
Abstract—For decades, architectural designs are created and communicated by de facto tools; either 2D CAD or 3D CAD. Architects and CAD operators alike use CAD software installed on their personal computers or workstations. The interaction techniques used when using these tools are restricted to the WIMP (windows, icons, mouse, and pointers) paradigm. Graphical User Interface (GUI) and accessibility to the functionalities of the CAD software are thus designed and revolved around this paradigm. In the Multi-Hand Gesture (MHG) interaction paradigm, a non-contact gesture recognition system is used to detect in real-time the hands and fingers movements, and also their positions in 3D space. These gestures are then interpreted and used to execute specific commands or tasks. This paper proposes a framework for a non-contact MHG interaction technique for architectural design. It also discusses 1) the WIMP and a non-contact MHG interaction technique, and 2) the development of an early prototype of a non-contact MHG recognition system. The prototype system consists of a software that is developed to recognize and interpret the multi-hand gestures captured by the Kinect sensor which does real-time scanning of three-dimensional space and depth.

Keywords-architectural design; interaction technique; multi-hand gesture; natural user interface

I. INTRODUCTION

In the Architecture, Engineering and Construction (AEC) industry, 2D or 3D computer aided design (CAD) software have been used by designers (architects and CAD operators alike) in the creation and presentation of architectural designs. CAD is used in all stages of architectural designs which are characterized by the different types of deliverables. The typical major design stages and the deliverables are: 1) Programming - conceptual and sketch drawings, 2) Schematic - detailed sketch drawings, 3) Preliminary - definitive detailed drawings, and 4) Final - Working drawings. Designers often starts with an abstract and conceptual design with vaguely-defined problems, then progresses to subsequent stages into more detailed design, to finally producing the blueprint or Working Drawings for buildings that are constructible by builders.

In all current CAD software, the interaction techniques used are still restricted to the WIMP (windows, icons, mouse, and pointers) paradigm [1]. The Graphical User Interface (GUI) and accessibility to the functionalities of the CAD software are thus designed and revolved around this paradigm.

This paper discusses a proposed framework for a non-contact Multi-Hand Gesture (MHG) interaction technique for architectural design. In this paper, the authors define MHG as gestures performed by a single or multiple users where each user is using either one or both hands. In a non-contact MHG interaction paradigm, a gesture recognition system is used to detect hand and finger movements, and their positions in real-time 3D space. These gestures are then translated into specific commands or tasks in the CAD software. In a non-contact MHG system, conventional intermediary interaction devices such as mice, wands, gloves, or markers are no longer required. The MHG system is also capable of supporting multiple users, whereby in a shared space, users can collaborate and perform tasks such as creating and reviewing designs.

The potential benefits to architectural design domain from the application and implementation of a non-contact MHG interaction technique are twofold. First, designers can interact intuitively and naturally when creating designs. Second, due to the non-present intermediary and non-invasive interaction devices, which sometimes can be a hindrance, designers will find that their ideas and creativity to be less restricted, and thus more expressive.

This paper also discusses 1) the WIMP and a non-contact MHG interaction technique, and 2) the development of an early prototype of a non-contact MHG recognition system. The prototype system consists of a software that is developed to recognize and interpret the MHGs captured by the Kinect sensor which does real-time scanning of three-dimensional space and depth.

II. WIMP VS. NON-CONTACT

In the AEC industry, drafting using 2D CAD software is an electronic extension of the traditional drawing board. 2D CAD
software are primarily used for creating plans and technical drawings, with all works being performed on the computer screen in place of a drawing paper. 3D CAD software ‘extends’ the drawing area in terms of allowing objects to be viewed from many different angles. When working with 3D CAD software, spatial awareness is required because the view is often changed for working on specific details of the 3D object. 3D objects created by 3D CAD software make use of solids that have width, depth and height, and are opaque (Ripley, 2010). With 3D CAD, the relationship among the 3D objects can be easily seen.

2D and 3D CAD software interfaces usually support both explicit and implicit style of interactions. Conventional interfaces support mainly the explicit style of interactions such as the use of a keyboard and mouse to navigate or point to particular objects. The interaction techniques are restricted to the WIMP (windows, icons, menus, and pointers) paradigm. Graphical User Interface (GUI) and accessibility to the functionalities of the CAD software are thus designed and revolve around this paradigm.

The WIMP together with the GUI has been the prevailing paradigm for the last three decades. In this paradigm, software and hardware are based on computer screen, keyboard, mouse, and their technical capabilities. The WIMP allows users to visualize objects on a two-dimensional plane from multiple angles or views in several windows. More technologically savvy architects use 3D CAD software such as 3DS Max, ArchiCAD or Revit to present virtual designs of buildings and facilities. Such software however still uses the 2D WIMP techniques for their menus and windows, and to manipulate 3D objects in the 3D virtual world. The final designs usually lack realism because, as stated by [2], “These techniques when used in 2D software that outputs various bi-dimensional views on a tri-dimension scene completely lack of realism. In addition, the limitation of bi-dimensional input devices, mainly mouse, keyboards or even 3D balls, does not provide intuitiveness of what users are designing.” Furthermore, according to [3], these interactions are often limited and unintuitive, and the devices are awkward, unwieldy and prone to distortion from the physical environment.

Implicit style of interactions allow more natural and easier to use human-computer interactions (HCIs) by allowing the arm, hand, head, or eye movement based interactions. Implicit style interactions are more complex to design compared to explicit style interactions. However, user has greater level of control and interaction, and can actively be involved. This kind of interaction is usually used in a 3D Virtual Environment (VE) using a more elaborate interaction devices that supports 3D spatial tracking in 3D space; such as the wand and data gloves. Using such devices, a user can interact with 3D virtual objects, and navigate by walking or flying through the VE with unconstrained movement with no predefined paths [4], [5].

III. GESTURE-BASED INTERACTION TECHNIQUES FOR ARCHITECTURAL DESIGN

Reference [6] describes architectural design as a creative process that combines art and engineering. Architects would prefer to quickly express their design ideas in 3D without any obstacle while they are in their creative design mode and while with inspirations. Reference [20] believes that sketches in 2D to represent 3D ideas and operating 3D CAD software are time consuming, and after such tedious operations, the inspiration and best design mode may have been long gone. As such, there is a need to develop a more intuitive 3D design method for architects to record their design ideas while they are still in their best inspiration and passion mode. Reference [20] further suggested a hand motion and gesture-based rapid 3D architectural modeling method. Reference [3] suggests the development of an alternative and natural interface that closely models the way of interaction with the real world, in place of the WIMP paradigm. Users should be able to reach out, grab, point and move 3D objects just as they do with real world objects.

Using a gesture-based system, an architect will be provided with a better and more immersive interaction with 3D space and virtual objects. Creating and manipulating 3D objects with one own hands, though virtually, would be more intuitive and natural than using an interface that uses intermediary devices for interactions. According to [3], the absence of intermediary devices and through direct manipulation interfaces, user’s cognitive load is minimized. Direct manipulation interfaces feature a natural representation of objects and actions that can hide the feeling of performing tasks through an intermediary i.e. the computer. The basic principle behind it is to allow users to directly perceive and interact with the 3D virtual objects which will lead to a more natural and effective interface. In a non-contact gesture-based interaction paradigm, a non-contact gesture recognition system is used to detect hand and finger movements and their positions in real-time, and uses these gestures to execute specific commands or tasks on the computer. Manipulation of 3D virtual objects is done with hand gestures. The architect designs and constructs the desired 3D virtual model while being immersed in the VE. Conventional intermediary WIMP devices such as mice, wands, gloves, or markers are no longer required.

To date there are three types of devices which are designed to capture motion data; glove-based devices, full body motion capture systems, and gesture recognition systems. Glove-based devices allow for the capture of detailed fingers’ and hands’ motions which in turn allows for the users to interact with objects in the VE [7], [8]. A full body motion capture system captures, tracks, and records full body movements and translates them to a digital model. Related research to this technology can be referred to [9] and [10]. Gesture recognition systems such as those developed by [11] use stereo vision and arm model fitting to recognize arm poses. Reference [12] developed a vision-based system that can interpret a user’s gestures in real time to manipulate windows and objects within a GUI.

Reference [13] reports on the works by scientists who developed a non-contact gesture and finger recognition system. The system does not require special gloves or markers and is capable of supporting multiple users. It detects multiple fingers and hands at the same time and allows the user to interact with objects on display. Users simply move their
hands and fingers in thin air, and the system will automatically recognizes and interprets the gestures accordingly. It is anticipated that this technology will open up new forms of knowledge exploration, particularly in the applications for complex 3D data simulation and visualization.

Reference [6] developed a method for generating 3D architectural models based on hand motion and design gestures captured by a motion capture system. A set of sign language-based gestures and architectural hand signs were developed. These signs are performed on the left hand to define various “components of architecture” such as external wall, internal wall, square pillar, and etc. The right hand holds a Marker-Pen to sketch out design geometry such as “location, size and shape” similar to a pen used by architects or designers. The hand gestures and motions are recognized by the system and then transferred into 3D curves and surfaces correspondingly. As a result, a rough 3D architectural model can be rapidly generated.

IV. THE CONCEPTUAL FRAMEWORK

In this paper, the authors propose a conceptual framework for utilizing non-contact MHG interaction technique for creating architectural design. The potential benefits to the architectural design domain from the application and implementation of this interaction technique are twofold, which have been mentioned earlier.

However, before such system can be developed, an exploratory research and experiments are required in deriving the components for the framework. Conceptually, and in reference to previous related works, for the system to be working and functional, the framework would consist of components as shown in Figure 1.

At the highest level, the framework is divided into three main components; the users, the software and the hardware; and the software and hardware having their own sub-components. Each component must coherently work together to form and function as a whole system. In the case of this particular non-touch MHG system, the end users are the architects or designers who will perform the specific gestures in relation to the intended architectural design tasks.

The second component is the hardware, and it consists of a gesture capturing device and large projection display. In this research, the authors use a single Kinect sensor as the gestures capturing device connected via a USB port on a Windows based computer. The Kinect is a motion sensing input device that allows users to naturally interact with game contents using gestures and voice commands; without the need of physically holding intermediary controller such as a game pad or joystick. Although it is one of the Xbox 360 game peripherals and was originally intended for game playing on the Xbox 360, the release of an open source driver by OpenNI has enabled the Kinect to be connected to a personal computer.

Due to this, many new research especially in natural user interfaces (NUIs) and interaction techniques are currently being conducted throughout the world. In support of future research and development in NUls, Microsoft recently released an official non-commercial SDK for the Kinect. The SDK allows for developers to write applications that can fully utilize the Kinect. In the next implementation iteration, the authors plan to use between three to four Kinect devices to provide a larger 360 degree capture area which in turn should reduce any possible occlusion issues when two or more users are occupying the same space. Besides the Kinect device and the PC, a large projection display is also used. While conducting a short experiment during development, the authors discovered that there is a minimum screen size required for the users to effectively manipulate objects on display. The minimum screen size must be no less than 27 inches diagonally. The users must also be at a minimum distance of 6 feet from the Kinect sensor. Due to this, a large projection display is preferable to ensure a better and immersive design experience. These requirements are the current limitation imposed by the Kinect sensor itself. However, as the technology improves in term of software and hardware, this may no longer be an issue.

The final part of the framework is the software itself. The software application currently being developed for this research is derived from the OGRE 3D engine. With the release of the official Kinect SDK from Microsoft, the authors are currently working on porting the application to use the SDK instead. At the very basic level within the software component, there are four sub-components; a VE, GUI, a library that contains a dictionary of multi-hand gestures, and databases to store and retrieve information, objects or designs. Presently, he main focus of this research is the establishment of a dictionary of multi-hand gestures purposely for handling architectural design related tasks in a VE.
A. The Proposed Gestures

With reference to [12], [11], [6], [13], it is proposed that the gestures which are suitable to be mapped and used for architectural design activities are: point and select, move, rotate, scale, zoom and pan, and extrude (Figure 2). Furthermore, [12] also defined the gestures shown in Figure 2 for controlling the windowing system for a vision-based system. This system can interpret a user’s gestures in real time to manipulate windows and objects within a GUI.

![Figure 2. Gestures defined for controlling windows by [12]](image)

Referring to Figure 2, [12] describes the Point gesture (a) is used to move the cursor on the screen. The Pick gesture (b) is used to select a window/object to manipulate. The Open gesture (c) is used to restore a window, while the Close gesture (c) is used to minimize window. The size of the window/object can be varied in different directions using different Enlarge and Shrink gestures. The Undo gestures can be used to reverse the previous action.

Figure 3 shows the left hand gestures as defined by [6] based on four main components of a simple building structure.

![Figure 3. Left hand gestures for basic components of building structure by Yi et al (2009)](image)

Reference [6] identified four main components of the building structure; the vertical components (vertical to the building plane, including main walls and columnisation), the transverse components (items parallel to the building plane - such as foundation, floor, roof and beam), doors and windows, and staircase. All vertical structures are expressed by vertical hand gestures, such as vertical palm (a) indicates a wall, while vertical (d) indicates a column. These gestures are designed for the left hand only while the right hand controls/holds a Marker Pen to input geometry such as location, size and shape.

Deriving from the work done by [12], [6], and since the non-contact MHG system can detect the user’s entire body gestures rather than just fingers, a reformulation is required to determine suitable gestures that involve the combination of the hands, arms, legs and body. The authors of this paper are also attempting to minimize the inclusion and use of the WIMP paradigm into the non-contact MHG system.

B. Proposed Gestures for Architectural Design

Some of the proposed gestures have been implemented into the prototype system (Figure 4). The gestures have been categorized into four different types; navigation, object creation, object selection and object transformation in the VE. An object can be selected by its face, vertex or line edge, and object transformation includes moving, rotating and scaling. Figure 5 shows some of the proposed gestures for a non-contact MHG system for architectural design in a VE.

![Figure 4. Testing the prototype system](image)

Navigation

- Move forward
- Move backward
- Rotate left
- Rotate right
- Strafe left
- Strafe right
Select & Move
Move up
Move down
Select & Rotate
Select & Scale
Face Select & Extrude

Object Creation
- Circle (line)
- Sphere (solid)
- Square (line)
- Square block (solid)
- Line

Object Selection
- Select whole object
- Select object by face
- Select object by vertex
- Select object by line/edge

Object Transformation

Figure 5. Examples of the proposed gestures

Even though the implementation of the basic gestures for navigation, object selection and object transformation have been implemented; further testing is required to confirm that the gestures are truly suitable for the specific architectural design task in the VE.

C. The Prototype System

In support of this proposed framework, a prototype system is currently being developed. The system utilizes Microsoft Kinect as the gestures recognition device, with the open source Ogre 3D Engine as the VE software development tool. When the research work first started, Ogre is used because Microsoft did not have the official Kinect SDK released at that time. Ogre is one of the few SDKs that can directly interact with the OpenNI open source Kinect device driver.

The prototype system is developed with a collaborative design approach in mind. At the moment, it supports up to nine individual users, and can recognize up to 18 different hands. However, due to the resolution limitation of the Kinect, the 3D scanning area and how far of a distance the Kinect device can be to effectively recognizes a user’ gestures is yet unknown. It is also yet unknown whether having nine users collaborating and simultaneously using the system at the same time would be efficient especially in terms recognizing the gestures from different users. However, a brief testing confirmed that the current system can comfortably and effectively accommodate up to three users within a 6x6 square-foot area. Since the architectural design space exists in a real-time 3D VE, before any architectural design related gestures are implemented, the basics and common gestures were developed and tested first. These common gestures focus on navigation and object selection in the VE.

V. CONCLUSION

This paper discussed a proposed framework for a non-contact Multi-Hand Gesture (MHG) interaction technique for architectural design. The MHG system is capable of supporting
multiple users. In a shared space, users can collaborate and perform tasks such as creating and reviewing designs. There are two main benefits that can be realized from the application and implementation of a non-contact MHG interaction technique in the architectural design domain. First, when creating designs, architects can experience a more natural and intuitive interaction with their creation. Second, architects will find that their ideas and creativity to be less restricted, thus more expressive due to not having to use intermediary and invasive interaction devices such as the mouse and keyboard. This paper also discussed 1) the WIMP and a non-contact MHG interaction technique, and 2) the development of an early prototype of a non-contact MHG recognition system. The prototype system consists of a software that is developed to recognize and interpret the MHGs captured by the Kinect sensor which can scan three-dimensional space and depth in real-time.

Through the use of an affordable gestures capturing device such as the Kinect, the framework opens up new avenues for research in NUIs in general, and more specifically NUIs that support architectural design related tasks in a VE. However, much research work is still needed especially in determining the most suitable MHG for specific architectural design related tasks. During the prototype system development cycle, no rigorous usability studies on the gestures were conducted. Future work will include a more quantitative approach to determine the most suitable gesture for a particular architectural design task. Once suitable gestures have been determined, the formulation and creation of a standardized library of multi-hand gestures will also be required.

ACKNOWLEDGMENT

The authors would like to thank Mr. Nicholas May, Mr. Mohammed Murad, Mr. Peter Rennie, Mr. Hamed Sedigh & Mr. Trent Varney for their assistance in developing the prototype system; and Associate Professor Peter Cole and Dr. Graham Mann for their support in this research.

REFERENCES