TAYLOR'S UNIVERSITY

SCHOOL OF COMPUTING & IT

PROGRAMMING PROJECT

AUGDROBE3D

BY: LAI SAI HOO

SUPERVISED BY: RIMI AZIZI RAMIDIN

FEB 2013

AugDrobe3D - Kinect Augmented Reality Wardrobe

Lai Sai Hoo

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 clothing. The user pose is tracked using the Microsoft Xbox Kinect sensor and the virtual clothes are mapped accordingly with the user silhouette and poses. Clothing 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 Research

2.1.1 Motion Sensing software development kits

At the beginning of the project, Microsoft Kinect SDK was our first option for SDK for this project. This is because of few of the advantages that Microsoft Kinect SDK have over the other SDK. However, the Microsoft Kinect SDK doesn't work well with the Game Engine we use which is Unity3D. After some effort trying to use Microsoft Kinct SDK, we start looking for the alternative SDK. Which in results, we found an open-source framework which is able to read sensor data from Kinect. In conjunction with the framework, PrimeSense, a depth sensing reference design based on Kinect and NiTE, a motion tracking middleware. With these additional open source drivers, OpenNI could capture the motion of the user from the Kinect.

Other than the SDK problem, the communication between Kinect and Unity3D is still a problem. As they don't come with a library that could be used directly by the Game Engine, so we found ZigFu. ZigFu act as the middleware between Unity3D and Kinect.

The other reason OpenNI is selected over Microsoft Kinect SDK is because of the platform. As this is developed in a team, some members are using Macintosh instead of Windows while Microsoft Kinect SDK is only build for Windows, so, we decided to give up on Microsoft Kinect SDK and go for OpenNI.

2.1.2 OpenNI with PrimeSense and NiTE

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 2 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 the user. Using OpenNI with these middleware libraries, we can control the project with just the user interaction itself.

2.1.3 ZigFu

ZigFu Development Kit (ZDK) is a cross-platform development kit that provide support for motion-controlled apps. ZDK is a library to simplify the porting of cross-platform applications. ZDK is used as the project is meant to be build for all platform.

2.1.4 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.

2.1.5 WPF

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

2.1.6 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.

2.1.7 GitHub

GitHub is a web-based hosting service for software development projects that use the Git revision control system. GitHub offers free accounts for open source projects. As this project is developed by more than one developer, versioning tool is very important as conflict can easily occur and with Git revision control system files can be easily merged and conflict could be solved with ease.

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 2.1a.

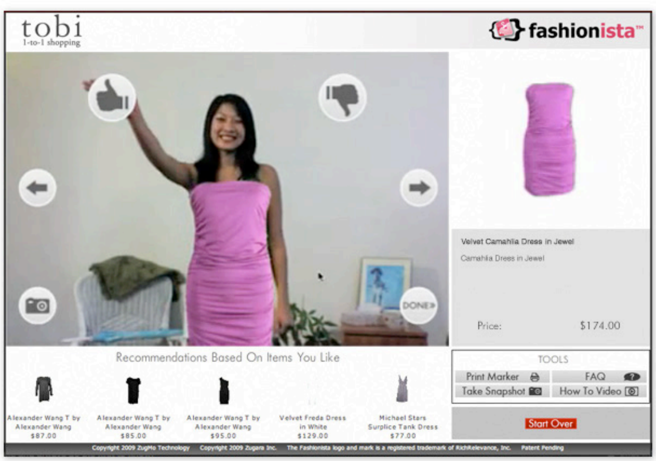


Figure 2.1a) Using tracker

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 2.1b. This solution is currently the state of the art augmented reality dressing room.

Figure 2.1b

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

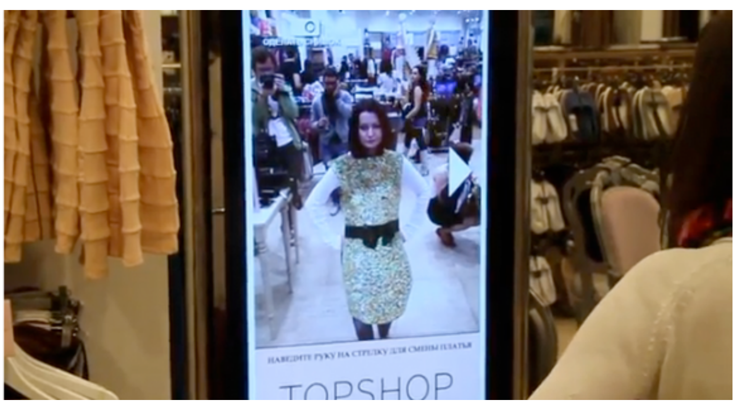


Figure 2.1c) Topshop virtual fitting in the UK

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.

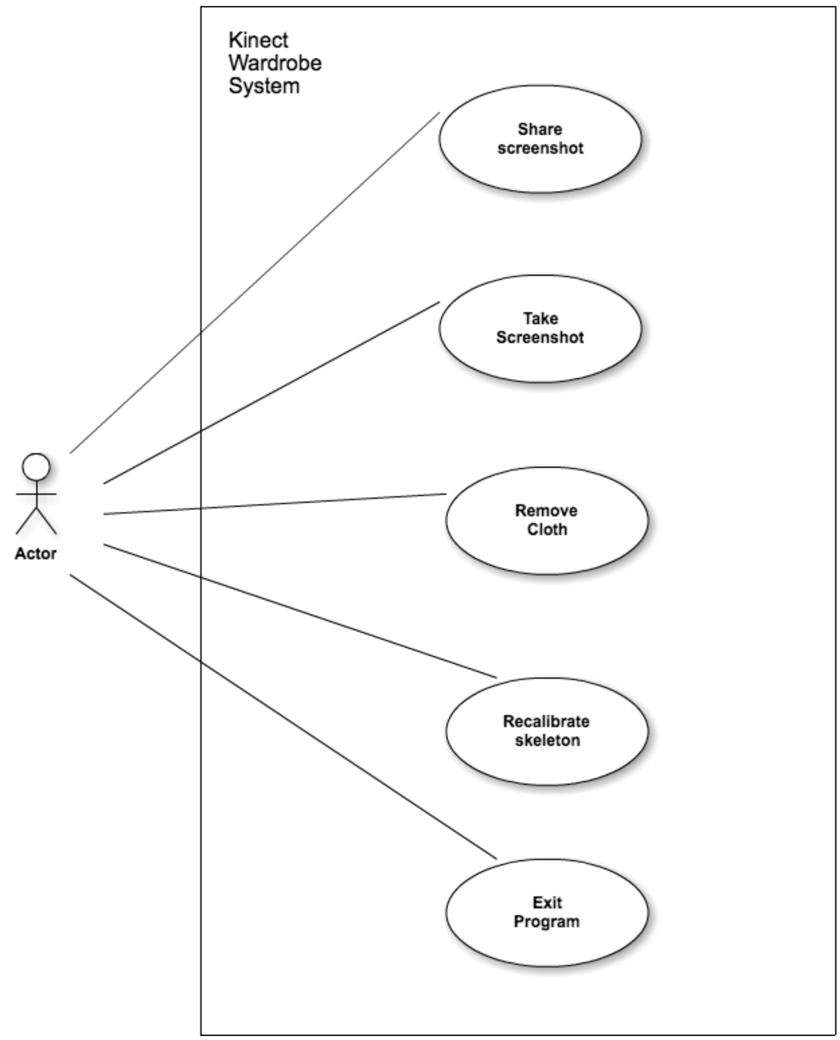
3.2.3 Measurement-taking process

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 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 630 and above

Memory: RAM 4GB and above

Hardware: Kinect Sensor

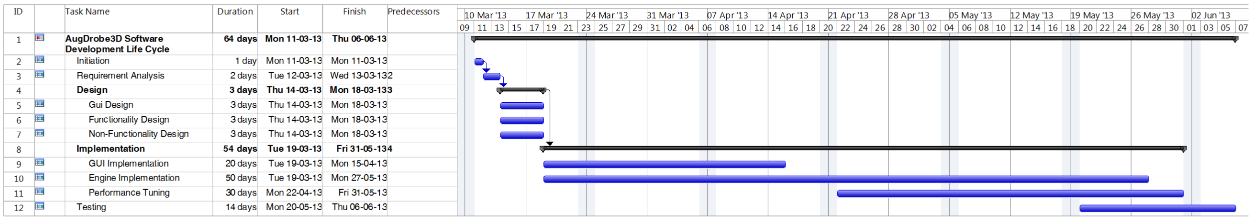
Software: OpenNI v1.5.4.0, PrimeSense v5.1.2.1 and NiTE v1.5.2.21 installed

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. **

** Microsoft Kinect SDK must be Un-installed before installing OpenNI and all the other libraries as there would be conflict and the program won't run properly with Microsoft Kinect SDK

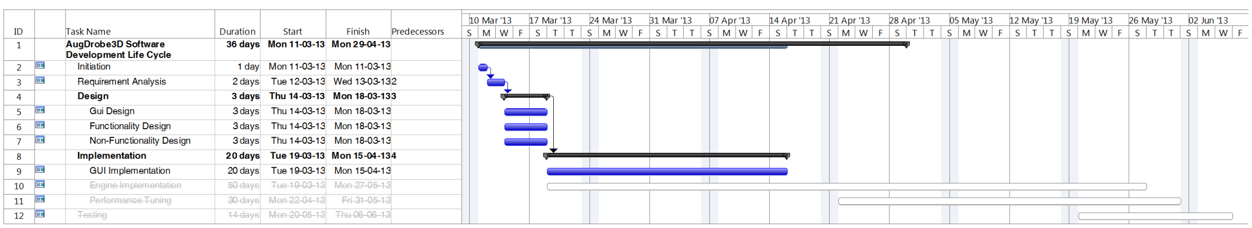
4.3 Gantt Chart

4.3.1 Original Project Gantt Chart



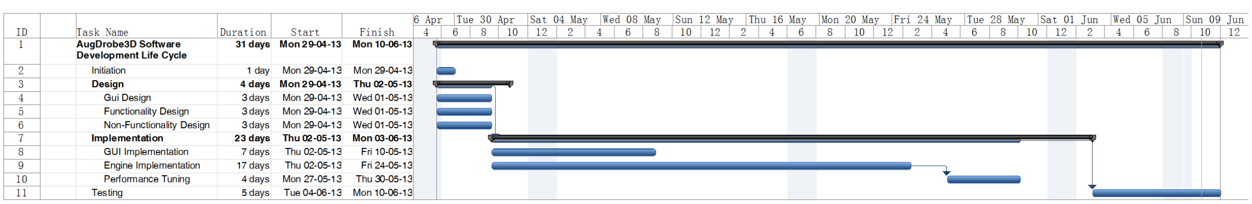
ID	Task Name	Duration	Start	Finish	Predecessors
1	**AugDrobe3D Software Development Life Cycle**	**64 days**	**Mon 11-03-13**	**Thu 06-06-13**	
2	Initiation	1 day	Mon 11-03-13	Mon 11-03-13	
3	Requirement Analysis	2 days	Tue 12-03-13	Wed 13-03-13	2
4	**Design**	**3 days**	**Thu 14-03-13**	**Mon 18-03-13**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

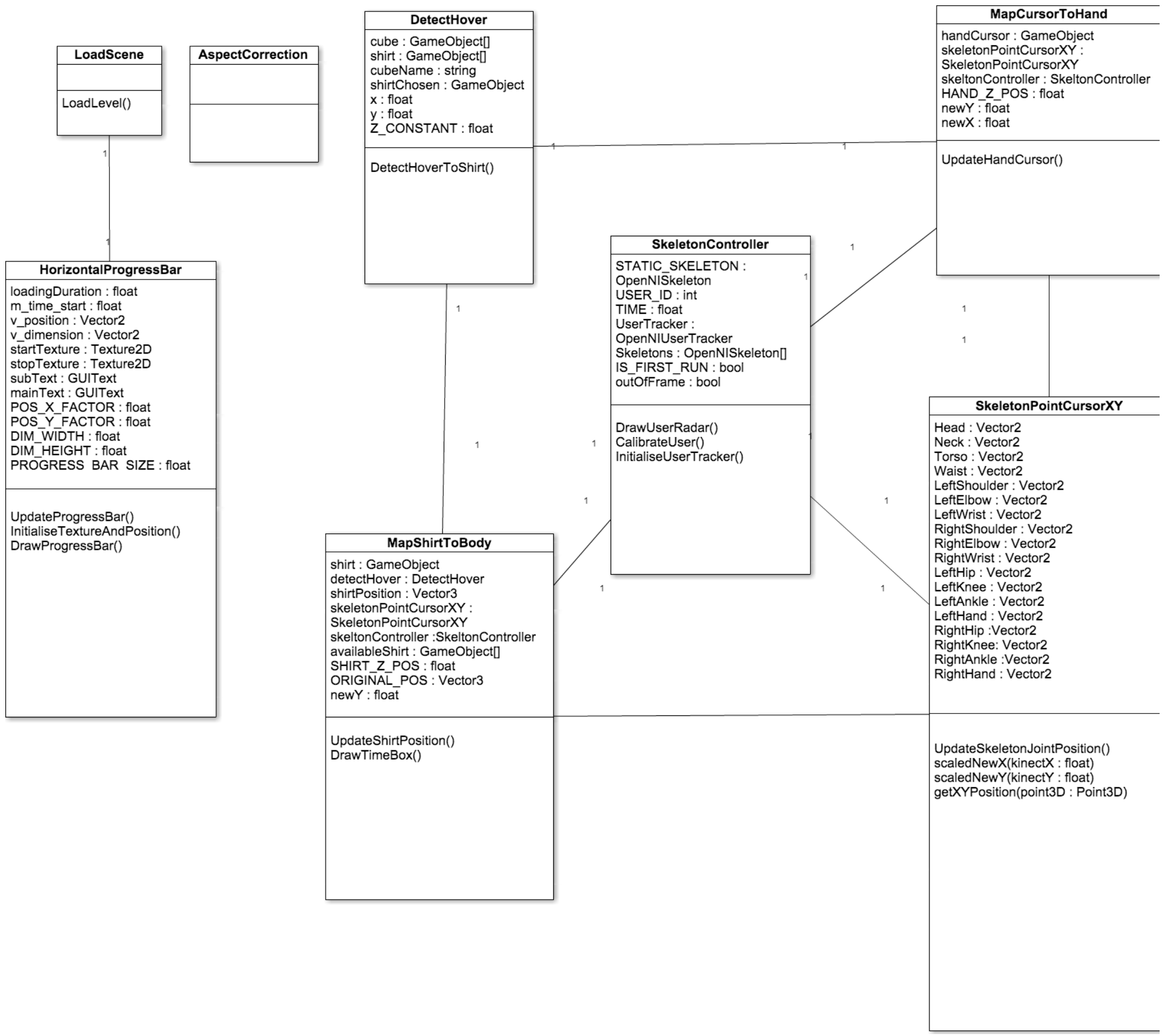
5 System Design

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.

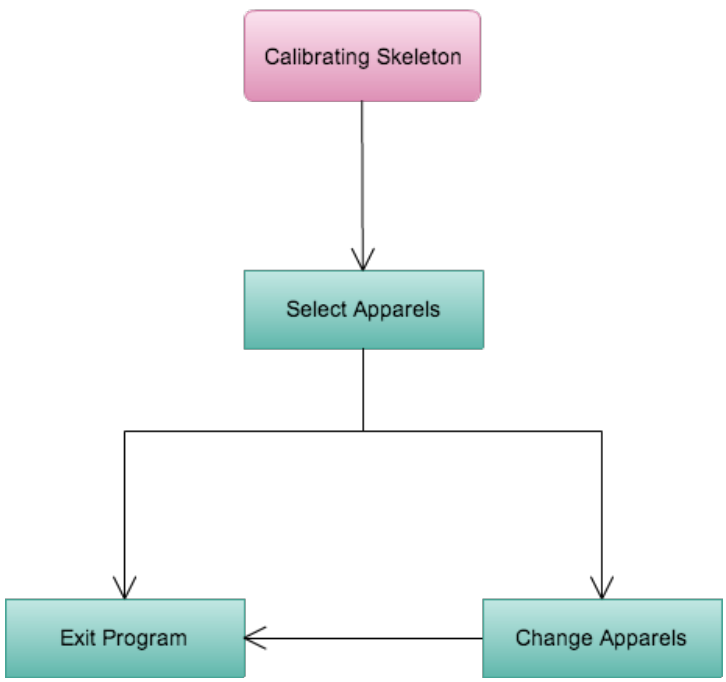
5.2 Class Diagram



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 Milestones

6.1.1 Develop application with Windows Presentation Foundation (WPF), and Microsoft Blend

The project is initially developed using WPF under Microsoft Visual Studio 2012. In WPF, graphics are rendered using Direct3D and UI is designed and written under the Extensible Application Markup Language (XAML). UI design is extremely simple on WPF as graphical objects are easily built, either by script or the interface builder included in the Visual Studio 2012.

User interface is pretty simple with few buttons on top simulating the use case designed in the very early days since the project had started. Initial use case includes screenshot, sharing and exit. Background of the natural user interface (NUI) is the image recorded by the RGB camera. To the right of the NUI is the hat collection where user could choose different varieties of head accessories, where to the left of the NUI is the collection of clothes which are into two main categories: topwear and bottomwear.

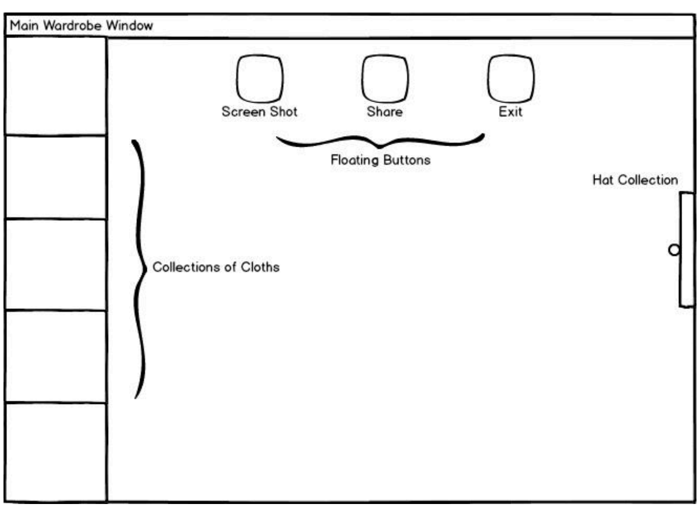


Figure 6.1.1a) Mock up of the interface

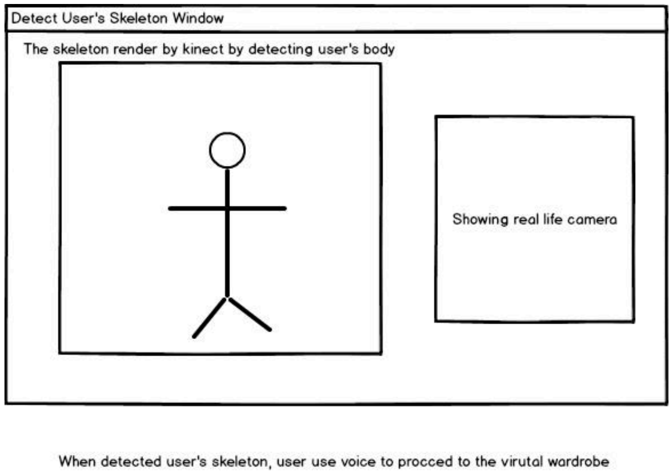


Figure 6.1.1b) Skeleton detection interface

For object modeling, we tried using Microsoft Blend as the best approach as Blend could directly export Waveform 3D models (.obj) files as a XAML type script that could be directly rendered using DirectX which will be described further in 6.1.3.

6.1.2 Using the Kinect SDK for motion sensing

Microsoft released the Windows-compatible Software Development Kit for Kinect in June 2011. It provides Kinect capabilities to developers to build applications with C++, C#, or Visual Basic by using Microsoft Visual Studio 2010 and includes following features, which are the raw sensor streams, skeletal tracking, noise suppression, voice recognition.

Recording and playback is also available from Kinect Studio which is one of the component of the Kinect SDK since version 1.5. As Kinect SDK and WPF is very compatible it is very easy for the project to start and be on track as the sole input works perfectly fine.

6.1.3 Moving into DirectX for rendering 3D model objects

For the cloth models to position, move and rotate according to user pose, DirectX is needed to be implemented underlying the 3D model to map the movements. It could allow the cloth models to fold and scale accordingly but the drawback is that it is computationally complex as there arbitrary mesh of the model need to be explicitly calculated for respective cloth model as it could differ from models.

6.1.4 Realising failure of reaching objective and expectation on the current platform

As few weeks went on after the project starts, the team found out that relying on DirectX10 may not achieve the project objective in the designated timeframe that had been set earlier due to the steep learning curve of DirectX10 native library and implementing it in our system, hence alternatives had been looked up without the project development being halted at the same time.

Cloth models are found difficult to attach motions onto due to the weakness of the 3D model alone which is hollow and nothing that could relate joint motions with the movement of the 3D model. This poses a big challenge as it is very complex.

6.1.5 Switching over to Unity3D and OpenNI

For this project , the whole system is developed in Unity3D. Unity3D has several advantages over others 3D game engine as Unity3D support multiple platform (Linux , OS X and Windows). Therefore, it makes this system very portable as it can export to multiple difference environment. In other words, more users would able to use this system, indirectly will discover much more bigger market for system that only support single operating system. Besides, Unity3D has built in IDE which has a easy learning curve as the user interface of Unity3D's IDE is very user-friendly.

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.

Using OpenNI in conjunction with Unity3D allows the application to be developed in various platform, which unlike Kinect SDK which only provide support for the Windows platform. It generates more flexiblity and at the same time provide ease of development.

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 User Interaction Testing

7.4.1 User Interaction

For our user test, we want to take our existing interface and see what parts of it make sense to users.

The questions we want answered are: Is the interface intuitive? Do we have too few, too many, or just the right amount of instructions? Do people want to use their hand as a cursor, or does it make more sense just to use a mouse? Does the application flow make sense? Is this application understandable by non-techies? Does it give realistic values? Are there specific body types that it works best with? Is it valuable to have a redo button? Is it fun? Would people use it?

We expect that most of these question can be answered simply by putting users in front of our application and then talking to them about what they think. Our goal in general is to provide a smooth and seamless user experience. Therefore we are looking to user testing to see what aspects of our application work well for users and determine which aspects are awkward.

We expect that the interface will be intuitive given our minimalist design, and by intuitive we mean that it should take users less than 10 seconds on the homepage to realize that they need to hover over the apparel to proceed. Also given the minimalist design and simple application flow we expect that users won't get too lost, and that error recovery should be as simple as restarting the process as one pass through takes less than 30 seconds. We think that people will be excited about using their hand as a cursor but that in practice users would prefer to use a mouse. We have no expectations about what kinds of body types our system will work best for - so whatever we learn will be valuable in that respect. We think people will find our system fun - or at the least a novelty.

7.4.2 Methods

We will test our system on a range of users. For now.

We will ask users about the last time they went online shopping, to find out what pain points they run into, if they actually know their size, and if our application will fit their needs. We'll then observe how excited they seem about the application (both before and after using it.

As they use the application, we will observe how well they seem to understand the application flow. In particular, we will watch out to see if they get confused or make mistakes that they then want to undo at any point in the experience. We will also observe whether they choose to control the homepage with the mouse or their hand. We will then ask them to also try it the other way, and see which one is faster, and which they prefer. We will see how many of them want to "redo" the process (and if they do, ask why).

We will measure people and compare the measurements from our system with their actual measurements.

Questions we asked our testers:

- Please talk aloud about your thoughts as you use our app.
- Describe the last time you went shopping (in a retail store/ online).
- Describe your experience using our app.
- Do you think you'd be more likely to use the cursor or a mouse?
- In what venue do you think this would be most useful?
- How accurate were the measurements?

7.4.3 Test Results

I. We tested the app on a variety of users, 5 male, approximately between the ages of 20 and 32.

II. Observations:

1. About 50% of users had trouble getting the cursor to select the pants that they wanted

2. Only one person's default action was to use the mouse instead of the Kinect cursor

3. Almost all users did a little dance when they saw the skeleton.

4. About 75% of users had some trouble reading the instructions from far away and asked us what they were supposed to do.

5. By the time they had read the instructions, their measurements had been taken, likely incorrectly

6. Everyone thought it was too fast ("Did it already go?")

7. Men's measurements were surprisingly accurate.

III. Some qualitative feedback:

1. Men are more likely to buy pants online, because their sizes are more consistent across stores. Women would rather go to the store, because they're going to need to try on a huge pile of pants to figure out which one fits the best anyway. Some women don't even bother looking at certain stores, because the chances are so slim that they'll fit anyway.

2. People tended to like having a cursor controlled from afar. ("It's nice to be able to maintain your distance from the system and not need to reset yourself to redo everything") But they also said that they would probably choose the mouse by default if they could have the time to get to their position before the measurements started (countdown of some sort)

3. Our color scheme at times made it difficult to discern different features on screen. "My distance vision is having trouble with this grey on grey"

4. What if you linked those sizes to the stores' websites?

5. Our system had too much reliance on text instructions.

6. Some users didn't know what waist/hips/inseam meant. Men just wanted to know their pants size (waist-inseam).

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].