Abstract

1 Introduction
1.1 Project Background and Description
1.2 Problem Statement
1.3 Rationale
1.4 Project Objective
1.5 Stakeholders
1.6 Project Scope

2 Research
2.1 Academic Research
2.1.1 Language
2.1.2 Library
2.1.3 Algorithm
2.1.4 Model and texture
2.1.5 Motion sensing software development kits
2.1.6 Unity3D
2.1.7 WPF
2.1.8 Kinect
2.1.9 Microsoft Blend
2.2 Secondary Research
2.2.1 Findings of similar Developed System

3 System Analysis
3.1 Analysis
3.1.1 Functionality
3.1.2 Communication with the user
3.1.3 Shortcomings or Caveats of the testing
3.2 Implication
3.2.1 Cursor

3.2.2 Measurement
3.2.3 Measurement-taking process

4 Requirement Specification
4.1 Functional Requirement
4.1.1 Use case diagram
4.1.2 Use case specification
4.2 Technical Requirements
4.3 Gantt Chart

5 System Design
5.1 Human Computer Interaction (HCI)
5.2 Class Diagram
5.3 Database Design
5.4 Flowchart

6 Development
6.1 Pre-processing
6.1.1 User extraction
6.1.2 Tracking
6.1.3 Model Positioning and Rotation
6.1.4 Point mapping
6.1.5 Model Scaling

7 Testing and Evaluation
7.1 Unit Testing
7.2 Methods
7.3 Test results

8 Conclusion
8.1 Success criteria

8.2 Limitation
8.3 Future Enchancement
9 References

AugDrobe3D - Augmented Reality Virtual Fitting Room with Kinect and Unity3D

Choo Kok Hong

Rimi Aziz Ramidin

Sanath Sukumaran

Programming Project 1

May 22, 2013

Abstract

This paper presents an augmented reality fitting room application software which allows single user to try on virtual apparels or clothings. The user pose is tracked using the Microsoft Xbox Kinect sensor and the virtual clothes are mapped accordingly with the user silhouette and poses. Clothings moves and folds realistically and the lighting intensity of the cloth render is adapted to match ambient lighting conditions.

The presented application software improves on related augmented reality applications by adding full user pose tracking and by using 3D clothing models, combined with cloth simulation instead of 2D images. Skeletons are mapped accordingly onto the cloth models accordingly to the user's body size.

1 Introduction

Back in 2009, Microsoft developed the Kinect sensor under the codename 'Project Natal' and released it in the end of 2010 and started revolutionising the gaming industry.

The Kinect sensor has interesting capabilities such as allowing one to generate a depth image alongside an RGB camera image. It comes with tools that provide human pose detection and tracking as well. It is capable of simultaneously tracking up to six users, including two active players for motion analysis with a feature extraction of 20 joints per player. These abilities can be used to create an immersive virtual reality presence for the user such as Dance Central 3 and also to create augmented reality application in which virtual objects interact with the user and his surrounding environment.

1.1 Project Background and Description

AugDrobe3D is developed by a team of four students which is pursuing their Bachelor Degree of Computer Science by RMIT University. This application is developed under Windows Platform with the use of Kinect sensors. This application is using Kinect sensors to capture input and after processing, shown on a display.

It is a virtual wardrobe that provides user convenience to try on different apparels. It provides an estimation of how the apparel would be look like on the user. Besides, apparel matching, which is sometime a headache for someone could be done easily with least amount of time. This application allows user to change the color of the apparel or even the accessories with just a hover or a click. The user can save the time of taking off and putting the clothes and also the time taken for queuing to the fitting room. All the user needs is to do is to stand in front of the Kinect sensors and they can start trying on apparels of their choice. If the user likes the combination, they can even take screenshots of it and share to their Social Networking Service such as Facebook or Twitter.

The Microsoft Xbox Kin-ect sensor is used in this project to create an augmented reality wardrobe in which the user could try on virtual clothing and apparels. The silhouette of the user is tracked to allow the clothing to move and scale accordingly with the user and the depth image is used to create an avatar of the user that approximates the user's body shape.

Next, cloth simulation is applied to the virtual clothing to make it move and fold realistically based on the user's movements. The depth image from the sensor is used to compute the girth of the user's body to adapt the user avatar and recommend clothing sizes to the user. The RGB image is used as a background over which the clothing is projected and is displayed on the user avatar when it occludes parts of the clothing.

The user is segmented from the background and the i00+

ntensity of this part of the image is calculated to adapt the lighting of the virtual clothes. This makes the clothing appear as if it is in the same room as the user by reacting to bright and shaded parts of the environment.

1.2 Problem Statement

People sometimes will meet the situation that waiting for a long queue to try on apparels. The reason of long queue normally is because the shop is lack of fitting room or it is during the weekend, public holiday and sales period. Even worst, there are always more than one size and color that they might want to test, this will make others waiting much longer to try their apparels. Some customer will just wait inside the fitting room and request the staff to change the size or color for them. When the customer losing their patient it will result the customer is not happy on apparel what they have choose.

1.3 Rationale

The increasing demand of clothes especially online shopping is evident. Abundant and fast access to latest styles and newest season of clothes has two main advantage: it enables people that are active in the fashion field access to all newest available apparels to try on immediately as it released; and increase of sales as people are more exposed to clothes imposes indirectly, to the increase of sales.

1.4 Project Objective

The AugDrobe3D project aims at bridging the gap between people and clothing by allowing people assessing and the "real-world" benefits without much hassle and reduces problems of unfitting or unsuitable clothing.

AugDrobe3D will provide a platform of the techno-economical environment and provide detailed roadmaps and guidelines on how to easily use and manage your personalized digital wardrobe, and also provides a channel to try on and even, buy a piece or two that you really like and put it in your collection, virtually and in the real world, of course.

It also aims to create an augmented reality dressing room. This requires real-time tracking of the user pose as well as realistic virtual clothing. For the pose tracking the Kinect is used which gives more complete and accurate tracking of the user pose than the marker based or image feature based tracking which is traditionally used in augmented reality applications. For the clothing we created a set of 3D models which are skeleton-binded which that can be rendered into the scene. The focus of this project is on realistic interaction between the user and the virtual clothing. To achieve this the clothing needs to at least satisfy the following basic conditions:

I. be aligned and fit in correctly with the user silhouette and pose
II. move and fold realistically.
III. be realistically rendered into the environment.

1.5 Stakeholders

Definition

There are many definitions of ‘stakeholder’ and an oft-quoted definition of ‘stakeholder’, taken from a key reference is: ‘A stakeholder in an organization is (by definition) any group or individual who can affect or is affected by the achievement of the organization's objectives.’ A much broader definition, which has also been attributed to Freeman, is that a stakeholder is ‘anything influencing or influenced by’ the firm, but this definition is problematic because it leads to the identification of a very broad set of stakeholders.

Project stakeholder

The definition of project stakeholder is a person group or organization with an interest in a project. Therefore, the stakeholder definition is narrowed down to project stakeholder which focus on the one who is/are interested on our project. Beside that, Information (IS) Researches also taken up the ideas of stakeholders and in software engineering fields, stakeholders have been defined as ‘Stakeholders are people who have a stake or interest in the project’. Therefore, what we concern about stakeholders are the clients or the customers who are interested on our project and the potential users in the future.

Clients

The potential clients and customers can be divided to the two difference fields which are the supplier and the users. The suppliers are mainly the fashion supplier who using this product to advertise their fashion apparels and the users are the group of people who use the product to test, buy and share their fashion apparels.

1.6 Project Scope

Assets (3D Models)

Limited to shirts, jackets, and hats only with few differences sizes available on each model. Each model would be able to changes their view and able to have momentum(small movements like rotating) according to user’s movements.

Kinect

Limited to single user at each time. Therefore, only need to handle one user and only single user is allowed to use at each time.

2 Research

The idea of trying on virtual clothing is not pioneer. With the massively growing interest in augmented reality, applications appeared in which it was possible to try on clothes by overlaying an image of clothing over the image captured by a webcam or digital camera. Like every other technique the virtual fitting room evolved from very simple to more intuitive solutions. The differences in these solutions can be largely reduced to two dimensions: the alignment of the clothing with the user and the realism of the clothing.

2.1 Academic Result

2.1.1 Language

C#

C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative. C# is one of the programming languages designed for the Common Language Infrastructure.

XAML

Extensible Application Markup Language (XAML) is a declarative XML-based language created by Microsoft that is used for initializing structured values and objects. It is available under Microsoft's Open Specification Promise. The acronym originally stood for Extensible Avalon Markup Language - Avalon being the code-name for Windows Presentation Foundation (WPF).

2.1.2 Library

OpenNI with PrimeSense and NITE3

Open Natural Interaction or in short OpenNI is an open source SDK used for the development of 3D sensing middleware libraries and applications. Using just this SDK isn't sufficient for the project, there are two additional middleware library is required for sensors input and gesture which is NiTE and PrimeSense respectively. NiTE is used to to capture the RGB camera input, Depth sensors input and also the IR-sensor input while PrimeSense is used to get the gesture of user. With combination of OpenNI with PrimeSense and NITE3, we can easily detect and manipulate skeleton that are detected as OpenNI provided skeleton detection and skeleton data storing in its SDK.

UnityEngine

Unity Engine is a game engine that developed by Unity Technologies which use scripts to manipulate Game Object (a object that introduced by Unity which is the fundamental object which can attach almost everything in Unity). Unity Engine library has been built based on total 3 difference language which is Boo, JavaScript and C#. This library must be included or used in every script in Unity. Unity Engine library has several important built in classes which are :

I. Transformation classes (Moving or Rotating objects around)
II. Animation System classes(All about animation)
III. Rigid Bodies classes (Collision)
IV. Character controller (FPS or Third person character)

Microsoft .NET Framework

As all scripts in this project are written in C#. Therefore Microsoft .NET Framework library is automatically imported as every scripts created in Unity3D for C# language comes with .NET Framework library. This library include many powerful and useful classes from .NET Framework which can be use in data collection. This library can combine with Unity Engine library to provide more powerful scripts to manipulate Game objects.

2.1.3 Algorithm

The main function of this application is to render the apparels on the user's body. Hence there are few algorithm used to calculate the user's body height and width. Besides, there is also calculations needed to calculate the model with respect to the distance of the users and the Kinect sensors.

2.1.4 Model and Texture

Modeling and texturing an object

3D modeling is the process of developing a mathematical representation of any three-dimensional surface of object (either inanimate or living) via software.

Texture mapping is a method to add details like colors into a 3D model. Therefore, normally a complete 3D model is build from 3D structure first such as coordinate points and then texture mapping other value or details such as image, color, to the 3D model.

Euclidean Transformation

This is basically normal transformation such as rotation, transition but in 3D spaces.

This is necessary when a 3D model is rotating, changing camera view and transforming. For example in this project, while the user is moving front and end hence affect the camera view of the 3D Model. Therefore, euclidean transformation is needed.

2.1.5 Motion Sensing software development kits

We first used the built-in skeletal tracker of Kinect SDK which is also used in Kinect gaming platform. Kinect SDK has some advantages over OpenNI namely per-frame initialization and recovery which provides continuous user tracking with occasional flickers. OpenNI on the other hand, requires a calibration step to recognize the user and initialize the tracking. If at any point user gets out of the scene or obscured by an object for a brief time it is required to do the calibration again with the calibration pose.

Thus, we preferred to use the Microsoft Research Kinect SDK (will be referred as the Kinect SDK in the rest of the paper) due to its robust and practical skeletal tracking algorithm.

But at the later stage while the production moved onto Unity3D apparently Kinect SDK does not support up to version four and above, its feature of only supporting Windows platform makes us shifts to OpenNI, where the whole system begun to develop in. As OpenNI framework is a multi-platform open source API which supports:

I. Hand gestures
II. Body Motion Tracking

2.1.6 Unity3D

Unity3D is a game engine that is cross-platform with built-in IDE and created by Unity Technologies. Unity3D able to support multiple programming language or scripting language such as C# , JavaScript and Boo. Beside that, Unity3D also support multiple platform such as for mobile, iOS , Android, Blackberry 10 and for PC , OS X , Linux , Windows , web browsers , flash and lastly console , PlayStation3 , Xbox 360 , Wii U. The game engine has two different version which is Unity and Unity Pro (Paid Version). Many games are developed based on Unity engine such as the famous Temple Run and Bad Piggies.

2.1.7 WPF

WPF , Windows Presentation Foundation is a computer-software graphical system for building user interface isn Windows-based application which is developed by Microsoft. WPF use XAML to build user interface and use C# to do backend logic. WPF follow Model-View View-Model (MV-VM) structure which directly manipulate front end (user interface) with back end logic directly.

2.1.8 Kinect

Kinect is a motion sensing input device which develop by Microsoft for console (Xbox 360) and PC (Windows). It is the main data input device for this system. Kinect can detect multiple form of input such as depth via depth sensor , voice via microphone and finally real time video (at least 30 frame per second). Therefore, Kinect could detect human skeleton and record user live time movement.Hence, this provide a very interesting and difference real time user interaction compared to traditional application system.

2.1.9 Microsoft Blend

Microsoft Blend is a user interface design software developed by Microsoft for building Windows-based application such as WPF application and Windows-Phone application. Microsoft Blend is easy to build user interface via XAML and most importantly it easy to import 3D model to WPF in XAML form.

2.2 Secondary Research

2.2.1 Findings of Similar Developed System

In the first virtual dressing rooms there was no tracking of the user at all. In this very primitive form of augmented reality only an image of the clothing was displayed on top of the camera image on a fixed position. In order to get the visual experience of wearing the clothing, the user had to align his body with the clothing image himself.

A more appropriate manner of alignment would be to adjust the position, rotation and scale of the clothing to the user. The use of markers in combination with video tracking and image registration techniques made it possible to receive some 3D in- formation from the RGB image using a normal webcam. Position, rotation and scale were adjusted by moving the marker as shown in Figure 2.2.1a.

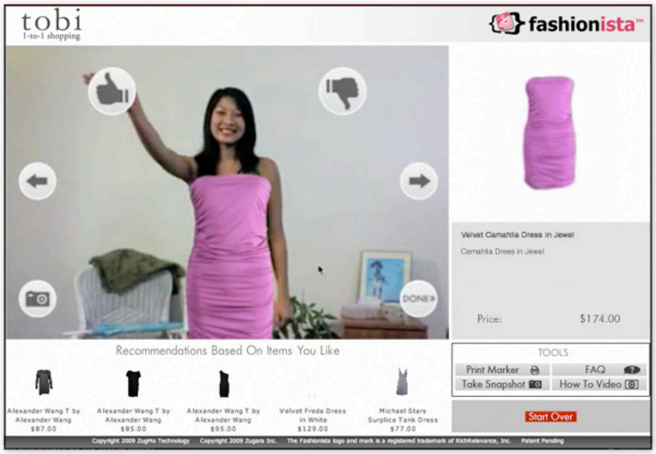


Figure 2.2.1a) Using tracker

The introduction of the Kinect gave relatively easy and cheap access to a depth camera. And with middleware such as the OpenNI framework the user's pose can be tracked quite accurately. A marketing company by the name of FittingReality recently implemented a virtual dressing room which aligns the image of the clothing with the user's body using the Kinect's pose tracking. An example frame from this application is shown in Figure 2.2.1b. This solution is currently the state of the art augmented reality dressing room.

Figure 2.2.1b)

The creators project has been implemented on Topshop in UK with limited function such as browsing its Topshop's catalogs fashion.

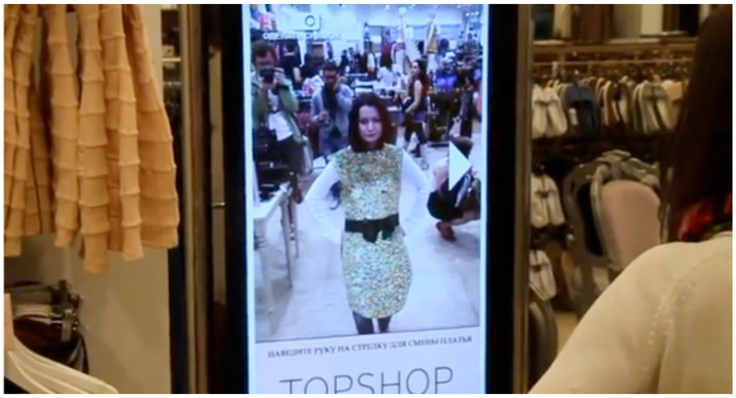


Figure 2.1c) Topshop virtual fitting in the UK

3 System Analysis

3.1 Analysis

3.1.1 Functionality

1. Our cursor still requires tweaking to get it smoother. Some users accidentally selected the wrong item, and others weren't able to move the cursor the way they wanted to.
2. Our measurements are somewhat accurate for men, if they are standing exactly 6 feet and 10 inches away, but women have a much bigger variety of body types.
3. We have no error recovery when we can't see joints - we just blindly proceed and cause C# errors.
4. Recalibrate button should take users back to loading page and should always be visible.
5. Recalibrate button is currently too far down at the bottom of the page.

3.1.2 Communication with the user

1. Everything happens so fast, it confuses the users.
2. Text on the screen can be hard to read from far away.
3. Users doesn't really understand what is going on on the screen and only focus on the interactive area.
4. When users "recalibrate", they want to reselect their clothing.
5. Voice commands or pictures would help communicate instructions.

3.1.3 Shortcomings or Caveats of testing:

1. Few test subjects is hardly representative of everyone who goes online shopping, and many were shaped pretty similarly.

2. We happened to use only male test subjects as the minimum viable product does not provide support for female test subjects as female has very much varying body sizes.
3. We did most of the developing and testing on a Mac, which tends to be slower and less responsive than the PCs. For instance, the cursor is not quite as problematic on a PC. For the cursor especially, we need to figure out how much of the unresponsiveness is due to the Mac and how much is due to our app not functioning correctly.
4. Some of the tests were in public but we envisioned the system being used on a big screen in people's homes. In the real world, there won't be someone watching you get your measurements -- that made some users a little uncomfortable. This was especially noticeable with our female test subjects.

3.2 Implications

User testing helped us find a number of issues with our application. Overall, we need to make sure to keep users informed of how to actually use it, slow everything down, improve cursor functionality.

We discovered that people quickly figured out what our app was doing but we didn't make the interface as intuitive as possible. Obviously there are some user expectations that we won't be able to meet in a single quarter, but we will focus on making the user interaction slower and harder to get wrong. Given the technical limitations of KinectJS we seem to have pushed the Kinect to its limits in terms of fine-grained measurements. As we move forward, we will focus less on increasing the accuracy of our measurements and more on improving our ability to guide the user through our process.

3.2.1 Cursor

1. Fix hover issues that the cursor is having
2. We made this a little better mid-testing by increasing the size of the cursor and restricting user input to only the right hand. The cursor is also animated to indicate how long the user should hover over an element to select it.
3. Bind cursor to width and height of screen so it doesn't travel outside the page
4. Check cursor/object overlap at the center rather than at the edges

3.2.2 Measurements

1. Currently, the application underestimates for male users. We are thinking of improving accuracy by asking the user to stand sideways and placing their hands on either side of their body. We will use wrist to wrist distance as an indicator of width.

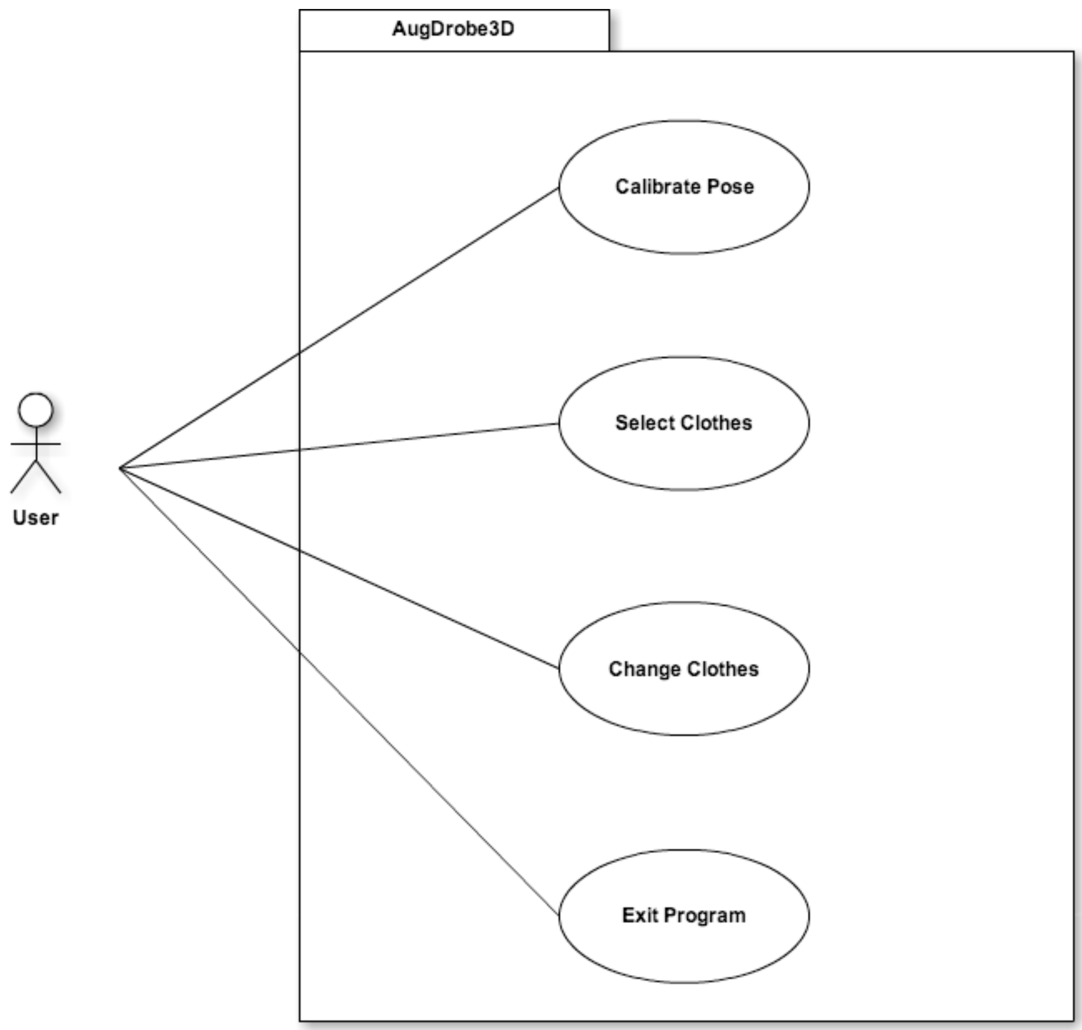
3.2.3 Measurement-taking process

1. Since the skeleton causes people to move, what if we don't show the skeleton, just put an image (3D mesh of a person in the correct position) there? Or we could overlay the skeleton on a reference image so users will know how to position their bodies. We will want to give users this visual feedback before the scanning process begins, so users have time to get into the proper position.
2. Slow it down (add a countdown before you have to stand still, take more samples, make everything take longer)
3. Add error recovery when we can't see joints, instead of freezing the program without telling the user what's wrong.

4 Requirement Specification

4.1 Functional Requirement

4.1.1 Use case diagram



4.1.2 Use case specification

Use Case ID	UC01
Use Case Name	Select Clothes
Description	The user can select the apparel by moving the hand cursor.
Author(s)	Choo Kok Hong
Last Updated By	Choo Kok Hong
Date Created	22 April 2013
Date Last Updated	24 April 2013
Actors	Users
Location(s)	
Status	Pathway Defined
Priority	2
Assumption(s)	Kinect is connected
Pre-Conditions	✓ User must be present at the camera frame
Post-Conditions	✓ User must be worn with virtual clothes model

Use Case ID	UC02
Use Case Name	Remove Cloths
Description	The user can remove the apparel that he or she is applied.
Author(s)	Choo Kok Hong
Last Updated By	Choo Kok Hong
Date Created	22 April 2013
Date Last Updated	24 April 2013
Actors	Users
Location(s)	
Status	Pathway Defined
Priority	2
Assumption(s)	Kinect is connected
Pre-Conditions	✓ User must be present at the camera frame ✓ User must be applied apparel
Post-Conditions	

Use Case ID	UC03
Use Case Name	Recalibrate Skeleton
Description	The user can recalibrate the skeleton which can provide a more accurate stimulation.
Author(s)	Choo Kok Hong
Last Updated By	Choo Kok Hong
Date Created	22 April 2013
Date Last Updated	24 April 2013
Actors	Users
Location(s)	
Status	Pathway Defined
Priority	2
Assumption(s)	Kinect is connected
Pre-Conditions	
Post-Conditions	

Use Case ID	UC04
Use Case Name	Exit Program
Description	User click this button to exit the program.
Author(s)	Choo Kok Hong
Last Updated By	Choo Kok Hong
Date Created	22 April 2013
Date Last Updated	24 April 2013
Actors	Users
Location(s)	
Status	Pathway Defined
Priority	2
Assumption(s)	Kinect is connected
Pre-Conditions	✓ Application is started
Post-Conditions	

4.2 Technical requirements

Operating System: Windows 7 or Mac OS X or Linux Ubuntu 12.04 and above

CPU: Intel core i5 2550 and above

GPU: Nvidia GTX 360 and above

Memory: RAM 2GB and above

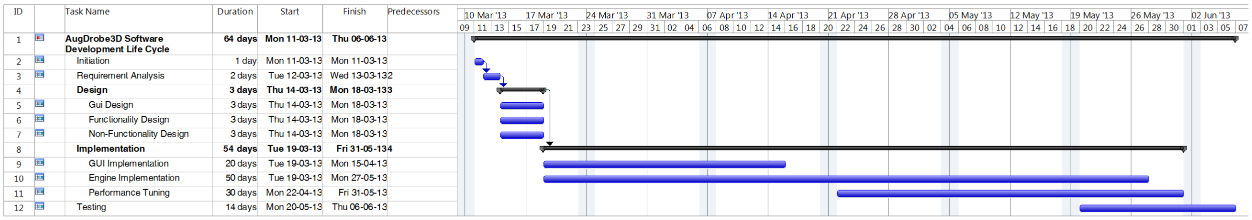
Hardware: Kinect Sensor

Software: OpenNI v1.5.4.0, PrimeSense v5.1.2.1 and NiTE v1.5.2.21 installed

Above requirements are the suggested requirement to run the project at optimum performance. Intel core i5 2550 is suggested as there are a lot of calculations that are running at the rate of up to 30 times per seconds, hence, a powerful processor is required. Nvidia GTX 630 is suggested as the input of the Kinect Sensor have a resolution of 640 x 480 @ 30 fps which need certain graphics processing power to render the image input stream. 4 GB of RAM is required as there are some memory occupied by the OS and also the other program that is running, hence, 4GB is recommended. Kinect Sensor is the main input source of the project, so it is needed and the program won't be able to function without a Kinect Sensor. Alongside with the Kinect Sensor, OpenNI version 1.5.4.0, PrimeSense version 5.1.2.1 and NiTE version 1.5.2.21 is required in order to use Kinect Sensor properly. **

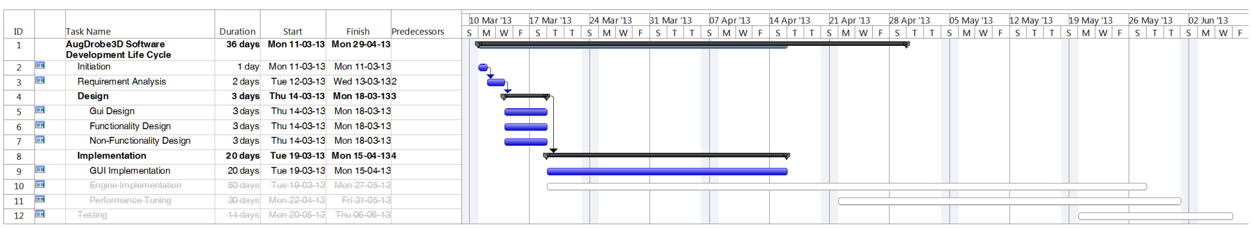
4.3 Gantt Chart

4.3.1 Original Project Gantt Chart



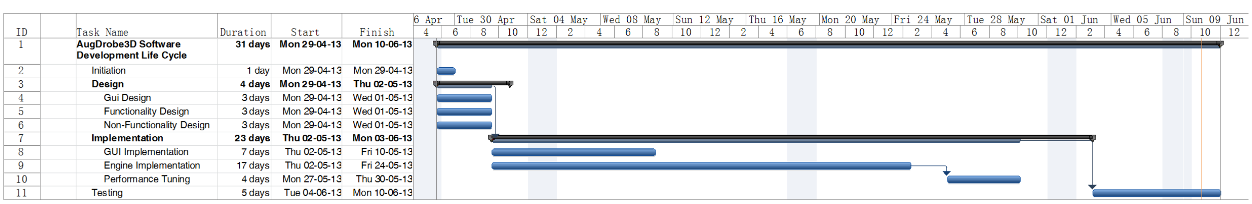
ID		Task Name	Duration	Start	Finish	Predecessors
1		**AugDrobe3D Software Development Life Cycle**	**64 days**	**Mon 11-03-13**	**Thu 06-06-13**	
2		Initiation	1 day	Mon 11-03-13	Mon 11-03-13	
3		Requirement Analysis	2 days	Tue 12-03-13	Wed 13-03-13	2
4		**Design**	**3 days**	**Thu 14-03-13**	**Mon 18-03-13**	**3**
5		Gui Design	3 days	Thu 14-03-13	Mon 18-03-13	
6		Functionality Design	3 days	Thu 14-03-13	Mon 18-03-13	
7		Non-Functionality Design	3 days	Thu 14-03-13	Mon 18-03-13	
8		**Implementation**	**54 days**	**Tue 19-03-13**	**Fri 31-05-13**	**4**
9		GUI Implementation	20 days	Tue 19-03-13	Mon 15-04-13	
10		Engine Implementation	50 days	Tue 19-03-13	Mon 27-05-13	
11		Performance Tuning	30 days	Mon 22-04-13	Fri 31-05-13	
12		Testing	14 days	Mon 20-05-13	Thu 06-06-13	

4.3.2 Original Gantt Chart of Original Project before change to another approach



ID		Task Name	Duration	Start	Finish	Predecessors
1		**AugDrobe3D Software Development Life Cycle**	**36 days**	**Mon 11-03-13**	**Mon 29-04-13**	
2		Initiation	1 day	Mon 11-03-13	Mon 11-03-13	
3		Requirement Analysis	2 days	Tue 12-03-13	Wed 13-03-13	2
4		**Design**	**3 days**	**Thu 14-03-13**	**Mon 18-03-13**	**3**
5		Gui Design	3 days	Thu 14-03-13	Mon 18-03-13	
6		Functionality Design	3 days	Thu 14-03-13	Mon 18-03-13	
7		Non-Functionality Design	3 days	Thu 14-03-13	Mon 18-03-13	
8		**Implementation**	**20 days**	**Tue 19-03-13**	**Mon 15-04-13**	**4**
9		GUI Implementation	20 days	Tue 19-03-13	Mon 15-04-13	
10		~~Engine Implementation~~	~~50 days~~	~~Tue 19-03-13~~	~~Mon 27-05-13~~	
11		~~Performance Tuning~~	~~30 days~~	~~Mon 22-04-13~~	~~Fri 31-05-13~~	
12		~~Testing~~	~~14 days~~	~~Mon 20-05-13~~	~~Thu 06-06-13~~	

4.3.3 Gantt Chart for new approach



ID		Task Name	Duration	Start	Finish
1		**AugDrobe3D Software Development Life Cycle**	**31 days**	**Mon 29-04-13**	**Mon 10-06-13**
2		Initiation	1 day	Mon 29-04-13	Mon 29-04-13
3		**Design**	**4 days**	**Mon 29-04-13**	**Thu 02-05-13**
4		Gui Design	3 days	Mon 29-04-13	Wed 01-05-13
5		Functionality Design	3 days	Mon 29-04-13	Wed 01-05-13
6		Non-Functionality Design	3 days	Mon 29-04-13	Wed 01-05-13
7		**Implementation**	**23 days**	**Thu 02-05-13**	**Mon 03-06-13**
8		GUI Implementation	7 days	Thu 02-05-13	Fri 10-05-13
9		Engine Implementation	17 days	Thu 02-05-13	Fri 24-05-13
10		Performance Tuning	4 days	Mon 27-05-13	Thu 30-05-13
11		Testing	5 days	Tue 04-06-13	Mon 10-06-13

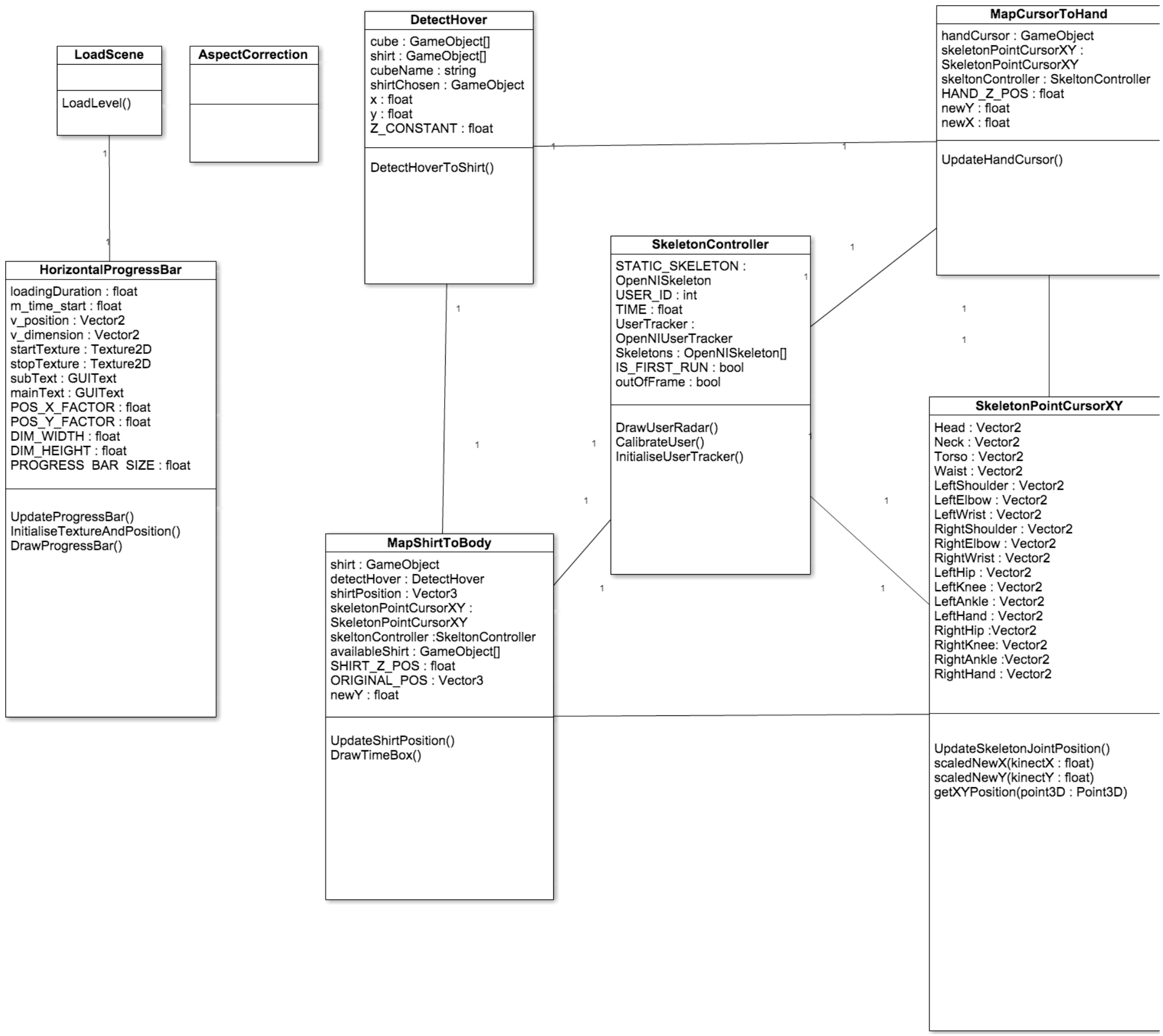
5 System Design

5.1 Human Computer Interaction (HCI)

For any augmented reality application it is important to have realistic interaction between the user and virtual objects, in this case the virtual clothing. Interaction with the clothing comes in two forms, firstly having the cloth move with the user as if he/she is wearing them, and secondly direct physical interaction between the clothing and the user body.

These two forms of interaction correspond to the two types of clothing simulation: skinning the cloth to the user skeleton allows for realistic movement, while the interactive cloth includes a physics simulation that allows for correct interaction between the cloth and physical objects in the scene, such as the user body. A combination between the two approaches could combine the best attributes of both. Such an approach would be to add collision detection to the vertices in the clothing mesh, but restricting the resulting transformation to the volume defined by the skinned cloth coefficients.

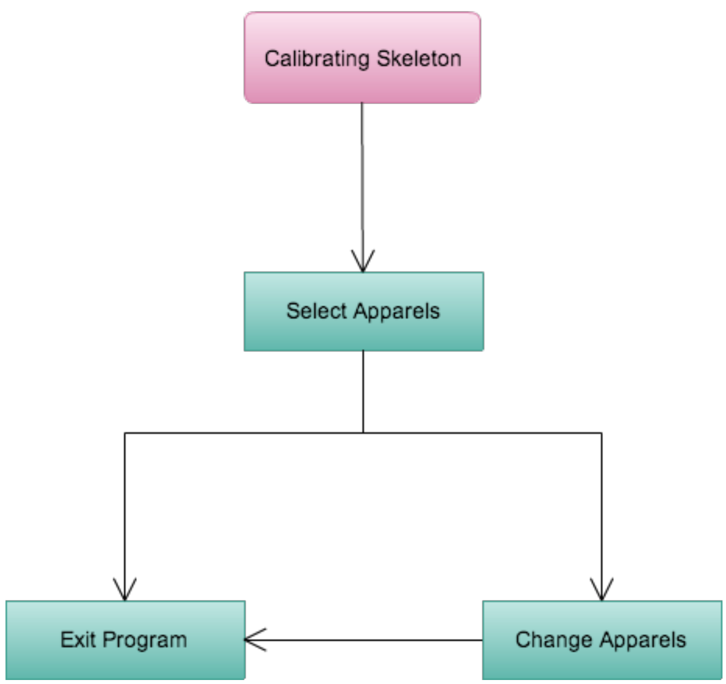
5.2 Class Diagram



5.3 Database Design

The only database of this project is the assets which is the apparels. The file format (.fbx) we are using is not a format that is created by us but a format that is developed by Kaydara and now owned by AutoDesk. This format is selected over other format is because of few of its advantages. Among them, FBX format can have geometry animations data, bone data and bone animation data. These three are the very important factor for our project as the whole project is greatly rely on the bone structure of the 3D model. The animations follow what is in the file, so that it won't transform abnormally.

5.4 Flowchart



6 Development

6.1 Pre-processing

6.1.1 User extraction

Extraction of user allows us to create an augmented reality environment by isolating the user area from the video stream and superimposing it onto a virtual environment in the user interface.

The OpenNI provides the depth image and the user ID. When the device is working, depth image is segmented in order to separate background from the user.

6.1.2 Tracking

The tracking algorithm is created using a large training data set with over 500k frames distributed over many video arrays. The data set have been generated synthetically by rendering depth images of common poses of actions which are frequently performed in video games such as dancing, running or kicking, along with their labeled ground truth pairs. In order eliminate same poses in the data set; a small offset of 5cm is used to create a subset of poses that all poses are at least as distant as the offset to each other. Pixel depths are then normalized before training so that they become translation independent. After that, training is done to form a decision forest of depth 20.

6.1.3 Model Positioning and Rotation

The skeletal tracker returns the 3D coordinates 20 body joints in terms of pixel locations for x and y coordinates and meters for the depth. Figure 4 illustrates the 9 of the 20 joints which are used to locate the parts of the 2D cloth model.

One may notice flickers and vibrations on the joints due to the frame based recognition approach. This problem is partially solved by adjusting the smoothing parameters of the skeleton engine of the OpenNI engine.

The rotation angles of the parts of the model are defined as the angles of the main axes of the body and arms.

Letting $x = x_{joint1} - x_{joint2}$ and $y = y_{joint1} - y_{joint2}$.

$$\theta = atan2\left(y, x\right) = 2\arctan\left(\frac{\sqrt{x^2+y^2}-x}{y}\right)$$

atan2 is a arctangent formula which is defined in the interval 0 to π.

Here the main body axis is defined as the line between the shoulder joint and the hip joint, and the axes of the body are defined such that they are the lines between corresponding shoulders and elbows.

For each model part a skeleton point is defined as the center of transformation which is set to the middle of the corresponding line of axis.

6.1.4 Point mapping

As the captured skeletal motion do not reflect what is in the real world, post-calculation has to be done for accurate positioning.

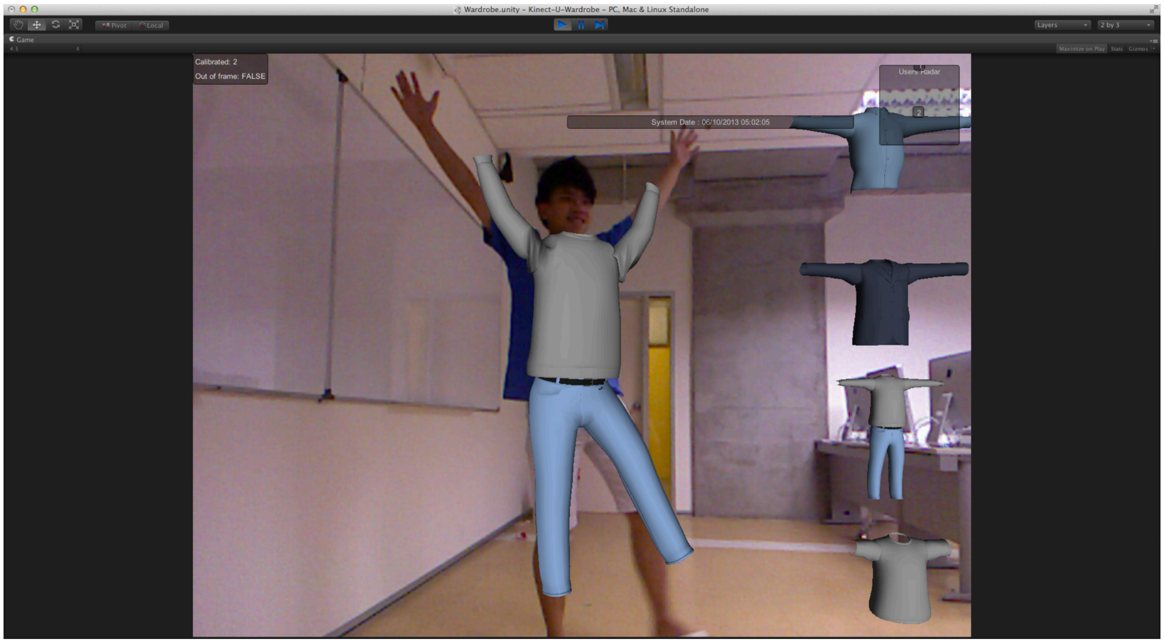
To map a point (x, y, z) from a world of width w, height h and depth p to (p, q, r) from a world of width w', height h' and depth p',

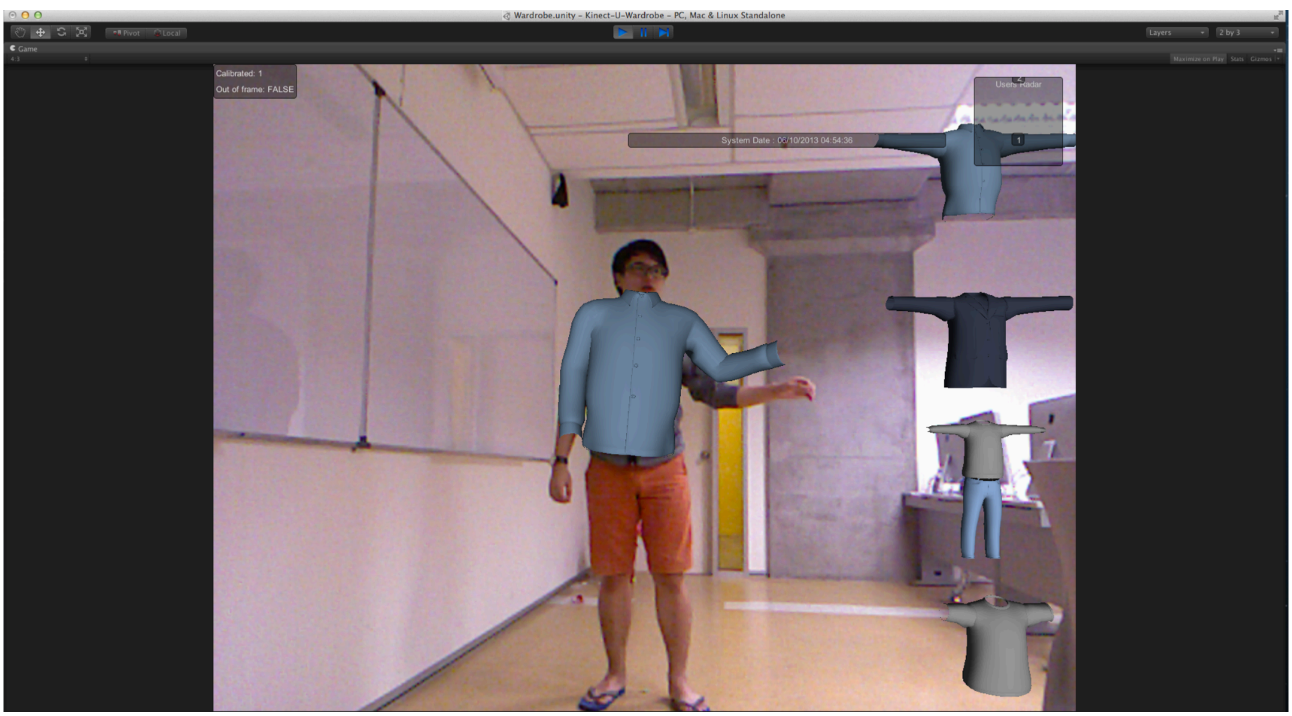
$$p = \frac{xw'}{w}, q = \frac{yw'}{w}, r = \frac{zd'}{d}$$

for where the origin of world is at the centre of mass.

6.1.5 Model Scaling

To employ the distance between the joints as a scaling factor, a naïve approach is did as it is convenient to accommodate the height and weight related variation of users. However the distance between the joints is not sufficiently precise to simulate the distance from the sensor. Hence, a hybrid approached is adopted in order to perform shape and distance based scaling with better precision.


Wardrobe.unity - Kinect-U-Wardrobe - PC, Mac & Linux Standalone
Calibrated: 2
Out of frame: FALSE


Wardrobe.unity - Kinect-U-Wardrobe - PC, Mac & Linux Standalone
Calibrated: 1
Out of frame: FALSE

7 Testing and Evaluation

7.1 Unit Testing

For our unit test, basically we need to test all the basic functionality of this system. In order to achieve our goal and objective, this system must meet these few requirements as below.

I. Shirt object should be able correctly mapped to user's body

This function is very important as this is the minimum requirement of this system which is the shirt must be wore on user's body. As there is a difference in resolution of input from Kinect and resolution of Unity3D. Therefore, mapping and scaling algorithm need to be applied in order for this function to work. Basically, this function is should be able to map the shirt object into user's body once user is detected and calibrated which meant user skeleton is detected and calibrated by Kinect input.

II. Shirt object should be able to follow user's movements

This function is similar to function one but it is a improved version of function one. This function objective is to make sure the shirt object able to follow user's movements like turing right or left , bending forward or backward , squad down , jump and any hand and leg movements. As user is asked to stand in a fixed point, therefore user moving forward , backward , left or right is not included and tested.

III. User should be able to use hand to control cursor

This function objective is user able to use hand to control cursor which this system will provide a hand cursor texture and let user to control this hand cursor texture. This function only available when user is calibrated correctly. The direction, consistency, movement speed and user friendliness of hand cursor is considered and tested.

IV. User should be able to change shirt object

This function is the improved version of function three which user able to change shirt object by hovering the hand cursor to shirt object that are present on the application. Therefore, user must be able to change cloths and the shirt that is currently wore by user must be changed.

7.2 Methods

We tested our system on a range of users. However , the gender of the users must be male as our target is only aim to be male in this stage. The user group consists of 3 people which have a difference in height and size. In order to make sure the testing is accurate and precise, the users are asked to stand a fixed position or fix a constant distance from Kinect. Below are the methods to test the following functions.

I. Shirt object should be able correctly mapped to user's body

As user are stand in a fixed position , therefore user are asked to be follow the instruction on the screen which is calibrate with the standard calibration post.

II. Shirt object should be able to follow user's movements

The pre-condition now is that user has shirt object mapped on his body now. After that, user are asked to do a sequence of movements as below:

1. Jump
2. Squad Down
3. Continuously jump twice
4. Jump once then directly squad down

III. User should be able to use hand to control cursor

The condition is that the user has been calibrated correctly. User is asked to moving his hand around and try to control the hand cursor. Beside that, user is also asked to move only left hand , move only right hand and move both hand together. Furthermore, user is asked to moving hand quickly and slowly in order to know the difference between them.

IV. User should be able to change shirt object

User is asked to control the hand cursor to select the shirt object that he wants. Furthermore, user is asked to hover to all available shirt object and observe the whether the output (whether shirt is changed) has any differences.

Lastly, we have asked multiple questions for the testers:

1. Does the user interface user friendly?
2. Does the instruction given is clear and easy to understand?
3. Does the hand cursor easy to control?
4. Does the cloths map to your body correctly?
5. Does the cloths follow your movements correctly?

7.3 Test Results

As we have total three tester which all three of them has a difference in term of height and size, therefore we have all the feedbacks and comments collected.

I. Shirt object should be able correctly mapped to user's body

In this stages, all the shirt object size is fixed to "M" size which will match most body's sizes of people in Asian. However, one of our tester body's size is bigger than "M". Therefore, the shirt does not really fix the size of the tester's body. About the consideration of height, it shows an acceptable result for mapping to all user's body but however there still have a gap between testers which has significant difference in height.

II. Shirt object should be able to follow user's movements

All the testers are asked to follow the sequence of movements that we designed and the results are surprisingly great. All the movements like jump and squad down followed user's movements precisely and accurately. The only deficiency is that the movements of shirt and user has a sightly synchronization delay which is not significant observed.

III. User should be able to use hand to control cursor

For the test result of this function, all the users are able to control the hand cursor by moving their right hand around but however the hand cursor is very flickering as the position of hand cursor is based on the detection of right hand joint. The detection of the joint is refreshed at least 60 times per second and the detection by Kinect is only approximation only. Therefore, it will have about small amount of difference of position point per second and this is the main reason for the flickering of hand cursor. However, the result is acceptable as all user able to move the cursor to anywhere they wanted to moved.

IV. User should be able to change shirt object

All of the testers able to change the shirt object by hovering the hand cursor to the shirt object that want to test to wear on. The shirt changing process is smooth and fast. Therefore, overall this function result is satisfied.

Lastly we have asked several question which stated in above and all of the testers have give their feedbacks and comments for this system based on the question we have asked.

1. Does the user interface user friendly?

All of the testers say the user interface is very user friendly this system is easy to use.

2. Does the instruction given is clear and easy to understand?

One of the tester says that the instruction box are a little bit small but the instruction given is easy to understand. However, the other testers give positive feedbacks about the instruction.

3. Does the hand cursor easy to control?

Almost all of the testers states that it is quite hard to control the hand cursor as it is flickering but they still able to control it.

4. Does the cloths map to your body correctly?

Almost all of the users satisfy the mapping of shirt object except for the tester that has a bigger body size as we only has “M” sizes shirt object at this stages.

5. Does the cloths follow your movements correctly?

All of the testers give positive feedback about this part.

8 Conclusion

8.1 Success Criteria

This report has presented an augmented reality application in which users can select and try on virtual clothes. These clothes are rendered on a screen over the image of the user and the lighting of the rendered clothing is adapted to match the light in- tensity of the user's environment. The main features of this application, corresponding to the goals in chapter 1.4, are as follows:

1) The clothing is correctly aligned with the user's position and movement by skinning the clothing models to a skeleton rig which is con- trolled by the Kinect sensor's skeleton tracking.
2) The clothing moves and folds realistically by applying Unity's clothing simulation to the clothing models.
3) The clothing is realistically rendered by adapt- ing the light intensity in the virtual scene to match the user's environment and by masking out the parts of the rendered clothing which would be occluded by the user.

The presented application is an improvement over similar existing augmented reality applications in that it offers both full user pose tracking using the Kinect, as well as 3-dimensional clothing mod- els with cloth physics simulation such that it can be viewed from any angle, smoothly moves and rotates with the user, and reacts similar to real clothing.

As a technology demo, the presented application shows how accurate augmented reality interaction with virtual objects can be realized using a depth camera with pose tracking. Considering that such sensors are now being released as consumer products we expect much future work in augmented reality will take advantage of these technologies.

8.2 Limitation

Since other objects from the user's environment have no mesh representation they are not used in this masking process and can not occlude the clothing. As a result the clothing might be projected over an object which occludes the user and should thus occlude the clothing.

Size estimation does work well under a simple algorithm, it will be much harder to estimate the true girth not knowing what clothing the user is wearing, and how much distance there is between the clothes and the user body. Accuracy might increase by using a large dataset of trainings data and linear regression methods to estimate the girth.

8.3 Future implementation

A smoothing algorithm should be developed to work along with the application as it currently would flicker when the detection of the joints is not precise and steady. This would greatly enhance user experience as people tend to get panic over flickering items especially applications that needs interaction.

Women clothing is a wide and open market waiting for exposure. As current product did not cater for women body which varies in body size, types and measurements, as well as for children, which opens us to the child clothing opportunity.

As discussed in the test results, there are more feature that still need tweaking and improvement and as the project moves ahead, better work is expected with better performance.

References

1. The Creators Project (2011). The Kinect Invades Dressing Rooms [Finally] | The Creators Project. [online] Retrieved from: http://thecreatorsproject.vice.com/blog/the-kinect-invades-dressing-rooms-finally [Accessed: 23 Apr 2013].
2. YouTube (2000). Kinect Virtual Fashion, the Future of Shopping at Home, Retail, and on Smartphone. [online] Retrieved from: http://www.youtube.com/watch?v=s0Fn6PyfJ0I [Accessed: 23 Apr 2013].
3. YouTube (2000). Try On Clothes VIRTUALLY With Xbox 360 Kinect Using Swivel By Face Cake (CES 2012). [online] Retrieved from: http://www.youtube.com/watch?v=WwUa14mNNhY [Accessed: 23 Apr 2013].
4. Josuttis, N. M. (2012). The C++ standard library: A tutorial and reference. Upper Saddle River, NJ: AddisonWesley.
5. Meyers, S. (2005). Effective C++: 55 specific ways to improve your programs and designs. Upper Saddle River, NJ: Addison-Wesley.
6. Stroustrup, B. (2000). The C++ programming language. Boston: Addison-Wesley.
7. Schildt, H. (2010). C# 4.0: The complete reference. New York: McGraw-Hill.
8. Dalal, M., & Ghoda, A. (2011). Xaml developer reference. Sebastopol, Calif: O'Reilly Media
9. Microsoft.com (n.d.). Kinect Sensor Setup, Requirements, Support | Kinect for Windows. [online] Retrieved from: http://www.microsoft.com/en-us/kinectforwindows/purchase/sensor_setup.aspx [Accessed: 23 Apr 2013].
10. Msdn.microsoft.com (n.d.). Kinect Fusion. [online] Retrieved from: http://msdn.microsoft.com/en-us/library/dn188670.aspx [Accessed: 23 Apr 2013].

11.Microsoft.com (n.d.). Kinect for Windows | Voice, Movement & Gesture Recognition Technology. [online] Retrieved from: http://www.microsoft.com/en-us/kinectforwindows/ [Accessed: 23 Apr 2013].

12.Pterneas, V. (2011). How-to: Successfully Install Kinect on Windows (OpenNI and NITE) - CodeProject. [online] Retrieved from: http://www.codeproject.com/Articles/148251/How-to-Successfully-Install-Kinect-on-Windows-Open [Accessed: 23 Apr 2013].

13.Podila, P., & Hoffman, K. (2010). WPF control development unleashed: Building advanced user experiences. Indianapolis, Ind: Sams.

14.Yosifovich, P. (2012). Windows Presentation Foundation 4.5 cookbook. Birmingham, UK: Packt Pub.

15.Rastertek.com (2011). DirectX 10 Tutorials. [online] Retrieved from: http://www.rastertek.com/tutdx10.html [Accessed: 23 Apr 2013].

16.Taking Initiative (2011). DirectX10 Tutorials. [online] Retrieved from: http://takinginitiative.net/directx10-tutorials/ [Accessed: 23 Apr 2013].

17.Rastertek.com (n.d.). Tutorial 4: Buffers, Shaders, and HLSL. [online] Retrieved from: http://www.rastertek.com/dx10tut04.html [Accessed: 23 Apr 2013].

18.Microsoft.com (n.d.). Kinect Tutorials | Kinect for Windows. [online] Retrieved from: http://www.microsoft.com/en-us/kinectforwindows/develop/tutorials.aspx [Accessed: 23 Apr 2013].

19.Creativecrash.com (2006). Maya Tutorials, Maya 3D Tutorials, Tutorials for Maya. [online] Retrieved from: http://www.creativecrash.com/maya/tutorials/ [Accessed: 24 Apr 2013].

20.Msdn.microsoft.com (n.d.). Learn Microsoft Expression Blend: videos, tutorials, whitepapers. [online] Retrieved from:http://msdn.microsoft.com/en-us/expression/cc197141.aspx[Accessed: 23 Apr 2013].

21.Kinecthacks.com (2013). Kinect Hacks - Supporting the Kinect Hacking news and community. [online] Retrieved from: http://www.kinecthacks.com [Accessed: 9 Jun 2013].

22.Unity Documentation (n.d.). Unity Script Reference – Overview: The most important classes. [online] Retrieved from: http://docs.unity3d.com/Documentation/ScriptReference/index.The_most_important_classes.html [Accessed: 9 Jun 2013].

23.Unity Documentation (2013). Unity - Using Scripts. [online] Retrieved from: http://docs.unity3d.com/Documentation/Manual/Scripting.html [Accessed: 9 Jun 2013].