AugDrobe3D - Augmented Reality Virtual Fitting Room with Kinect and Unity3D

Lai Li Hao

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 Kinect 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 intensity 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 statically typed, free-form, multi-paradigm, compiled, general-purpose programming language. It is regarded as an intermediate-level language, as it comprises both high-level and low-level language features. Developed by Bjarne Stroustrup starting in 1979 at Bell Labs, C++ was originally named C with Classes, adding object oriented features, such as classes, and other enhancements to the C programming language. The language was renamed C++ in 1983, as a pun involving the increment operator.

The need of C++ is evident as it is needed to build DirectX application where we would need to render the various model of clothing.

2.1.2 Library

DirectX10

Microsoft DirectX is a collection of application programming interfaces (APIs) for handling tasks related to multimedia, especially game programming and video, on Microsoft platforms. The DirectX software development kit (SDK) consists of runtime libraries in redistributable binary form, along with accompanying documentation and headers for use in coding. DirectX 10 ships with and is only available with Windows Vista and later; previous versions of Windows such as Windows XP are not able to run DirectX 10-exclusive applications. Programs that are run on a Windows XP system with DirectX 10 hardware simply resort to DirectX 9.0c code path.

DirectX10 is the main API in our program, it is using to import and render costume's 3D models to the program. It is also used to manipulate and control the 3D models to follow the skeleton's movement which detected by Kinect. The reason we are using DirectX instead of OpenGL because our program target operating system is Windows and the Kinect is working more stable on the Windows Operating System, so DirectX will be our first choice to use.

DirectX 10 is the main API in the program, it is using to import and render costume's 3D models to the program. It is also used to manipulate and control the 3D models to follow the skeleton's movement which detected by Kinect. The reason using DirectX instead of OpenGL because the program target operating system is Windows and the Kinect is working more stable on the Windows Operating System, so DirectX will be the first choice to use. The version 10 is selected because the difference between 10 and 11 is pretty small, since some hardware on the market currently still cannot fully support version 11 so the version 10 still is the best choice for most users.

Direct3D HLSL (High Level Shader Language)

Direct3D is a low-level API that you can use to draw triangles, lines, or points per frame, or to start highly parallel operations on the GPU. Direct3D is available for Microsoft Windows operating systems (Windows 95 and above), and for other platforms through the open source software Wine. It is the base for the graphics API on the Xbox and Xbox 360 console systems. Direct3D is used to render three dimensional graphics in applications where performance is important, such as games. Direct3D also allows applications to run fullscreen instead of embedded in a window, though they can still run in a window if programmed for that feature. Direct3D uses hardware acceleration if it is available on the graphics card, allowing for hardware acceleration of the entire 3D rendering pipeline or even only partial acceleration. Direct3D exposes the advanced graphics capabilities of 3D graphics hardware.

The HLSL (High Level Shading Language) is a proprietary shading language developed by Microsoft for use with the Microsoft Direct3D API. Using HLSL, we can create C like programmable shaders for the Direct3D pipeline. In our program, we will use HLSL to code vertex and pixel shader.

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

UV Mapping Basics

This process projects a texture map onto a 3D object. The letters "U" and "V" denote the axes of the 2D texture because "X", "Y" and "Z" are already used to denote the axes of the 3D object in model space.

UV texturing permits polygons that make up a 3D object to be painted with color from an image. The image is called a UV texture map,but it's just an ordinary image. The UV mapping process involves assigning pixels in the image to surface mappings on the polygon, usually done by "programmatically" copying a triangle shaped piece of the image map and pasting it onto a triangle on the object. UV is the alternative to XY, it only maps into a texture space rather than into the geometric space of the object. But the rendering computation uses the UV texture coordinates to determine how to paint the three dimensional surface.

As UV mapping can be generated automatically with 3D model software. Hence, it is included in our model assets.

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 4 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 Autodesk 3Ds Max

Autodesk 3Ds Max is a famous 3D modeling and rendering software especially for graphic artists and video game developers to create better 3D animation, models and image in less time. It was developed by Autodesk Media and Entertainment. In addition to its modeling and animation tools, it also features shaders, dynamic simulation, particle systems and normal creation. Not only that, Autodesk 3Ds Max allow user to install addition plugin that function not provide by Autodesk 3Ds Max or enhance the basic function. Autodesk 3Ds Max is a user friendly application which easy to use for novice and its support most of the common 3D format in the market, some of the common0 format is created by Autodesk so it can ensure the application can present the 3D model correctly. Autodesk 3Ds Max is not for creating 3D models or 3D animation in this project but it helps the entire project to enhance the original 3D models with adding skeleton. It makes the 3D model easier to render correctly on the project and can easy to follow the input from Kinect.

2.1.8 FBX Model

FBX is a file format (.fbx) developed by Kaydara and now owned by Autodesk. It is used to provide interoperability between digital content applications. The content included in FBX and the most common 3D format OBJ (.obj) which developed by Wavefront Technologies are different as the table below, the main reason we choose FBX instead of OBJ is because of the content of Geometry and Animation, because the project is highly depend on the geometry animation and bone animation. The 3D models will has a skeleton which will map to the Kinect input and the 3D models will auto do the animation for the vertex transform.

	Geometry and Animations			
	Geometry	Geometry Animation (RTS)	Bone-skinned Data	Bone Animation (RTS)
FBX	O	O	O	O
OBJ	O	X	X	X

	Materials and UV Settings									
	Multi-material	UV	UV Tiling	UV Animation	Reflection	Refraction	Self-illumination	Specular	Glossiness	Opacity
FBX	O	O	O	X	X	X	O	O	O	O
OBJ	O	O	X	X	X	X	X	X	O	O

	Colors		
	Diffuse Color	Ambient Color	Specular Color
FBX	O	O	X
OBJ	O	O	O

T = Texture, S = Strength.

	Textures and Strength													
	Diffuse		Opacity		Bump/ Normal		Specular		Glow		Reflection		Blend	
	T	S	T	S	T	S	T	S	T	S	T	S	T	S
FBX	O	O	O	O	O	O	O	O	X	X	O	O	X	X
OBJ	O	X	O	X	O	X	O	X	X	X	X	X	X	X

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.

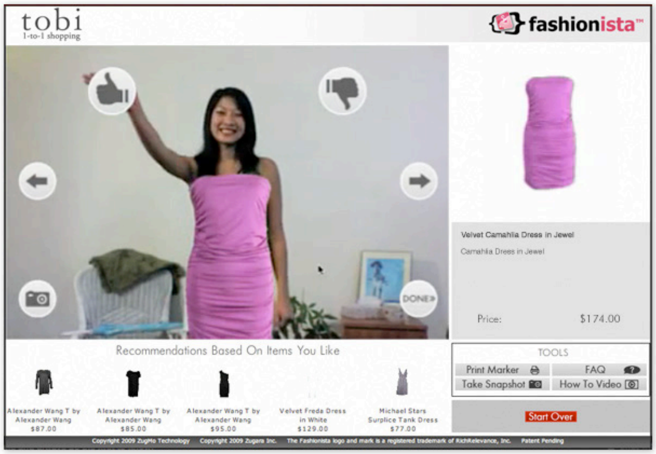


Figure 2.2.1a) Using tracker

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.

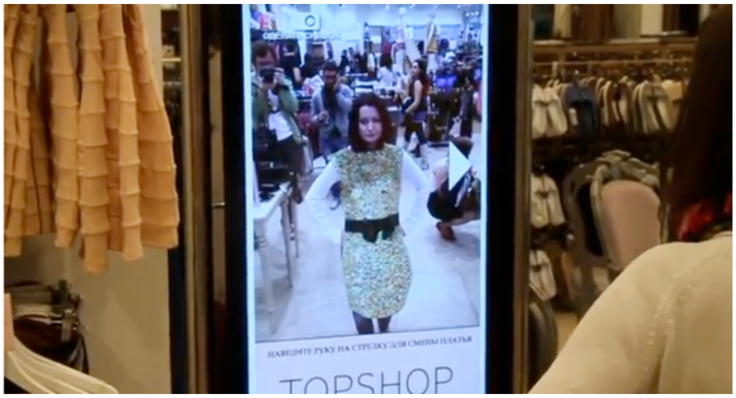


Figure 2.1c) Topshop virtual fitting in the UK

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.

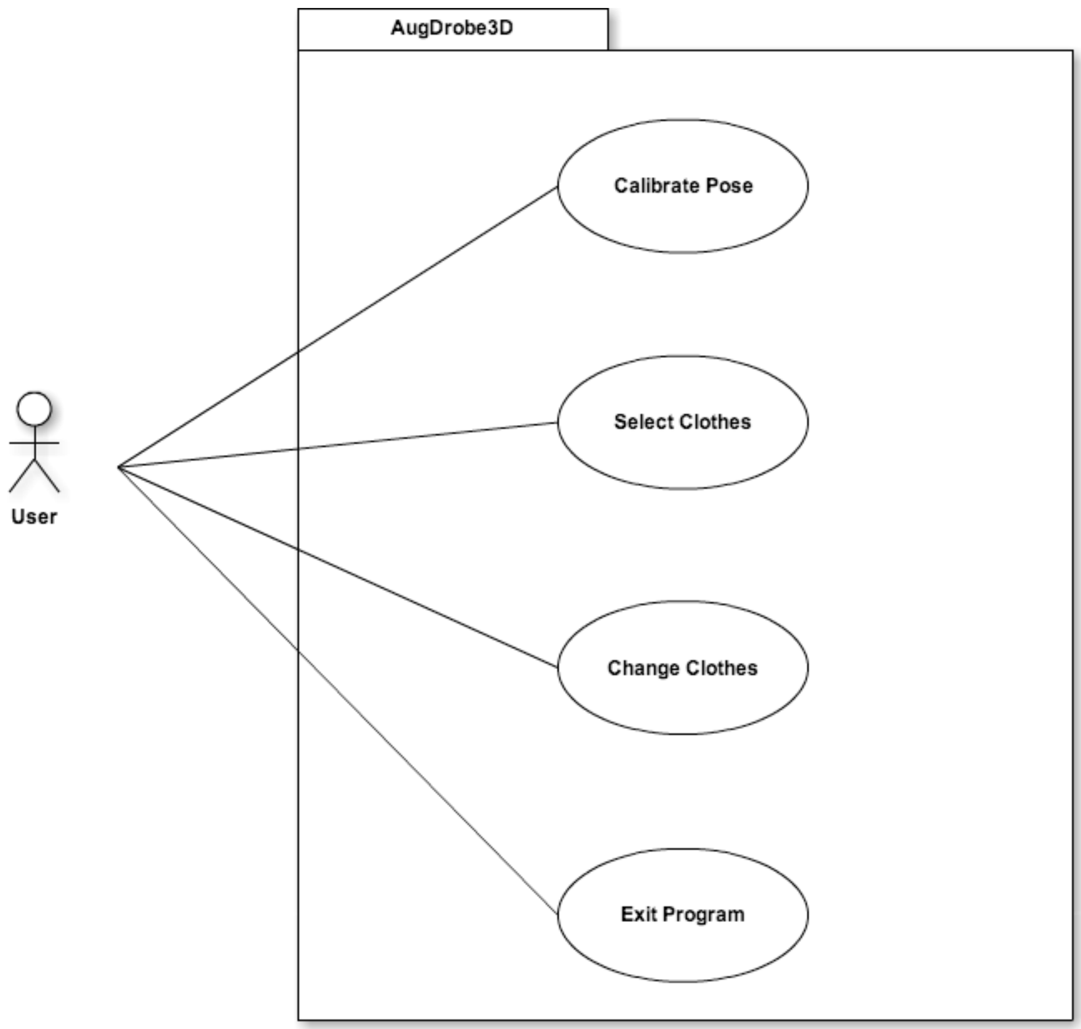
3.2.3 Measurement-taking process

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 Non-functional requirement

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 630 and above

Memory: RAM 4GB and above

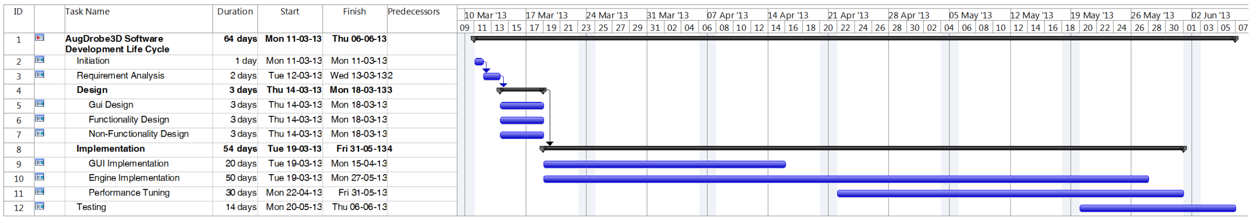
Hardware: Kinect Sensor

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. **

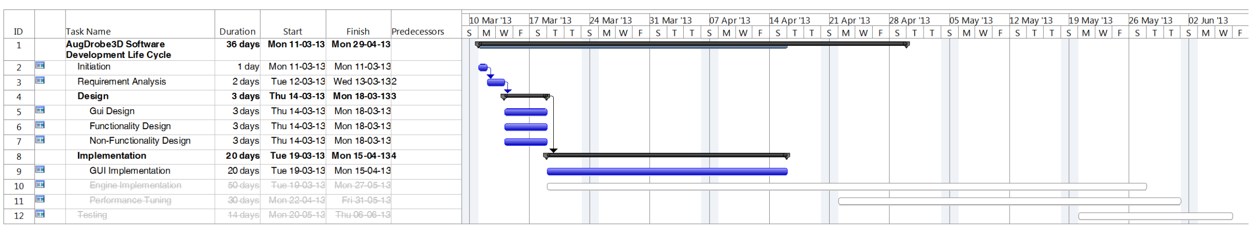
4.3 Gantt Chart

4.3.1 Original Project Gantt Chart



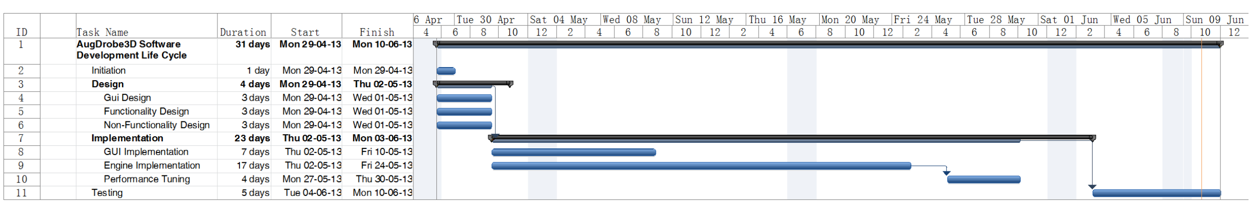
ID	Task Name	Duration	Start	Finish	Predecessors
1	**AugDrobe3D Software Development Life Cycle**	**64 days**	**Mon 11-03-13**	**Thu 06-06-13**	
2	Initiation	1 day	Mon 11-03-13	Mon 11-03-13	
3	Requirement Analysis	2 days	Tue 12-03-13	Wed 13-03-13	2
4	**Design**	**3 days**	**Thu 14-03-13**	**Mon 18-03-13**	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

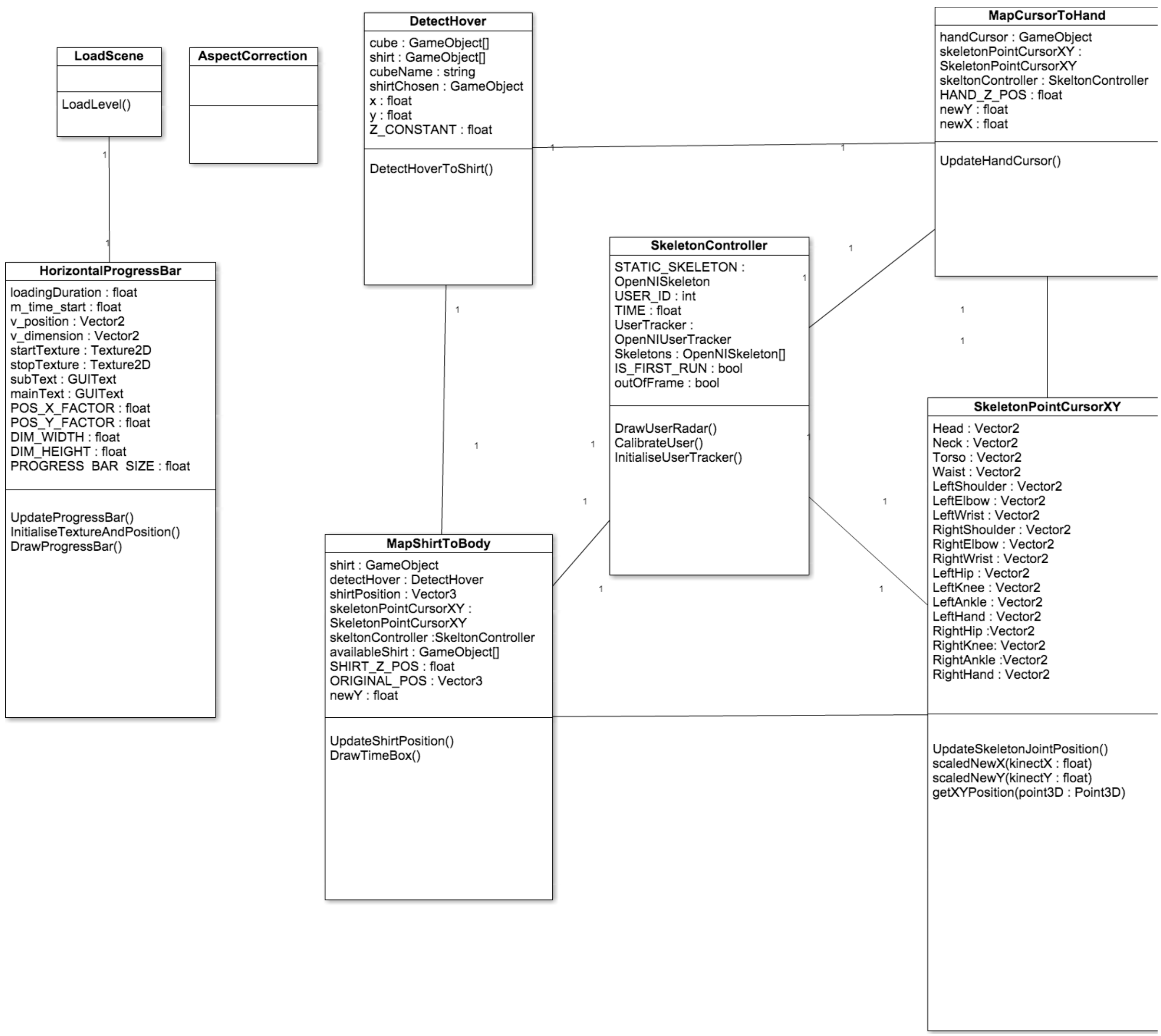
5 System Design

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.

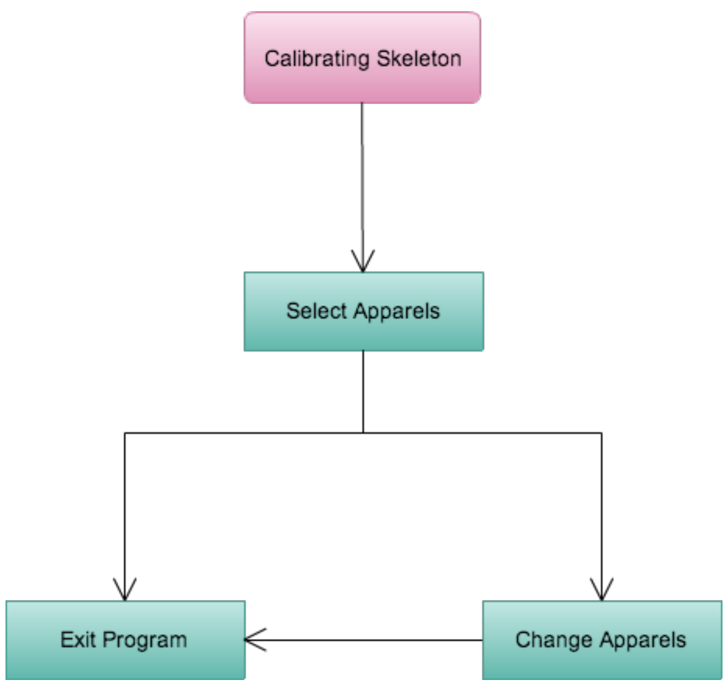
5.2 Class Diagram



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 Work on clothing models

6.1.1 Alignment of clothing

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

A better approach of alignment is to adjust the position, rotation and scale of the cloth models to the user instead of the other way around to allow more flexibility and makes it even user friendly. The use of markers in combination of video tracking and image registration techniques made it possible to receive some 3D information from theRGB image using a normal USB camera. Position, rotation and scale were adjusted using marker until the introduction of Kinect.

Kinect gives relatively cheap and easy access to a depth camera, and with the OpenNI framework the user's silhouette can be traced very precisely. By mapping the detected joints position to the clothing skeletal system, position and rotation is easily done.

6.1.2 Realism of clothing

The goal of a virtual fitting room is to give realistic visual experience of trying on apparels. Besides the alignment, the realism of the clothing movement is an important aspect in providing this experience. In the early days, apparels were just static images. It was only possible o see ow the clothes looked from the front view, and then moved onto multiple 2D images of the clothing from different angles which is providing a more realistic experience as it was possible to turn around and have a look from different angles. But the problem with this approach is that static 2D images is still being used and there is completely no animation or interaction at all with the clothes beside changing its position, rotation and scale hence we proceeded to the approach of using 3D images.

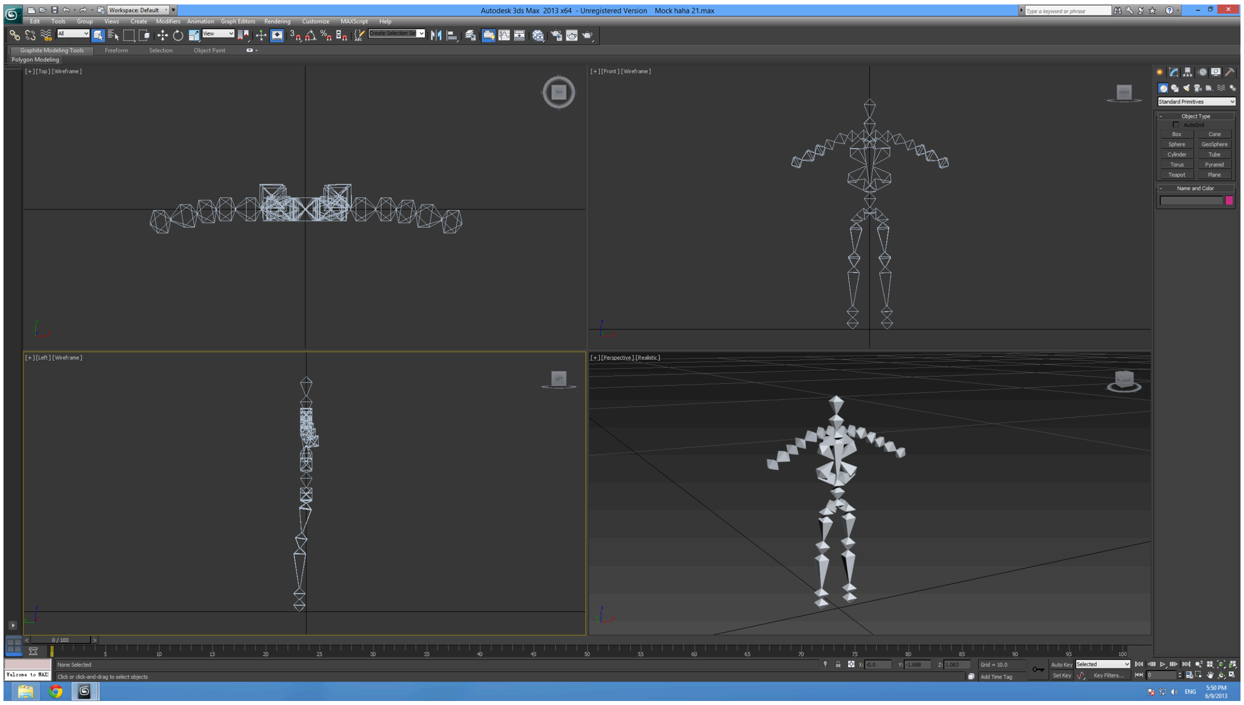
2D images are limited in few ways, which the most significant limitation is that cloth images are taken from a limited of angles, hence it wouldn't rotates smoothly but in fixed intervals. A 3D model could rotate freely and precisely. 2D images could not be simulated to perform moving an folding as user moves unlike a 3D model which also allows physical interaction between the virtual apparel and the user pose, which is again some impossibilities within 2D images.

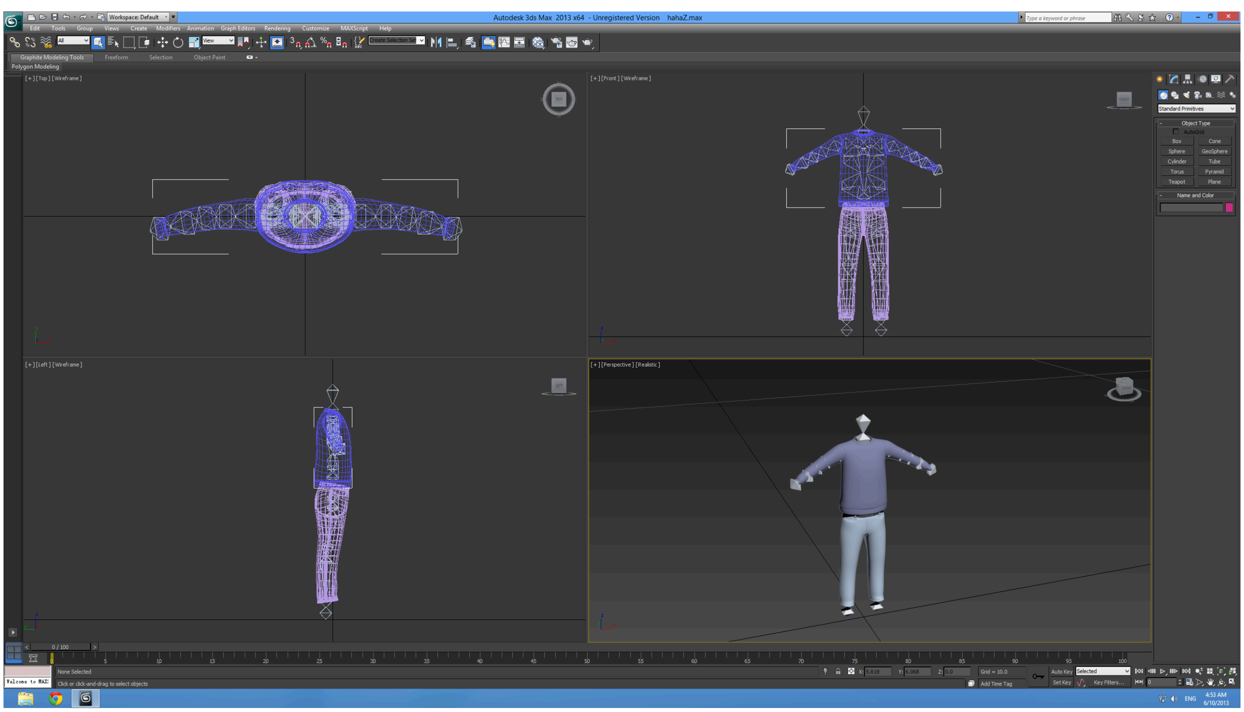
6.1.3 Skeletal mapping of clothing

Since Unity 3.0 the interactive cloth component is included as an alternative for skinned cloth which it adds cloth simulations to an arbitrary mesh. It also simulates interaction with physical objects through a physics engine which computes forces from interaction with colliders, and applies those on the apparels By modeling the user with an avatar and adding a collision hull to the body avatar of the user's avatar, a model with interactive components should react with the user pose just like in reality.

Full simulation of clothing physics in addition of collision response made this approach even more interesting for our application. The full interaction with environment and the user pose has a drawback: it is computationally complex.

By using 3ds Max 2013, skeleton and joints are added accordingly, depending on what main parts of user body are important to be detected and use as pivot point for cloth interactivity. By doing this the corresponding joints can be detected and mapped accordingly onto the user body. Cloth models are given a bone structure and each vertex in the mesh is tied to one or more bones with a certain value of weight. When transformation is applied to one of the bones it is also applied to each of the vertices connected to the bone scaled by the connection weight. The skinned cloth model then applies clothing simulation to the resulting vertex positions itself based on the speed of the movements. Per vertex a set of coefficients determine how freely the simulated model could move, by defining a volume of space within which the vertices may be moved.





7 Testing and Evaluation

7.1 Unit Testing

Unit testing is done from the very beginning of the project as this is the most basic testing and it is also the most important test to show the unit is working. To show that the function is working, it must pass the unit testing. Unit testing for this project is mainly on getting the skeleton data from the Kinect Sensor and trying to move the object of the User Interface.

7.2 Module Testing

As a project is made up with a lot of unit, similar unit is grouped as a module. Which means that a module is made up of units with similar functionality. Hence, module testing is mainly testing on the interaction between the units. Most of the module testing of this project is mapping skeleton and joints point with model bone point. For instance, a joint point from a skeleton should return value of x, y and z of the point and pass it to the model animation module. This is crucial because if the module doesn't "communicate" properly, the whole projects couldn't be continue as the result is not the result the user expect.

7.3 System Integration Testing

System Integration testing is very important as it test the communications of the modules. In order for the whole project to function normally, this test is essential as it could assure that the module interact properly and not giving errors. For instance, the user motion and the model animation should be matching. The animation of the model needed to follow the user's movement as well. For instance, when the user move their right hand, the right and of the model should also move instead of other parts. In this project, the main system testing is combining all the modules before it is release for alpha testing.

7.4 3D Model Testing

On the 3D Model Testing, we are focusing on the render and transformation animates of the cloth. We keep improving our skeleton for the cloth to ensure we have the best transformation on the cloth.

Most of the question we asked user are: Does the 3D models transformation animates smooth enough? Does the 3D models render correctly? Does the 3D models process the animation correctly? Does the 3D models follow the user's action? Does the 3D models's joint accurate enough with the user? How the difference on testing the cloth on 3D models and real cloth?

We can get most of the answered after users test our application, we always improve our cloth's skeleton after receiving feedback from users. We are trying to create a cloth's skeleton that can make the 3D models have a smooth and correct transformation animates, can follow user's action. We hope user can feel like the cloth is really wear on their body.

We expect the 3D models can follow all the basic action that user will do during they try the shirt in fitting room, such as turn to look the back of cloth. We also trying to implement some action that we will do usually like jump, squat and kneel. It is fun that a 3D models will follow the action, we hope user not only enjoy to control the 3D models and also will enjoy the transformation animates.

7.5 Methods

The testing methods of 3D models is very easy, user just need to complete a set of action that provided by us and they can do what ever action they like. User will request to use the system at start screen, they will start the 3D models test after complete calibration. At first, user will complete all the action requested after that user can do extra action they like. User also will need to answer the list of question we asked.

The action we request testers to do:

- Turn 45 degree of the body.
- Turn 180 degree of the body.
- T-Shape post.
- Stand at attention.
- Raise up both hands.
- Raise up one of their thigh.

The questions we asked our testers:

- Does the 3D models can follow the set of action we requested?
- Does the 3D models follow all your action?
- Does the 3D models transformation animates smooth enough?
- Does the 3D models support enough action to test the cloth?
- Can you accept to use 3D models test the cloth?
- How accurate are the joint?

7.6 Test Results

I. We tested the app on a variety of users, 3 male, approximately between the ages of 20 and 24.

II. Observations:

1. All testers had no trouble to perform the set of action we requested.
2. Two testers said they have few action the 3D models no perform the transformation animates correctly but will recover automatically after few movements.
3. One of the testers said he thinks the animation can be smoother and others think the animation is smooth.
4. All testers think the action that 3D models support is enough to test the cloth.
5. All testers can accept to use 3D models to test cloth.

6. Two testers said the joint is detect accurately but one of the testers show us some action the joint will perform not normally.

III. Some qualitative feedback:

1. Some testers told us sometimes when they jump, the 3D models cannot perform the animation correctly, we have try the action several time and fix the problem.

2. Two testers told us the left shoulder was not raise up equally with right shoulder, we have adjust the 3D models to fix it.

3. About a feedback ask us to make the 3D models animation smoother, we have try to split some of the large bone to small, try to make the skeleton transformation can be more natural and smooth.

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.Reallusion.com (n.d.). The Comparison of OBJ and FBX. [online] Retrieved from: http://www.reallusion.com/iclone/Help/3DXchange5/Pipeline/05_Export/The_Comparison_of_OBJ_FBX.htm [Accessed: 9 Jun 2013].