecycle management tool from Siemens. The program allows users to collaborate during the design process and review conceptual designs. Mockup provides designers the ability to review designs using immersive viewing with navigation, model manipulation, and pointing and annotation tools. The program was selected for the research presented in this paper because of wide industry support and the vast number of features incorporated from both CAD and CAE. The variety of features provided a rich test bed of options for the researchers to explore. A screen shot of the standard Mockup interface is shown below in Figure 8. The screen shot below is an example of what the standard Mockup interface looks like. The standard interface is filled with a wide variety of menu options and is very customizable. The immersive side of the application is more streamlined and contains fewer features. These features are accessed via the immersive menu show in Figure 12.

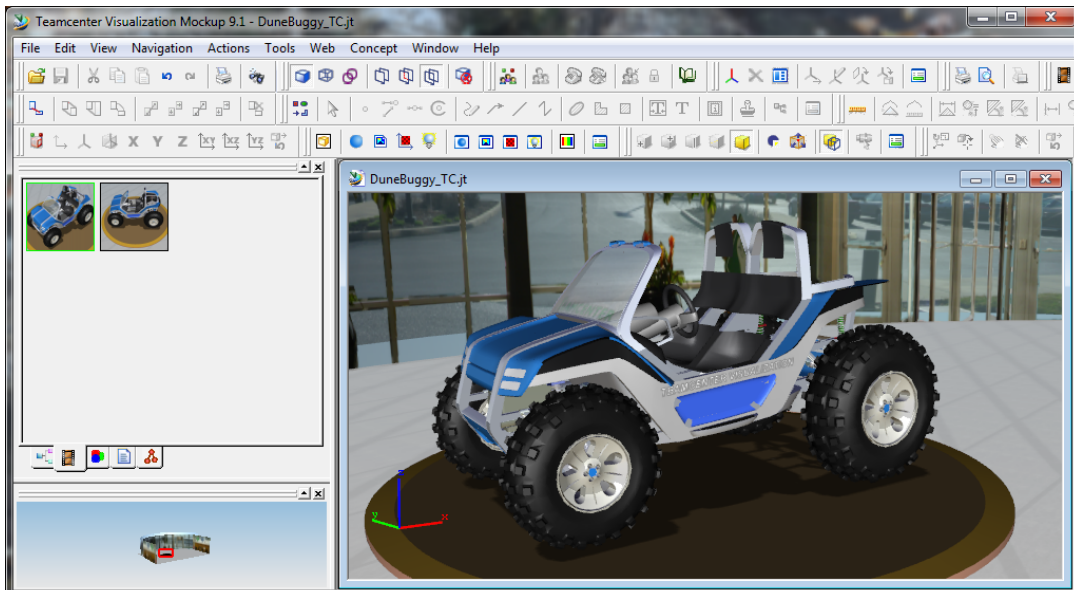


Figure 8: Mockup Interface

The system developed for testing is composed of three major components: the code developed by Siemens in collaboration with the researchers, code developed by the researchers at Iowa State University and the commercial hardware used to capture the user's movements. The chain of events leading to a user seeing a change in Mockup starts when the motion hardware, in this case a Kinect<sup>TM</sup>, captures a user's movement. This data is deciphered as either a gesture or joint movement. After the type of data is determined, it is packaged into a command and passed onto Mockup for display through the VisController Application Programming Interface (API). Figure 9 shows the diagram of project components along with information travel.

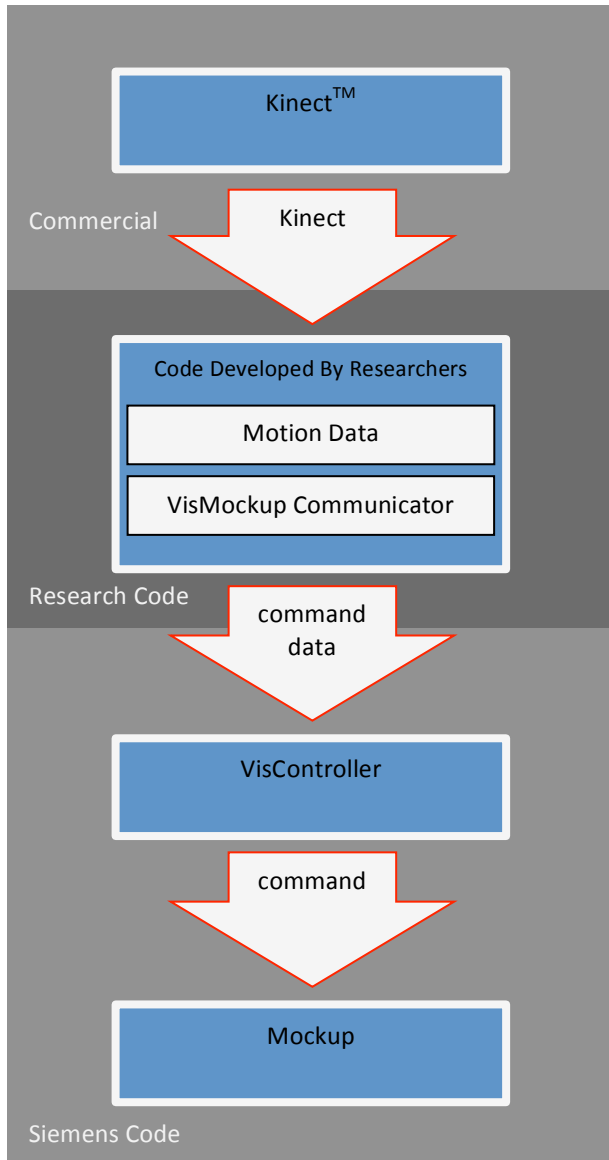


Figure 9: Program Information Flow Diagram modes are going to be a viable alternative to standard interfaces then work needs to focus on usability. The overarching goal when selecting the gestures for the project was to design around user's needs. The gestures available were from Omek Beckon's

Program functionality revolves around four gestures mapped to Mockup commands and head tracking of the user. The four gestures are left swipe, right swipe, right push and left push. Since the program was an initial proof of concept, integrating a number of components as discussed in the background section, only a limited number of gestures were included for testing. Diagrams of left hand gesture motions are shown in Figure 10 and Figure 11. The right hand motions are mirror images of those shown below.

If these new types of interaction

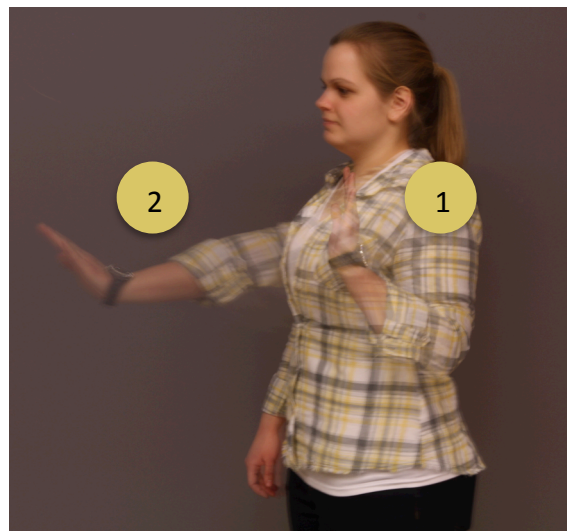


Figure 10: Left Click Gesture



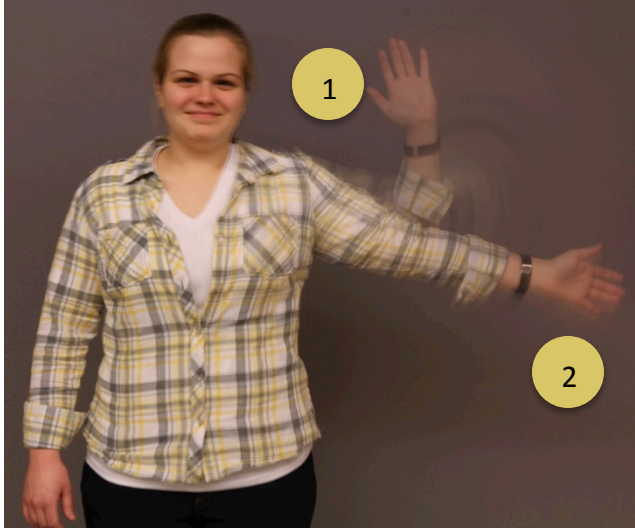


Figure 11: Left Swipe Gesture

pre-trained gesture library. The researchers decided to stick with the already available gestures and focus on assigning them to intuitive commands.

The first step in the NUI interface process for the prototype system was the selection of the features in Mockup to map to gestures. For the prototype the

researchers selected functions that were representative of working in a CAD/CAE environment like in previous research [23]. The functions selected from Teamcenter Visualization Mockup™ were rotate model clockwise (CW), rotate model counter clockwise (CCW), activate/deactivate the immersive menu, and activate/deactivate wand-tracking mode.

The Mockup functions rotate model CW and CCW were assigned the swipe left and swipe right gestures, respectively. These were chosen since previous research suggested that swiping is intuitive and easy for a user to understand [29, 39, 54].

Right swipe and left swipe actions were assigned as mirror images of one another to simplify the interface. Left swipe is mapped to rotate the on screen model clockwise and right swipe rotates the model counter clockwise. Previous research has users perform rotation as one gesture, rotating the model with respect to the range of motion of the user [27]. However, their study results suggested this creates confusion. In addition, this requires the entrance and exit from a rotation mode, which for ease of use the authors wanted to avoid. Previous work indicated that a continuous rotation mode that is activated/deactivated by a singular gesture

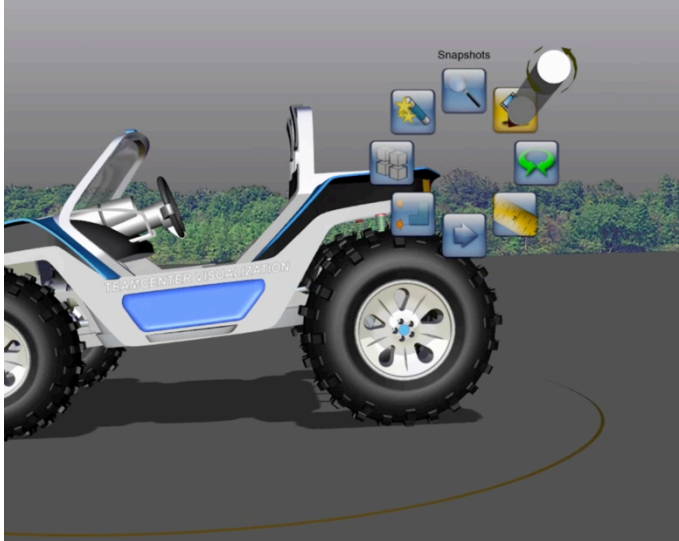


Figure 12: Immersive Wand Interacting With Immersive Menu

can confuse users [27]. To avoid this problem seen in previous research for the prototype developed, gestures for rotation were broken into two distinct motions.

The third function activate/deactivate wand-tracking mode was assigned to the gestures right click.

Right click activates and deactivates the

immersive virtual wand-tracking mode. Previous research indicated that using duplicate gestures for activation and deactivation helped users more easily recall commands [39]. When wand-tracking mode is engaged, the immersive virtual wand is mapped to the user's right hand movements. When the immersive wand-tracking mode is active and the virtual wand comes into contact with part of the model, the piece will turn a solid color. This indicates that the part can be selected or modified. Research shows that clicking for activating/deactivating is a very intuitive function [40, 55]. The wand tracking mode gesture was assigned to the right hand because a selection task, like those done using the wand, requires a certain degree of accuracy. Considering on average most of the population is right hand dominant, the majority of users would feel most comfortable using their dominant hand for finer selection tasks. Also activating/deactivating the wand-tracking mode was assigned the same gesture because: 1) it simplified the interface and 2) was intuitive to most users [40].

The fourth function show/hide immersive menu was assigned the left click gesture. Left push activates and deactivates the immersive menu, shown in Figure 12. When the Mockup immersive wand comes into contact with one of the menu icons, it turns a solid color indicating that it is available for selection. This feedback allows natural interaction with the Mockup interface elements and user via the immersive virtual wand. Again this was justified by the research that shows users find clicking for activation/deactivation to be intuitive [40, 55]. The immersive menu functionality was assigned to the left hand since it does not require a high degree of accuracy.

Further, the researchers took care to make sure that all gestures were short quick motions to reduce the chance of user fatigue, an issue noted in previous work [23]. Gestures required to run the program are short one to five-second bursts. These short gestures avoid the static poses that can tire users [39, 55].

### **Head Tracking**

Another important piece of the prototype system developed was head tracking. Background research indicated that head tracking improved overall immersion and understanding of the model on the screen for a user [18 – 20]. Head tracking in the prototype system adjusts the view of the model in a 2D plane as the user moves. For example, if the user moves their head down, the view on screen raises so they can inspect the underside of an aircraft just as one would do in a physical environment as shown in Figure 13.

Head tracking was implemented utilizing the change in users head position based on the skeleton information provided by the Omek Beckon Kinect™ API. To accomplish this the image size from the Kinect™ was recorded. Then the coordinate system between the Kinect™ image and the VisController API was rectified for the y-scale using an experimentally determined constant. For the x-scale adjustment the value passed to Mockup was calculated as simply the ratio of the users head position to the width of the Kinect™ image. The y-scale image required transforming the coordinate system of the Kinect™ image and user joint position into the same right handed Mockup coordinate system. After this transformation the ratio of the users y head position to the image size is multiplied by an experimentally determined constant.

In order for the onscreen model interactions to scale properly with movement, the user must perform a onetime set up of the Mockup immersive scene before launching the program. Set-up steps ensure the immersive viewing window is centered in a stereo comfort zone. Once the immersive viewing window is set-up, this and the model can be saved in a Mockup VF file and used repeatedly if the model file is unchanged.

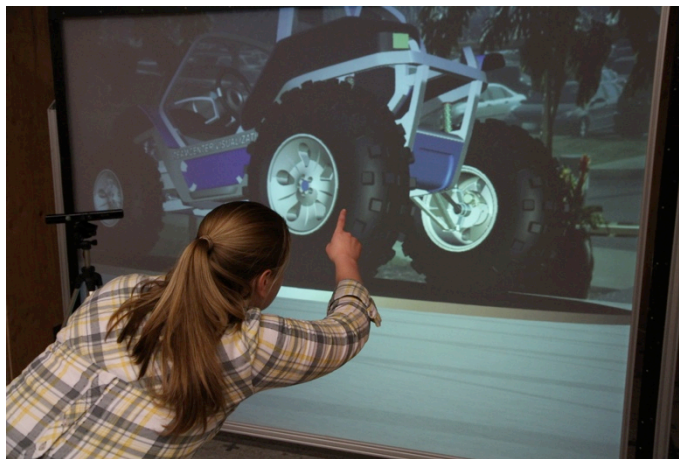


Figure 13: Head Tracking View Adjustment

### Methodology – Interface Evaluation

For the analysis the researchers decided to model three interfaces in Mockup: immersive, standard interface, and the gesture user interface developed. These three were selected to provide an overall sample of the interfaces that users encounter during a design review or typical CAD / CAE work. The program developed above is a proof of concept system that is not ready for full user testing. The researchers wanted to use NGOMSL analysis to test the premise that the gesture user interface is easier to learn and use for typical CAD/CAE engineering activities. Using a cognitive model like NGOMSL provides insight into whether the gesture interface moves in the correct direction when compared to the current Mockup user interface. A cognitive model like NGOMSL allows for testing of the interface without the time and expense of a full user study. In addition, the flexibility of NGOMSL lets the interface be evaluated at any time during the design process. This flexibility is advantageous for proof of concept prototype systems like the one presented, it ensures a wide range of options can be explored and evaluated for potential.

The first step required for conducting the NGOMSL model is a task analysis. This analysis serves to break down the steps a user must conduct to successfully complete an objective or method. The methodology and notation for the task analysis followed the guidelines in the Handbook of Human-Computer Interaction [48]. Through conducting a task analysis the researchers were able to break down into actionable steps the requirements to complete actions like rotate a model and activate menus. Splitting up the actions into the most basic actionable steps allowed the researchers to examine and assign primitive operators, a key component for the NGOMSL analysis. The primitive operators represent the most basic actions,

like clicking a mouse button or pointing to an onscreen target, required to complete an NGOMSL task or goal. A standard primitive operator is necessary to estimate each step's duration and compute the overall execution/learning time. Analysis for the work used primitive operators such as pointing (P), key press (BB), time required to place hands on mouse or keyboard (H) and mental preparation time (M) based on the GOMS family KLM method developed by [56]. Under the KLM method each statement is assigned a time required for completion.

After determining and assigning a primitive operator, the steps for each method were evaluated to determine if they were a statement that the user would need to "learn". NGOMSL classifies a learned statement as one that is specific to the program that the user would not have seen before. The combination of learned statements and the type of training, strenuous or not, provides an estimate of the pure method learning time. The pure method learning time estimates the time that a user needs to learn and become familiar with the task steps. For the learning time the decision was made to only model pure method learning time. Long-term memory was not included in the estimation because the researchers only desired to model users who are new to the program. This is an assumption made because not all users would be familiar with a CAD/CAE software environment.

The execution time is the next NGOMSL component. This provides an estimate of the time to complete the sequential tasks listed in the task analysis breakdown for each feature in the prototype system and Mockup. Figure 14 shows the task analysis breakdown for activating the Mockup immersive menu in each of the three interface types (Mockup immersive, Mockup GUI, prototype system). The same task analysis breakdown and assignment of primitive

operators was conducted for each NUI feature in the proof of concept prototype system. In the task analysis below, the letter after each step indicates the standard primitive operator assigned based on the guidelines outlined in the KLM method. Values are assigned to the primitive operators, except for the user defined gesture operator G, using the recommended KLM execution times [56]. The user defined gesture operator G represents the time required to complete the gesture for the step. This time varies with the gesture performed. For the work presented the time is based on values in Omek Beckon documentation [57].

*Method for Goal: Activate Immersive Menu (Mockup GUI – via Concept)*

- Step 1. Locate "Concept" menu (M)*
- Step 2. Move cursor to "Concept" menu (P)*
- Step 3. Click "Concept" menu (BB)*
- Step 4. Locate "Immersive Display" submenu (M)*
- Step 5. Move cursor to "Immersive Display" submenu (P)*
- Step 6. Locate "Immersive Menu" option (M)*
- Step 7. Click "Immersive Menu" (BB)*
- Step 8. Return with goal accomplished*

*Method for Goal: Activate Immersive Menu (Immersive Mockup Menu – via Concept)*

- Step 1. Locate "Concept" menu (M)*
- Step 2. Move cursor to "Concept" menu (P)*
- Step 3. Click "Concept" menu (BB)*
- Step 4. Locate "Immersive Display" submenu (M)*
- Step 5. Move cursor to "Immersive Display" submenu (P)*
- Step 6. Locate "Immersive Menu" (M)*
- Step 7. Click "Immersive Menu" (BB)*
- Step 8. Return with goal accomplished*

*Method for Goal: Activate Immersive Menu (Gesture UI)*

- Step 1. Prepare to execute "left click" gesture (M)*
- Step 2. Move to start position (H)*
- Step 3. Execute "left click" gesture (G)*
- Step 4. Return with goal accomplished*

Figure 14: NGOMSL Task Analysis

Notice that for the immersive Mockup and Mockup GUI the steps are identical. This is the case for many of the methods these two share. This can be attributed to the fact that many of the ways to activate certain features are available in both immersive and standard desktop GUI mode. For completeness, all steps were modeled for the task analysis in each interface.

After conducting the task analysis and applying the NGOMSL time estimating methodology, the time to complete each goal for each interface method was compared. The gesture functionality was compared against different methods for accomplishing the same goals using Mockup methods.

For the analysis, keyboard shortcuts were ignored because by default Mockup does not have assigned shortcuts. These shortcuts are user defined, thus they are not consistent between computer workstations. This assumption translates into no learning between methods. In addition, each goal analysis method predicts the task completion time for a user that is unfamiliar with the Mockup interface.

#### Results – Interface Evaluation

The section contains the resulting time estimations from the NGOMSL model for each method evaluated. Table 1 above holds the projected times for activating the Mockup immersive menu via different interfaces and menu options. For Mockup there are multiple timing rows due to the variety of options available to accomplish a certain task. The time estimations were modeled using the NGOMSL notation and formulas found in the Hand-Book of Computer Interaction [48].

Table 1: Activate Immersive Menu Timings by Method

<b>Activate Immersive Menu</b>	Learning Time [sec]	Execution Time [sec]	Learning + Execution [sec]
Mockup Immersive [Via Immersive Toolbar]	51	2.95	53.95
Mockup GUI [Via Immersive Toolbar]	51	2.95	53.95
Gesture UI [Via Left Click Gesture]	85	5.25	90.25
Mockup GUI [Via Navigation Menu]	85	7.15	92.15
Mockup Immersive [Via Navigation Menu]	102	7.25	109.25



Results from activate immersive menu timings indicate the Gesture UI is located in the middle of the set time wise. It is markedly faster to learn and execute the left click gesture than using the standard Mockup interface file menu system shown above in Figure 8. The gesture interface is an improvement over the Mockup menu system time wise. Notice though if using the standard Mockup interface with the immersive tool bar visible less time is required to activate the immersive menu than the gesture interface. However, the immersive tool bar is not turned on by default and may be unavailable for the user to interact with. The times for deactivating the immersive menu for all three interface options Mockup immersive, Mockup standard, and gesture were identical. This is a result of the gestures and the Mockup interface having identical steps to toggle off the immersive 3D menu.

The timings indicate that the gesture interface for activating the immersive menu is not as fast as some of the standard interface options. Using the toolbar buttons in Mockup outperformed the gesture UI in both learning and execution time. The menu bar is easy to locate when activated supporting this result. However, while Mockup gives the user the ability to configure menu options and show/hide toolbars all CAD/CAE packages do not. From the table above one can see for actions where customization using toolbars is not possible the gesture interface shows potential for both being faster to learn and execute.

Table 2: Rotate Model CW Timings By Method

<b>Rotate Model CW</b>	Learning Time [sec]	Execution Time [sec]	Learning + Execution [sec]
Gesture UI [Via Left Swipe Gesture]	85	4.25	89.25
Mockup GUI [Via 3D Navigation Toolbar]	85	4.65	89.65
Mockup Immersive [Via 3D Navigation Toolbar]	85	4.65	89.65
Mockup GUI [Via Navigation Menu]	85	7.4	92.40
Mockup Immersive [Via Navigation Menu]	102	7.4	109.40
Mockup GUI [Via Immersive Menu]	170	9.35	179.35
Mockup Immersive [Via Immersive Menu]	170	9.35	179.35

Table 2 contains the modeled results, by method, representing the time required to learn and execute rotating the model counter clockwise. Results indicate that for a more complicated task such as rotating a model, the gesture UI takes considerably less time to learn and execute than other methods. This can be attributed to the limited number of steps in the gesture UI when compared with the navigation menu. Lengthy times in Table 2 for activating rotation mode using the immersive menu are a result of the additional time required to activate the immersive menu. Activating the immersive menu is a necessary sub method in the overall rotation goal. Time required to navigate the immersive menu is compounded with the time required to turn on the immersive menu. For the analysis, the immersive menu activation time used was the fastest immersive menu tool bar case found in Table 1.

The reduction in learning time demonstrated by the gesture interface shows its potential to improve CAD/CAE interaction. The large reduction in time for a more complicated action like rotating the model is inline with previous research since the gesture interface introduces a new interaction paradigm. Freeing the user from traditional CAD/CAE interfaces and seeing a reduction in completion time when using novel interaction modes like gestures for

complicated actions was predicted in previous work by Tumkor et al. [21]. This shows the potential of gesture based UIs when freed from the constraints of CAD/CAE type menu interaction. When interface designers really are truly free to take a look at interaction from the ground up and are not constrained by previous ideas or conventions improvement is possible.

The results from the NGOMSL model for counter clockwise rotating methods mirror those seen in the clockwise model. This can be attributed to the mirrored nature of the gestures, keeping the interface simple and intuitive for users.

Table 3: Activate Wand Tracking Projected NGOMSL Task Times by Method

<b>Activate Wand Tracking</b>	Learning Time [sec]	Execution Time [sec]	Learning + Execution [sec]
Mockup Immersive	51	2.5	53.50
Gesture UI	85	5.25	90.25
Mockup GUI	NA	NA	NA

Table 3 contains the model results for activating wand tracking. The Mockup GUI could not be compared because the standard user interface does not support the immersive wand. For this type of action the gesture interface learning and execution time is almost double the standard immersive interface. Based on the immersive Mockup set up this is expected. Immersive Mockup can be hooked up to a tracking system to control the wand and is always on. The user does not have to use any movements or button presses to activate wand mode. To activate wand mode they simply have to move the wand to the point of interest. This always on state gives the immersive wand an edge over the gesture-controlled wand in a time based comparison model. However, the intuitiveness of the hand gesture method coupled with the overall benefits of the gesture interface maybe enough to outweigh the time. Again the

deactivate wand results, because of the mirrored nature of the programs, are the same as in the previous table.

Table 4: Task Completion Time Comparison

Goal	Mockup Immersive		Mockup GUI		Gesture UI	
	Learning Time [sec]	Execution Time [sec]	Learning Time [sec]	Execution Time [sec]	Learning Time [sec]	Execution Time [sec]
Rotate model CCW	170	9.35	170	9.35	85	4.25
Activate immersive menu	102	7.25	85	7.15	85	5.25
Activate wand tracking	51	2.5	NA	NA	85	5.25
<b>Total</b>	<b>323</b>	<b>19.1</b>	<b>255</b>	<b>16.5</b>	<b>255</b>	<b>14.75</b>

Table 4 contains an example goal execution pattern that one might expect a user to perform. This is shown to further evaluate the feasibility of the gesture UI. The goals in the table above correspond to a use case where a presenter wants to bring into focus on screen an area of interest by rotating the model. The immersive menu is then activated to gain access to evaluation functionality. Finally the wand-tracking mode is activated to interact with the immersive menu or the model.

Timings from the table show the gesture interface being markedly faster than the immersive, in both learning and execution time. Where learning time is the one time cost for becoming familiar with the system and execution time is the time required to complete a task for an expert user. Also the gesture interface matches or improves upon the Mockup GUI interface even though the GUI contains one less step. The time savings demonstrated in the table is for only one instance of the action and may seem negligible on a micro scale. However,

with designers and users performing the same actions numerous times throughout the day time savings add up quickly, resulting in minutes to hours saved each day for common engineering visualization tasks. This savings can translate to increased worker productivity and more efficient enjoyable interaction with CAD/CAE programs.

Overall the results from the NGOMSL analysis point to potential of a gesture interface built off low cost immersion devices to create a new interaction mode to help users interact with CAD/CAE in a more intuitive and effective manner. While there is potential the results also indicate that in every situation gestures are not the most efficient solution. When a traditional GUI interface contains easily accessible and configurable menu bars, a gesture interface may not reduce completion time for simple tasks. However, that assumes that users will take the time to activate and customize these toolbars. Many users work with software's default settings. But for more complicated tasks or where customizable menus are not an option, gestures show promise. In addition to potential time savings, gestures tend to be more intuitive interaction wise for users [1, 5, 10, 29, 58].

Based on the results of the analysis it is evident that in some cases gesture interfaces show potential to help break free from the traditional mouse and keyboard interaction seen in CAD/CAE programs. The time savings shown when using gesture interfaces for complex tasks demonstrates their potential as a new interaction medium for CAD/CAE programs. The NGOMSL analysis conducted validates the proof of concept system on a conceptual level. The development and analysis of the system was conducted over a period of weeks instead of the months or longer required to develop a functioning system suitable for a user study. This fast flexible development and evaluation along with the promising results shows the value of this

type of agile analysis for emerging interaction paradigms. In addition, while the results for this prototype were positive there are many other devices that could be evaluated in the same manner.

### Conclusions

The resulting project provides a step towards an immersive, user-friendly, low cost platform for expanding the interaction options with CAD/CAE programs through the integration of VR principles and low cost technology. The work in this paper provides evidence that a natural user interface can stand up to CAD/CAE based interfaces. When gesture interfaces are compared with menu-based programs like Teamcenter Visualization Mockup™, the advantage to having a simplified interface for common features is evident. A natural intuitive interface can help users navigate a program in a quick, efficient and intuitive way.

The work demonstrates the value of flexible NGOMSL analysis when evaluating proof of concept systems for feasibility. The analysis demonstrated that a NUI using the Kinect™ is a feasible interaction mode for CAD/CAE programs. In addition to the strengths of the NUI, the analysis pointed out limitations such as the NUI's inability to compete time wise with easily accessible customizable menu options. This type of development and analysis cycle has the potential to guide concept selection and reduce the potential of squandering resources on a fundamentally flawed prototypes.

## References

- [1] Wang, L., Shen, W., Xie, H., Neelamkavil, J., Pardasani, A., 2002, "Collaborative Conceptual Design—State of the Art and Future Trends", *Computer-Aided Design*, 34(13), pp. 981-996.
- [2] Stacey, M., and Claudia E, 1999, "CAD System Bias in Engineering Design," *Proceedings of the 12th International Conference on Engineering Design*, Vol. 2.
- [3] Yeo, H., Lee, B., Lim, H., 2013, "Hand Tracking and Gesture Recognition System for Human-Computer Interaction Using Low-Cost Hardware," *Multimedia Tools and Applications*, pp. 1-29.
- [4] Hartson, H.R., 1998, "Human-Computer Interaction: Interdisciplinary Roots and Trends," *Journal of Systems and Software*, 43(2), pp. 103-118.
- [5] Shackel, B., 1997, "Human-Computer Interaction—Whence and whither?," *Journal of the American Society for Information Science and Technology*, Vol. 48, pp. 970-986.
- [6] van Beurden, M. H., IJsselsteijn, W. A., & Hopf, K., 2011, "User centered design of gesture-based interaction technology," In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, (pp. 1-4). IEEE.
- [7] Fidopiastis, C. M., Rizzo, A. A., & Rolland, J. P., 2010, "User-centered virtual environment design for virtual rehabilitation," *Journal of Neuroengineering and Rehabilitation*, 7(11).
- [8] Hix, D., Swan, J. E., Gabbard, J. L., McGee, M., Durbin, J., & King, T., 1999, "User-centered design and evaluation of a real-time battlefield visualization virtual environment," In *Virtual Reality, 1999. Proceedings*, (pp. 96-103). IEEE.
- [9] Jaimes, A., Sebe, N., 2007, "Multimodal human-computer interaction: A survey, *Computer Vision and Image Understanding*," Vol. 108, No. 1-2, pp. 116-134.
- [10] Pavlovic, V.I., Sharma, R., Huang, T.S., 1997, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: a Review," *Pattern Analysis and Machine Intelligence*, 19(7), pp. 677-695.
- [11] Li, X., Gunal, M., 2012, "Exploring Cognitive Modeling in Engineering Usability Design," *Journal of Engineering Design*, 23(2), pp. 77-97.
- [12] Swink, M.L., Sandvig, J.C., Mabert, V.A., 1996, "Customizing concurrent engineering processes: Five case studies," *Journal of Product Innovation Management*, 13(3), pp. 229-244.
- [13] Shen, W., Hao, Q., Li, W., 2008, "Computer Supported Collaborative Design: Retrospective and Perspective," *Computers in Industry*, 59(9), pp. 855-862.
- [14] Belkadi, F., Bonjour, E., Camargo, M., Troussier, N., & Eynard, B., 2013, "A situation model to support awareness in collaborative design," *International Journal of Human-Computer Studies*, 71(1), 110-129.

- [15] Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J., Welch, B., 1992, "Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92), pp. 599-607.
- [16] Bouchlaghem, D., Shang, H., Whyte, J., Ganah, A., 2005, "Visualisation in Architecture, Engineering and Construction (AEC), Automation in Construction," 14(3), pp. 287-295.
- [17] Zhong, H., Wachs, J. P., & Nof, S. Y., 2014, "Telerobot-enabled HUB-CI model for collaborative lifecycle management of design and prototyping," *Computers in Industry*.
- [18] Berta, J., 1999, "Integrating VR and CAD," *Computer Graphics and Applications, IEEE* , 9(5), pp.14-19.
- [19] Witmer, B. G., Singer, M. J., 1998, "Measuring presence in virtual environments: A presence questionnaire," *Presence: Teleoperators and virtual environments*, 7(3), 225-240.
- [20] Schuemie, M. J., Van Der Straaten, P., Krijn, M., Van Der Mast, C. A., 2001, "Research on presence in virtual reality: A survey," *CyberPsychology & Behavior*, 4(2), 183-201.
- [21] Mujber, T.S., Szecsi, T., Hashmi, M.S.J., 2004, "Virtual Reality Applications in Manufacturing Process Simulation," *Journal of Materials Processing Technology*, Vol. 155-156, pp. 1834-1838.
- [22] Bowman, D.A., and McMahan, R.P., 2007, "Virtual Reality: How Much Immersion Is Enough?," *Computer* , 40(7), pp.36-43.
- [23] McMahan, R.P., Gorton, D., Gresock, J., McConnell, W., Bowman, D.A., 2006, "Separating the Effects of Level of Immersion and 3D Interaction Techniques," Proceedings of the ACM symposium on Virtual reality software and technology (VRST '06), pp.108-111.
- [24] Barfield, W., Hendrix, C., Bystrom, K., 1997, "Visualizing the Structure of Virtual Objects Using Head Tracked Stereoscopic Displays," *Virtual Reality Annual International Symposium*, pp.114-120.
- [25] Gruchalla, K., 2004, "Immersive Well-path Editing: Investigating the Added Value of Immersion," *Virtual Reality - Proceedings*, pp.157-164.
- [26] Kosmadoudi, Z., Lim, T., Ritchie, J., Louchart, S., Liu, Y., Sung, R., 2013, "Engineering design using game-enhanced CAD: The potential to augment the user experience with game elements," *Computer-Aided Design*, 45(3), pp. 777-795.
- [27] Ju, W., Madsen, S., Fiene, J., Bolas, M., McDowall, I., Faste, R., 2003, "Interaction Devices for Hands-On Desktop Design," *SPIE*, pp. 585-595.
- [28] Francese, R., Passero, I., Tortora, G., 2012, "Wiimote and Kinect: Gestural User Interfaces Add a Natural Third Dimension to HCI," Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12), pp.116-123.



- [29] Shiratuddin, M.F.; Wong, K.W., 2011, "Non-contact Multi-hand Gestures Interaction Techniques for Architectural Design in a Virtual Environment," *Information Technology and Multimedia (ICIM)*, pp.1-6.
- [30] Purschke, F., Schulze, M., Zimmermann, P., 1998, "Virtual Reality-new Methods for Improving and Accelerating the Development Process in Vehicle Styling and Design," *Computer Graphics International*, pp.789-797.
- [31] Fiorentino, M., Uva, A.E., Monno, G., Radkowski, R., 2012, "Augmented Technical Drawings A Novel Technique for Natural Interactive Visualization of Computer-Aided Design Models," *Journal of Computing and Information Science in Engineering*, Vol. 12, pp. 1-8.
- [32] Nanjundaswamy, V.G., Kulkarni, A., Chen, Z., Jaiswal, P., Shankar, S.S., Verma, A., Rai, R., 2013, "Intuitive 3D computer-Aided Design (CAD) System With Multimodal Interfaces," *Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers in Engineering Conference*, pp. 1-11.
- [33] Tumkor, S., Esche, S. K., Chassapis, C., 2013, "Hand Gestures in CAD Systems," *Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition*, pp. 1-9.
- [34] Song, J., Cho, S., Baek, S., Lee, K., Bang, H., 2014, "GaFinC: Gaze and Finger Control Interface for 3D Model Manipulation in CAD Application," *Computer-Aided Design*, Vol. 46, pp. 239-245.
- [35] Park, H., Park, J., Kim, M., 2012, "3D Gesture-Based View Manipulator for Large Scale Entity Model Review," *Asia Simulation Conference*, pp. 524-533.
- [36] Sabir, K., Stolte, C., Tabor, B., O'Donoghue, S., 2013, "The Molecular Control Toolkit: Controlling 3D Molecular Graphics via Gesture and Voice," *Biological Data Visualization*, pp. 49-56.
- [37] Gallo, L., Placitelli, A.P., Ciampi, M., 2011, "Controller-Free Exploration of Medical Image Data: Experiencing the Kinect," *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, pp. 27-30.
- [38] Wright, M., Lin, C., O'Neill, E., Cosker, D., Johnson, P., 2011, "3D Gesture Recognition: An Evaluation of User and System Performance," *International Conference on Pervasive Computing*, pp. 294-313.
- [39] Lee, S., Chae, J., Kim, H., Lim, Y., Lee, K., 2013, "Towards a More Natural Digital Content Manipulation via User Freehand Gestural Interaction in a Living Room," *Proceedings on the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*, pp. 617-626.
- [40] Fikkert, W., van der Vet, P., van der Veer, G., Nijholt, A., 2010, "Gesture for Large Display Control," *8<sup>th</sup> International Gesture Workshop*, pp. 245-256.
- [41] Nintendo – Corporate Information, 2014, "Company History", <http://www.nintendo.com/corp/history.jsp>.

- [42] Lowensohn, J., 2011, "Timeline: A Look Back at Kinect's History," CNET.
- [43] LEAP Motion, 2014, <https://www.leapmotion.com/>.
- [44] Eisert, J., Gartenberg, D., Thornton, R., Youmans, R., 2012, "Optimal Interface Location and Limits of Gesture Proficiency in an Automobile," Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 56(1), pp. 1882 – 1886.
- [45] Gray, W.D., John, B.E., Atwood, M.E., 1993, "Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance," Human-Computer Interaction, Vol. 8, pp. 237-309.
- [46] John, B.E., Kieras, D.E., 1996, Using GOMS for User Interface Design and Evaluation: Which Technique?," ACM Transactions on Human-Computer Interaction, 3(4), pp. 287-319.
- [47] Kieras, D., 1996, "A Guide to GOMS Model Usability Evaluation Using NGOMSL."
- [48] Helander, M., Landauer T.K., Prabhu, P.V., 1997, *Handbook of Human-computer Interaction*. Amsterdam: Elsevier, Print.
- [49] Parsons, B., White, A., Warner, P., Gill, R., 2006, "Validation Methods for an Accessible User Interface Rehabilitation Robot," Universal Access in the Information Society, 5(3), pp. 306-324.
- [50] Lee, S., Koubeck, R.J., 2011, "Communicative NGOMSL: Development of an Evaluation Method for a Text-Based Communication System," International Journal of Human-Computer Interaction, 27(11), pp. 1013-1036.
- [51] Mohan, R.E.; Calderon, C.A.A.; Changjiu Zhou; Pik Kong Yue; Lingyun Hu, 2008, "Modelling human-humanoid robot interaction in soccer robotics domain using NGOMSL," *Robot and Human Interactive Communication (RO-MAN 2008), The 17th IEEE International Symposium on*, pp. 407-412.
- [52] GitHub, OpenNI, 2013, <https://github.com/OpenNI/OpenNI>.
- [53] Siemens, Siemens PLM Software,  
[http://www.plm.automation.siemens.com/en\\_us/products/teamcenter/](http://www.plm.automation.siemens.com/en_us/products/teamcenter/).
- [54] Rodriguez-Esquivel, J.C., Viveros, A.M., Perry, N., 2014, "Gestures for Interaction Between the Software CATIA and the Human via Microsoft Kinect," HCII 2014 Posters, pp. 457-462.
- [55] Farhadi-Niaki, F., Etemad, S.A., Arya, A., 2013, "Design and Usability Analysis of Gesture-Based Control for Common Desktop Tasks," HCI International, pp. 215-224.
- [56] Kieras, D., 2001, "Using the Keystroke-Level Model to Estimate Execution Times."
- [57] Omek Interactive, Ltd., 2014, Beckon,  
<http://www.omekinteractive.com/products/beckon-3-for-pc/>.

- [58] Carrino, S., Caon, M., Khaled, A., Ingold, R., Mugellini, E., 2013, "Functional Gestures for Human-Environment Interaction," International Conference on Human Computer Interaction, pp. 167-176.

## CHAPTER 5: GENERAL CONCLUSIONS

The resulting project provides a step towards an immersive, user-friendly, low cost platform for expanding the interaction options with CAD/CAE programs through the integration of VR principles and low cost technology. The work in this paper provides evidence that a natural user interface can stand up to CAD/CAE based interfaces. When gesture interfaces are compared with menu-based programs like Teamcenter Visualization Mockup, the advantage to having a simplified interface for common features is evident. A natural intuitive interface can help users navigate a program in a quick, efficient and intuitive way.

The work also demonstrated the value of flexible NGOMSL analysis when evaluating proof of concept systems for feasibility. The analysis demonstrated that a NUI using the Kinect™ is a feasible interaction mode for CAD/CAE programs. In addition to the strengths of the NUI, the analysis pointed out limitations such as the NUI's inability to compete time wise with easily accessible customizable menu options. This type of development and analysis cycle has the potential to guide concept selection and reduce the potential of squandering resources on fundamentally flawed prototypes.

## CHAPTER 6: FUTURE WORK

Future project work will focus on adding features to the program and conducting user studies to refine user interaction and measure benefits to understanding. Additional features will focus on expanding the capabilities of the program. Any additional features will provide the user more control over the system and allow more manipulation of the environment on screen. When adding additional features care will be taken to address the potential for user fatigue. While not addressed in the work presented, prevention of fatigue will be an important consideration when developing a fully functioning program. Functions that would greatly enhance the user's experience are: selecting parts, the ability to move parts individually, zoom and triggering animations. Selection will be especially challenging due to the limited resolution of the Kinect™. Moving forward, selection and part manipulation will be aided by the addition of two-way communication from Mockup. The next version of VisController will be able to send the position and rotation information for each part, in addition to the part selected. This will open the door to more interaction between the user and individual parts, increasing the platforms ability to add value during a design review.

After expanding the number of features, next is a user study to help refine the gesture selection. User feed back will ensure gestures are intuitive and easy to use. After gesture refinement comes a study focusing on users interacting and interpreting designs using the tool. Such a study would focus on time required before users feel comfortable with the program and the extent the tool helps users unfamiliar with a design build a mental representation.