A PROJECT REPORT ON

Virtual Mirror – A Hassle Free Approach to the Use of Trial Room

Submitted by:

S Dhanashree 4SO14CS085
Mohmmed Awasf Ali 4SO15CS413
Prithvi V 4SO15CS415
Ranjith Kumar B 4SO15CS416

In partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of:

Ms Nisha J Roche
(Assistant Professor, CS&E Department)

DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING

ST JOSEPH ENGINEERING COLLEGE
(Affiliated to VTU-Belagavi, Recognized by AICTE, NBA Accredited)
Vamanjoor, Mangaluru – 575 028, Karnataka.

2017-2018
A PROJECT REPORT ON

Virtual Mirror – A Hassle Free Approach to the Use of Trial Room

Submitted by:

Name          USN          Signature
S Dhanashree  4SO14CS085
Mohmmmed Awsaf Ali  4SO15CS413
Prithvi V       4SO15CS415
Ranjith Kumar B  4SO15CS416

In partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

Under the Guidance of:

Ms Nisha J Roche
(Assistant Professor, CS&E Department)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ST. JOSEPH ENGINEERING COLLEGE
(Affiliated to VTU Belagavi, Recognized by AICTE, B.E. (CSE)-NBA Accredited)

Vamanjoor, Mangaluru – 575 028, Karnataka.

2017-2018
ACKNOWLEDGMENT

We wish to express our sincere gratitude to our guide Ms Nisha J Roche, Assistant Professor, Department of Computer Science and Engineering, for her valuable guidance and suggestions, hence improving our knowledge.

We would also like to express our sincere gratitude to our coordinator Ms Supriya Salian, Assistant Professor, Department of Computer Science and Engineering, for her valuable inputs and constant support in carrying out our project and helping make it better.

We are grateful to Dr Sridevi Saralaya, Head of the Department of Computer Science and Engineering, for providing us with the right ambience to carry out our project and for her unfailing encouragement.

We are indebted to our respected Principal Dr Joseph Gonsalvis, beloved Director Rev. Fr Wilfred Prakash D’Souza, Assistant Director Rev. Fr Rohith D’Costa, the Vice Principal Dr Rio D’Souza and the Management of St Joseph Engineering College, Vamanjoor, Mangaluru for having provided all the facilities that helped us in the timely completion of this project.

Finally, we would like to thank all the Teaching and Non-teaching staff of the Department of Computer Science and Engineering for their valuable help and support.
ABSTRACT

In-store shopping experience is still the first thing that springs to mind for a majority of people when they think of shopping. Although customers can try out clothes in real time, this process is too time consuming whenever there are insufficient trial rooms. The project Virtual Mirror can change the way a person tries out clothes and shop for the right one. Using the concept of “Virtual Reality”, the customers can try out a large variety of clothes without the need of physically wearing them.

The concept of a Virtual Trial Room, has been implemented by various research groups using different approaches. One of the approaches involves the use of humanoid models to simulate the movements of the user and enables users to try the clothes on the model. Another approach uses a fixed static model of the clothing which is displayed on the screen and the user must position themselves according the position of the cloth.

The “Virtual Mirror” implements a real-time virtual trial room system using the Kinect Sensor which senses the human body and then uses this data to superimpose the clothes. The system first captures the image before it using the Kinect Sensor. The data generated is the skeletal data of the user which is then sent to Unity. This skeletal data is then used in Unity to generate the clothes and impose it on the user. A livestreaming video of the user with the clothes imposed is then displayed as the output on the screen.

The advantage of using this method would be the reduction of time and effort spent in trying out the clothes physically. The customers can then buy the clothes by scanning the QR code provided on the screen. This project helps in Market Management so that a customer need not try each and every clothes. Also, Retailers need not have a huge stock in the store thus saving time and space.

This project can be improvised by including gesture recognition for selection of clothes, thereby eliminating the need of an external device to select clothes.
# I. TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Title</th>
<th>Page No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.1 Problem Definition</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2 Scope and Importance</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Software Requirement Specification</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.1 Overall Description</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.1.1 Product Perspective</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.1.2 Product Functions</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.1.3 User Characteristics</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.1.4 Specific Constraints</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.1.5 General Constraints</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2 Specific Requirements</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2.1 External Interface Requirements</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2.2 User Interface</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2.3 Hardware Interface</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2.4 Software Interface</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2.5 Communicational Interface</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2.6 Functional Requirements</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.3 Performance Requirement</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.3.1 Standard Compliance</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.3.2 Availability</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.3.3 Reliability</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.4 Design Constraints</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Design</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>3.1 Abstract Design</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>3.1.1 Architectural Diagram</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>3.1.2 Use Case Diagram</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>3.2 Functional Design</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>3.2.1 Modular Design Diagram</td>
<td>7</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Data Flow Diagram</td>
<td>8</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
<td>---</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Sequence Diagram</td>
<td>10</td>
</tr>
<tr>
<td>3.3</td>
<td>Control Flow Diagram</td>
<td>11</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Complete System Flow Diagram</td>
<td>11</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Activity Diagram</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td><strong>Implementation</strong></td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Hardware Requirements</td>
<td>14</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Kinect Sensor</td>
<td>14</td>
</tr>
<tr>
<td>4.1.2</td>
<td>LCD Monitor</td>
<td>15</td>
</tr>
<tr>
<td>4.1.3</td>
<td>USB 3.0</td>
<td>15</td>
</tr>
<tr>
<td>4.2</td>
<td>Software Requirements</td>
<td>16</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Kinect SDK</td>
<td>16</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Unity</td>
<td>16</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Mono Develop</td>
<td>17</td>
</tr>
<tr>
<td>4.3</td>
<td>Pseudocode</td>
<td>17</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Pseudocode for Virtual Mirror</td>
<td>17</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Pseudocode for AvatarCloth</td>
<td>18</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Pseudocode for FittingRoom Sample</td>
<td>18</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Pseudocode for LongSleeve</td>
<td>21</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Pseudocode for TShirt</td>
<td>21</td>
</tr>
<tr>
<td>4.3.6</td>
<td>Pseudocode for Pants</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td><strong>Testing</strong></td>
<td>23</td>
</tr>
<tr>
<td>5.1</td>
<td>Test Cases</td>
<td>23</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Test Case for User’s Distance</td>
<td>23</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Test Case for User’s Angle</td>
<td>24</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Test Case for Shoulder Detection</td>
<td>24</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Test Case for Number of Users</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td><strong>Conclusion and Future Scope</strong></td>
<td>26</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusion</td>
<td>26</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Scope</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Appendix</td>
<td>28</td>
</tr>
</tbody>
</table>
## II. LIST OF TABLES

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Title</th>
<th>Page No</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Test case for user’s Distance with respect to Kinect</td>
<td>24</td>
</tr>
<tr>
<td>5.2</td>
<td>Test case for Angle of the User with respect to Kinect</td>
<td>24</td>
</tr>
<tr>
<td>5.3</td>
<td>Test case for Shoulder Detection of the User</td>
<td>25</td>
</tr>
<tr>
<td>5.4</td>
<td>Test case for Number of Users</td>
<td>26</td>
</tr>
</tbody>
</table>
### III. FIGURE INDEX

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Title</th>
<th>Page No</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Architectural Diagram</td>
<td>5</td>
</tr>
<tr>
<td>3.2a</td>
<td>Use Case Diagram for Admin</td>
<td>6</td>
</tr>
<tr>
<td>3.2b</td>
<td>Use Case Diagram for User</td>
<td>7</td>
</tr>
<tr>
<td>3.3</td>
<td>Modular Design Diagram</td>
<td>8</td>
</tr>
<tr>
<td>3.4a</td>
<td>Data Flow Diagram Level 0</td>
<td>9</td>
</tr>
<tr>
<td>3.4b</td>
<td>Data Flow Diagram Level 1</td>
<td>9</td>
</tr>
<tr>
<td>3.5</td>
<td>Sequence Diagram</td>
<td>11</td>
</tr>
<tr>
<td>3.6</td>
<td>Control Flow Diagram</td>
<td>12</td>
</tr>
<tr>
<td>3.7</td>
<td>Activity Diagram</td>
<td>13</td>
</tr>
<tr>
<td>3.8</td>
<td>Skeletal Tracking</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Kinect Sensor</td>
<td>16</td>
</tr>
<tr>
<td>4.2</td>
<td>LCD Monitor</td>
<td>16</td>
</tr>
<tr>
<td>4.3</td>
<td>USB 3.0</td>
<td>17</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Problem Definition

Stores generally have a large and varied options for clothes. It is physically impossible for a customer to try out all those clothing without having to spend hours on it. Also, in a physical store, in order to try on some selected clothes a common practice is to queue up and take turns to use the fitting rooms. Due to the limited number of in-store fitting rooms, shoppers usually have to spend most of their shopping time on queuing up (which will be even longer during peak hours). Prolonged waiting time will affect customer’s patience, which leads to lower customer satisfaction.

1.2 Scope and Importance

The main scope of the project Virtual Mirror – A Hassle Free Approach to the Use of Trial Room is to provide a virtual trial room to its users. Here an idea called the “Virtual Try On” using The Virtual Mirror is put forward. Customers would be able to perceive the visual image of how they look by trying on clothes virtually, so that they can easily filter out some selections without taking turns to use the fitting rooms and the embedded feature of e-commerce can be utilized to make purchases more convenient. Comparing with “Physical Try On”, “Virtual Try On via a Virtual Mirror” takes much less time. Thus, it increases the shopping efficiency for all customers and enhances the shopping experience.
CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Overall Description

2.1.1 Product Perspective

This product is Augmented Reality based system. Virtual mirror using Kinect sensor proposes a virtual dressing room application to help users to try virtual garments in front of a virtual mirror.

The application displays as the output an augmented image of the user with the clothes super imposed over the user’s body. Augmented reality is used to develop the system we need to create.

2.1.2 Product Functions

In our Virtual Mirror the user must stand in front of a screen fitted with a Kinect sensor. The Kinect scans the human body from the environment using an algorithm. The monitor which is connected with the Kinect sensor and the system processor displays the users image with clothes fitted on them.

2.1.3 User Characteristics

Our design vision is such that the system should do the majority of the work for the user. The user just needs to interact with the system to state their needs.

2.1.4 Specific Constraints

These constraints are the specific factors that affect the product and its requirements. Some of the specific constraints are:

- The application is language specific. No other language except English will be accepted.
- The user must be located at a distance of at least 5 feet from the Kinect Sensor and must not move to a distance greater than 8 feet from it. If the user moves outside the range specified, the desired output may not be produced.
• The customer can not be aligned at an angle greater than 45°.
• The system requires continuous power supply.

2.1.5 General Constraints

• Only one customer must stand in front of the mirror at a time.
• User must not be able to alter the basic functionality provided by the system.
• Developer must have an in-depth knowledge of C# and Unity.

2.2 Specific Requirements

2.2.1 External Interface Requirements

This provides detailed description of all inputs and outputs from the software. This requirement is organized in the following subsection.

2.2.2 User Interface

• Front-end software: Unity, HTML
• Back-end software: PHP, C#

2.2.3 Hardware Interface

• The Kinect Sensor: To predict the node points of the human body
• USB 3.0: To support Kinect
• A monitor with HDMI-in: To project the image
• USB power adapter: To get the continues power supply.

2.2.4 Software Interface

Operating System: Windows 10

Editor: Unity, MonoDevelop, Paint 3D

2.2.5 Communicational Interface

• This application uses internet connection to connect with the web.
2.2.6 Functional Requirements

The Virtual Mirror is expected to provide its user the following features:

- The system shall capture the image before it is using the Kinect Sensor.
- The data generated is used to generate an image of the user.
- A rendering technique is used to simulate a virtual environment.
- The cloth models are displayed to cover the human body which is then displayed in the colour video frame.

2.3 Performance Requirement

2.3.1 Standard Compliance

The system will be efficient and will not have any kind of any delay. The graphical user interface will have a consistent look and feel. The image processing will be effective enough to detect changes in the frames.

2.3.2 Availability

The system will be available at all times as long as the system is in proper working condition. The functionality of the system will depend on the hardware resources such as the Kinect sensor and the availability of power supply.

2.3.3 Reliability

The system will be reliable and should be able to provide the desired output at all times without any errors. The system is optimized to provide error free services.

2.4 Design Constraints

- Guidelines are provided for the user
- Requires physical space for the system
- Requires a well-lit space.
CHAPTER 3

DESIGN

3.1 Abstract Design

Design represents the development phase for any engineering product or system. It is defined as the process of applying various techniques and principles for the purpose of defining a process of a system in sufficient detail. Design is the first step in moving from problem domain to the solution domain. The goal of the design process is to produce a model or representation of a system, which can be used later to build that system.

3.1.1 Architectural Diagram

The design process of identifying subsystems and establishing the framework for a subsystem control and communication is called architectural design. It is a creative process where one tries to establish a system organization that will satisfy the functional and non-functional requirements of the system. It represents the structure of data and program components. It represents a set of abstraction that enable software engineers to describe architecture in predictable ways. The product of the architectural design process is an architectural design document which includes graphical representations of the system along with associated descriptive text.

The architectural design process is concerned with the establishment of a structural framework. It defines the major components of a system and communication between those components. Figure 3.1 gives an architectural overview of the project.

![Figure 3.1 Basic Architectural Diagram of Virtual Mirror](image)
3.1.2 Use Case Diagram

The use case diagram is a graphical notation of the interaction among elements. A use case is a list of actions or event steps, typically defining the interactions between a role and a system, to achieve a goal. The use case diagram is used to identify the primary elements and the processes that form the system. The primary elements are called as “actors” and the processes are called as “use cases”.

Use case diagrams is a graph of actors, a set of use cases enclosed by a system boundary, communication association between actors and the use cases. It describes how system interacts with outside actors; each use case represents piece of functionality that the system provides to the users. The use case diagram consists of ellipses containing name of the use case and a stick figure that depict the actor with his name. All use cases are enclosed within a rectangle.

Figure 3.2a and Figure 3.2b shows the use case diagram for Admin and the User for the Virtual Mirror respectively. The admin can install Kinect add and remove clothes. The user can select, try and also buy the clothes.

![Use case diagram for Admin](image.png)
3.2 Functional Design

Functional design is the model used to simplify the design of hardware and software devices. The functional design assures that each modular part of the device has only one responsibility with minimum side effects on the other parts.

3.2.1 Modular Design Diagram

Modular design is a design approach that subdivides a system into smaller parts called modules, which can be independently created and then used in different systems. A modular system can be characterized by functional partitioning into discrete scalable, reusable modules, rigorous use of well-defined modular interfaces, and making use of industry standard for interfaces. The Figure 3.3 provides the modular diagram for the Virtual Mirror.
3.2.2 Data Flow Diagram

The Data Flow Diagram (DFD) is the graphical representation of the flow of data between the various processes in the system. Each actor is involved in triggering various events that lead to the data transmission among various components. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFD’s can also be used for the visualization of data processing. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. Figure 3.4a and 3.4b shows the dataflow diagram of the Virtual Mirror.
VIRTUAL MIRROR - A Hassle Free Approach to the Use of Trial Room

Figure 3.4a Data Flow Diagram Level 0 of the Virtual Mirror

Figure 3.4b Data Flow Diagram Level 1 of the Virtual Mirror
3.2.3 Sequence Diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and shows the order of process operation. It depicts the participants in an interaction and the sequence of messages among them. It also shows the interaction of the system with its actors to perform a use case.

A sequence diagram shows the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with the use case realization in the Logical View of the system under development.

In the sequence diagram, each actor as well as system is represented by a vertical line called the life line and each message by a horizontal arrow from sender to receiver. The parallel vertical lines show different objects that live simultaneously. Time proceeds from top to bottom in this diagram. Figure 3.5 shows the sequence diagram for Virtual Mirror.

- NUI is the component that communicates with Kinect Sensor. Depends on whether a user is detected, NUI asks Stage components to prepare different scenes and asks DrawDevice to render the prepared scenes.
- DrawDevice is the component that displays the tasks using the graphic card. It renders whatever 3D virtual scenes that were prepared by Stage onto the screen.
- Stage is the component that manages what should be displayed. If there is no user detected, Stage does not need to prepare any user motion data into current scene. Otherwise, Stage loads current user position data and integrates the data with virtual clothes to form the current scene. Meanwhile, Stage also asks Pose to detect gestures and updates Stage status accordingly.
- Pose is the component that stores and analyses motion data. Pose keeps updating motion data history once someone is detected. It also contains functions to recognize gestures.
3.3 Control Flow Diagram

3.3.1 Complete System Flow Diagram

System Flow Diagram is basically a graphical and sequential representation of the major steps involved in a systematic process. A System Flow Diagram shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. Figure 3.6 shows the Control Flow Diagram for the Virtual Mirror.
3.3.2 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is a flow chart to represent the flow form of one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagram deals with all types of flow control by using different elements like fork, join etc.

Activity Diagram are also useful in formulating a use case by describing what action needs to take place and when they should occur. It describes complicated sequential algorithm and modelling applications with parallel processes. Activity diagrams should not give details about
how the objects behave or how objects collaborate. The Figure 3.7 shows the Activity Diagram for Virtual Mirror.

Figure 3.7 Activity diagram for Virtual Mirror
CHAPTER 4

IMPLEMENTATION

Implementation is the realization of a technical specification or algorithm as a program, software component or other computer system. In this phase the design of the system is translated into code and the aim is to implement the design in the best possible manner. Implementation plays an important role in system development life cycle. It is a stage that transforms the design into working module.

Implementation is the final and important phase of software development. It refers to conversion of new system design to an operation. Implementation produces reusable, extensible and robust code. Implementation is the process of guiding a client from purchase to the software or hardware that was purchased. This includes requirement analysis, scope analysis, customization, system integration, user policies, user training and delivery.

4.1 Hardware Requirements

4.1.1 Kinect Sensor

Kinect (code named Project Natal during development) is a line of motion sensing input device that was produced by Microsoft for Xbox 360 and Xbox One video game consoles and Microsoft Windows PCs. Based around a webcam-style add-on peripheral, it enables users to control and interact with their console/computer without the need for a game controller, through a natural user interface using gestures and spoken commands. It is a highly innovative combination of cameras, microphones and software that turns your body into the video game controller.

The name Kinect is inspired by the words "kinetic," which means to be in motion, and "connect," which means it "connects you to the friends and entertainment you love" The innovative technology behind Kinect is a combination of hardware and software contained within the Kinect sensor accessory that can be added to any existing Xbox 360.

The Kinect sensor is a flat black box that sits on a small platform, placed on a table or shelf near the television. There is a trio of hardware innovations working together within the Kinect sensor and they are:
• Color VGA video camera - This video camera aids in facial recognition and other detection features by detecting three color components: red, green and blue. Microsoft calls this an "RGB camera" referring to the color components it detects.

• Depth sensor - An infrared projector and a monochrome CMOS (complementary metal-oxide semiconductor) sensor work together to "see" the room in 3-D regardless of the lighting conditions.

• Multi-array microphone - This is an array of four microphones that can isolate the voices of the players from the noise in the room.

![Figure 4.1 Kinect Sensor](image)

### 4.1.2 LCD Monitor

Liquid crystal display (LCD) monitor is a computer monitor or display that uses LCD technology to show clear images and is found mostly in laptop computers and flat panel monitors.

![Figure 4.2 LCD Monitor](image)

### 4.1.3 USB 3.0

USB 3.0 is a Universal Serial Bus (USB) standard, released in November 2008. Devices that adhere to the USB 3.0 standard can theoretically transmit data at a maximum rate of 5 Gbps, or 5,120 Mbps. This is in stark contrast to previous USB standards, like USB 2.0, that at best can only transmit data at 480 Mbps or USB 1.1 that tops out at 12 Mbps.
Kinect does not function on a USB2 port or controller. The Kinect for Windows v2 sensor will not work on a USB 2 port or controller. To use the sensor, you must have a USB 3 port or controller. Only USB 3 controllers from Intel and Renesas are supported. If you use a different brand of USB 3 controller, the Kinect sensor may not function correctly. For example: The sensor may fail to enumerate, Depth may not stream, Colour may not stream, the sensor may stop functioning, Packet loss may increase.

4.2 Software Requirements

4.2.1 Kinect SDK

Official Microsoft Kinect for Windows SDK for building software, using C++, C# or Visual Basic, targeted at the Kinect for Windows hardware.

The Kinect for Windows SDK is Microsoft's official development kit for developing software applications targeting the Kinect for Windows hardware, with language support for C++, C# and Visual Basic. The Kinect for Windows SDK allows developers to create a touchless and immersive user experience through voice, movement and gesture control using the Kinect hardware.

4.2.2 Unity

Unity is a cross-platform game engine developed by Unity Technologies, which is primarily used to develop both three-dimensional and two-dimensional video games and simulations for computers, consoles, and mobile devices. First announced only for OS X at Apple's Worldwide Developers Conference in 2005, it has since been extended to target 27 platforms. Six major versions of Unity have been released.
4.2.3 Mono Develop

Mono Develop (also known as Xamarin Studio) is an open source integrated development environment for Linux, macOS, and Windows. Its primary focus is development of projects that use Mono and .NET frameworks. Mono Develop integrates features similar to those of NetBeans and Microsoft Visual Studio, such as automatic code completion, source control, a graphical user interface (GUI) and Web designer.

4.3 Pseudocode

Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm. It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading. It is used for creating an outline or a rough draft of a program. Pseudocode summarizes a program’s flow but excludes underlying details. System designers write pseudocode to ensure that programmers understand a software project’s requirements and align code accordingly.

4.3.1 Pseudocode for Virtual Mirror

The Virtual Mirror composed of five major scripts, namely:

1) AvatarCloth
2) FittingRoomSample
3) LongSleeve
4) TShirt
5) Pants

The Pseudocode for the Virtual Mirror would thus be:

BEGIN

Enter the Script for FittingRoomSample.

Initialize AvatarCloth with the values obtained from FittingRoomSample.

IF selected cloth = Long Sleeve

THEN use LongSleeve Script.

ENDIF
IF selected cloth = T-Shirt

    THEN use TShirt Script.

ENDIF

IF selected cloth = Pant

    THEN use Pants Script.

ENDIF

END

4.3.2 Pseudocode for AvatarCloth

BEGIN

    Set pivot of type Spinebase.

    Declare object PivotOrigin which is automatically initialized to the pivot value of the cloth when placed on screen.

    Declare object ScaleOrigin which is automatically initialized to the scaling value for the cloth when placed on the screen.

    FUNCTION OnInitialized ()

        Set the PivotOrigin to the pivot position of the user dynamically obtained from Kinect.

        Set ScaleOrigin to the value by which the cloth needs to be scaled for the user obtained dynamically from the Kinect.

    FUNCTION OnReset ()

        Reset the values of PivotOrigin and ScaleOrigin to the Original values.

END

4.3.3 Pseudocode for FittingRoomSample

BEGIN
Declare an instance of Body.

Declare an array to hold the Clothes.

Declare an array to hold the selected clothes called selected

**FUNCTION** Awake ()

Enable the Package called Avateering

FOR each of the clothes defined

Initialize the Clothes array.

**ENDFOR**

**FUNCTION** OnApplicationQuit ()

Disable Avatareering

FOR each cloth in the array

Dispose the clothes in the array.

**ENDFOR**

**FUNCTION** Update ()

READ Visualization data from Kinect

**CASE OF** Visualization

‘Colour’: Update Colour Frame values

‘Depth’: Update Depth Frame values

Other: Update Infrared Frame values

**ENDCASE**

Update the Body Frame

**IF** body! = NULL

**THEN**
FOR each element in the selected array

IF an element is selected

THEN

Corresponding cloth is selected.

Avateering package is initialized with the value of this cloth and the body values.

IF user distance is not outside the Frame

THEN Render the cloth on the user

ENDIF

ENDIF

ENDFOR

ELSE

Reset the value of clothes.

ENDIF

FUNCTION OnGui()

FOR each cloth in the clothes array

Get position of the cloth

Create a button below the respective cloth

IF Button pressed

THEN

IF selected==True

THEN set selected = False

ELSE
4.3.4 Pseudocode for LongSleeve

BEGIN

FUNCTION OnUpdate ()

Update positions for SpineBase, SpineMid, Neck, Shoulder Left, Elbow Left, Wrist Left, Shoulder Right, Elbow Right, Wrist Right in the Avateering Package using data from Kinect

END

4.3.5 Pseudocode for TShirt

BEGIN

FUNCTION OnUpdate ()


Update positions for SpineBase, SpineMid, Neck, Shoulder Left, Elbow Left, Shoulder Right, Elbow Right, in the Avateering Package using data from Kinect.

END

4.3.6 Pseudocode for Pants

BEGIN

FUNCTION OnUpdate ()

Update positions for SpineBase, Hip Left, Knee Left, Ankle Left, Hip Right, Knee Right, Ankle Right in the Avateering Package using data from Kinect

END
CHAPTER 5

TESTING

5.1 Test Cases

5.1.1 Test case for User’s Distance with respect to Kinect

Table 5.1 Test case for User’s Distance with respect to Kinect

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Test Cases</th>
<th>Expected Outcome</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>When the user is at a distance between 5 feet and 8 feet from the Kinect sensor</td>
<td>User with the close imposed properly</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>When the user is beyond 8 feet distance from Kinect sensor</td>
<td>The cloth is not imposed on the user</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>When the user at a distance lower than 5 feet from the Kinect sensor</td>
<td>The imposed image should be distorted</td>
<td>Pass</td>
</tr>
</tbody>
</table>
5.1.2 Test case for Angle of the User with respect to Kinect

Table 5.2 Test case for Angle of the User with respect to Kinect

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Test Cases</th>
<th>Expected Outcome</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>When the user is parallel and opposite to the sensor</td>
<td>The cloth imposed perfectly on the user</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>When the user is at an angle up to 45 degrees clockwise.</td>
<td>The cloth is imposed on the user as per the user’s angle orientation.</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>When the user is at an angle up to 45 degrees anticlockwise.</td>
<td>The cloth is imposed on the user as per the user’s angle orientation.</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>When the user is at an angle greater than 45 degrees both clockwise and anticlockwise.</td>
<td>It should give the distorted image.</td>
<td>Pass</td>
</tr>
</tbody>
</table>

5.1.3 Test case for Shoulder Detection of the User

Table 5.3 Test case for Shoulder Detection of the User

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Test Cases</th>
<th>Expected Outcome</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>When both the shoulders are detected</td>
<td>A live video stream of the user with the cloth imposed perfectly</td>
<td>Pass</td>
</tr>
<tr>
<td>Sl. No</td>
<td>Test Cases</td>
<td>Expected Outcome</td>
<td>Result</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------</td>
<td>-------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>If the number of user is just one</td>
<td>Live video stream of user with cloth imposed</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>If the number of users is greater than one</td>
<td>The cloth is imposed on any one user who meets the constraints</td>
<td>Pass</td>
</tr>
</tbody>
</table>

### 5.1.4 Test case for the Number of Users

Table 5.4 Test case for the Number of Users
CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

A common problem faced by customers while shopping for clothes is the need to spend hours trying out a variety of clothes physically. This can be tiring and the time available might be short. The proposed solution to overcome this problem is the use of a Virtual Mirror that acts as a virtual trial room. It uses a Kinect sensor to plot nodes points of the human body and this data is then used to render the image of clothes over the user’s body, thereby eliminating the need to physically try on the clothes and hence helping save time.

6.2 Future Scope

The Virtual mirror project proposed here allows the customers to choose the clothes of their choice by using a tablet which is then identified by the mirror. This project can be improvised by including gesture recognition for selection of clothes, thereby eliminating the need of an external device to select clothes.

The Project works currently for male garments like T-shirt (full sleeve, half sleeve), Pants and Shorts. The Project can be improvised to include female garments also.
REFERENCES


[3] LanZiquan, Augmented Reality – Virtual fitting Room using Kinect, Department of Computer Science, School of Computing, National University of Singapore 2011/2012


[8] https://docs.unity3d.com

[9] https://msdn.microsoft.com
SNAPSHOTS

Figure 1 Image of the Home page

Figure 2 Image of Long Sleeve T-shirt Menu

Figure 3 Image of Screen when VM Red selected

Figure 4 Image of User with VM Red T-Shirt
VIRTUAL MIRROR - A Hassle Free Approach to the Use of Trial Room

Figure 5 Image of Pant Menu

Figure 6 Image of User with VM Pant