VIRTUAL WARDROBE FOR PHYSICALLY IMPAIRED USING MICROSOFT KINECT SENSOR

Md. Farhan Hamid 12201044

Supervisor: Dr. Md. Ashraful Alam

Department of Computer Science and Engineering,
BRAC University.

Submitted on: 21st August 2017

DECLARATION

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor	**Signature of Author**
______________________________	______________________________
Dr. Md. Ashraful Alam	Md. Farhan Hamid

ACKNOWLEDGEMENTS

I am eternally grateful to my professors, friends and family who helped me in every step of the way to complete this research.

Firstly, I would like to express my sincerest gratitude to mysupervisor Dr. Md. Ashraful Alam for his continuous support, encouragement and guidance in every step of my research. Without him this research would not have been completed.

I would also like to thank Ms. Yeasmin Parvin, executive director of National Forum of Organisations Working with the Disabled (NFOWD) for providing the assistance to make this research a reality.

Table of Contents

List of Figures

List of Tables

ABSTRACT

This research intends to solve the virtual wardrobe problem for physically impaired. It is difficult for the physically impaired to try out outfits by themselves in a trial room. It is far more convenient for them to try the outfits virtually. To best of our knowledge, there is no such system for the physically disabled. Our approach is to solve this problem using head gestures and voice commands. We intend to solve this problem using Kinect Head Tracking and Voice Recognition features. Pitching their heads right or left, they can navigate through the items and blinking allows to select. The system will be designed with friendly interface, simple process and easy operations for the physically impaired to use with ease.

CHAPTER 1

INTRODUCTION

1.1 Motivation

The World Health Organization (WHO) estimates that 7-10% of all human being have some degree of impairment or disability [1]. Unlike others, they do not have the privilege to try out clothes on their own. Furthermore, it is impossible for them to trial clothes on a trial room. For example, someone without an arm has to go through a lot of troubles to try on a cloth. Sometimes, it is almost impossible for people with disabilities to try out clothes without assistance. Thus, this system aims to provide an alternative for them. Using this system they will be able to try different clothes and see the outcome on a monitor. All these will be in real time.

1.2 Objectives

Virtual wardrobe has been a concept tried in the past. There have been approaches based on manipulating virtual clothes, internet based garment design, trying outfits on avatars online. Gesture control had been approached using multiple-fingers static hand gestures, body part gestures and gestures to control human structures. These approaches have served well for the implementation of virtual wardrobes. Regular people can use it with ease. Most of the system uses navigation through the user's hands. But this will be impossible for someone who doesn't have hands. Thus, we have tried to solve this conundrum for the physically disabled. [1] There is no existing virtual wardrobe system for the physically disabled. Our approach to solve this problem is using head gestures. As everyone can use their head for navigation, we have chosen to use head gestures for navigation. The system will be designed with friendly interface, simple process, and easy operations for the physically impaired to use with ease.

CHAPTER 2

BASIC CONCEPT

2.1 Overview

Our aim is to build a virtual wardrobe system for people with disabilities. We have considered having a easy to use and user friendly interface for the users. Using this they can easily try out outfits to see how those fit. We have incorporated Microsoft's Kinect for XBOX One in our research.

Virtual Wardrobe has been implemented in the past following many approaches. But to best of our knowledge, no such system exists for people with disabilities. It is our goal to adopt the best from the previous to work on delivering a product that works accurately with the users.

2.2 Literature Review

Several virtual clothing systems in the past used the 3D avatar approach. Some researches like Cordier et al. worked on making garment design process easier by manipulating virtual clothes [2]. Yuwei et al. and other researches focus on making avatars and virtual clothes more realistic [3]. Internet based garment design is also popular among researchers, which allows online shoppers to try out outfits on their avatars [4].

Other approaches involve use of markers on the shirt to change clothing patterns by overlying different patterns on the mirror image.

Before the invention of Kinect, people used one or multiple video cameras in computer vision systems [5]. Du et al. tried using computer vision to track human body has been famous for a long time. Other systems use depth value for tracking [6].

Gesture Control is another area that has been worked on for a long time. Many researches are based on recognizing multiple-finger static hand gestures. Some are based on other body part gestures. There have also been works on gestures to control virtual human structures [7].

One of the main goals of the previous works on gesture control is to make it easy to be used by the elderly and the disabled [8].

One of the first works on a virtual wardrobe using Kinect is "Augmented Reality: Virtual fitting room using Kinect" by Ziquanet. al. [9]. Ziquan uses the Kinect sensor to convert a user's motion to motion data and with other virtual environment elements, like user interface components and dress models, are integrated as a virtual scene. It outputs a 2D avatar for the user to try on different apparels.

Since then there have been other works on virtual dressing rooms and magic mirrors using Kinect. [10]

There have been couple of works on a virtual dressing room in BRAC University. The first one is "Magic Mirror Using Kinect," by A. B. Habib, A. Asad, W.B. Omar, BRAC University (2015).They used a front image for each product to superimpose it onto the user and outputted a 2D virtual avatar for the user [11]. Some of their limitations were:

- User has to move to adjust cloth.
- Inaccurate dresses.
- No user interface.

- Dresses do not adjust to user movement.

Another one is “Design and Implementation of a Magic Mirror Using Kinect,” by Monir Md. , Siddique Md. , Tanni A. , Hashem Md. , BRAC University(2016). They have introduced a virtual dressing room application with avatar generation and tracking in real time [12]. It uses a static position, display size of dresses, using hand gestures for navigation, dynamic dresses based on user size. Some of the limitations are:

- Does not contain 3D avatars,
- Dresses do not adjust to user rotations.

CHAPTER 3

SYSTEM DESIGN

3.1 System Architecture

We intend to solve this problem using head gesture tracking for Kinect. Primarily, we are using pitch navigation and nodding. Pitching their heads right or left, users can navigate through the items and nodding twice would allow to select a specific item.

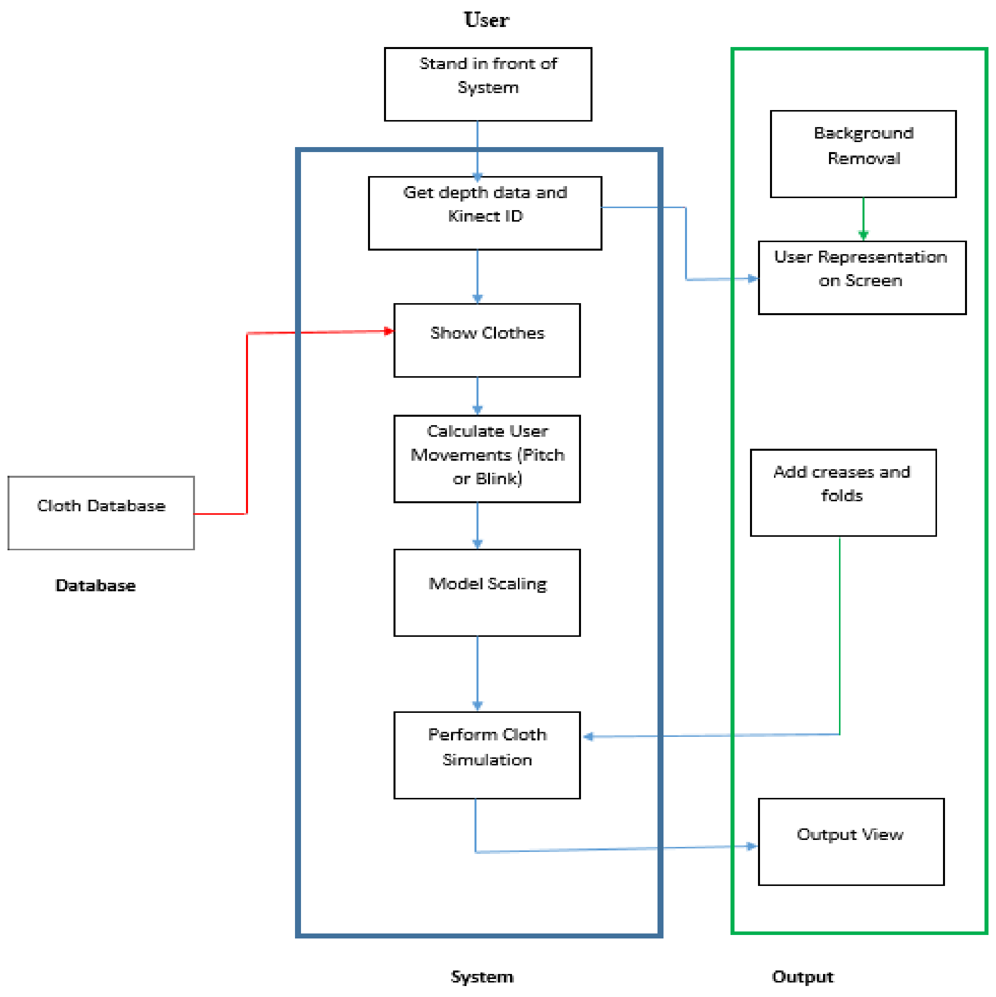


Fig 1. Architecture of Proposed System

The aim of the system is to make life easier for people with disabilities by combining technology and real life components together.

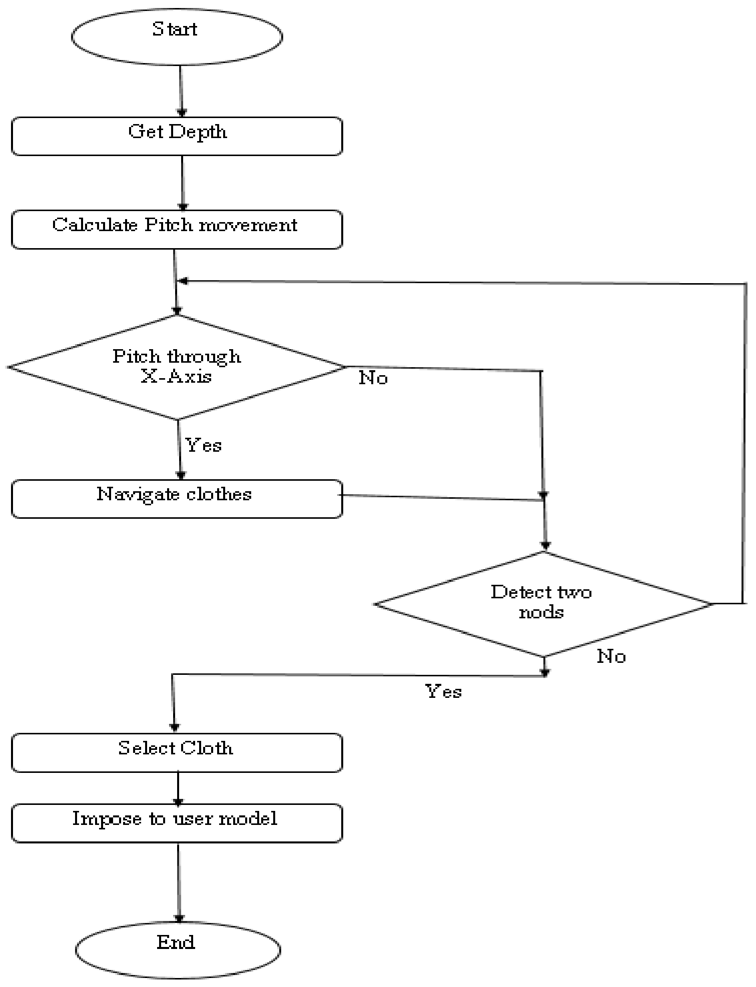


Fig 2. Flowchart for the Proposed System

Figure 1 shows the system as a whole. The system would start with the input of a person with disability. The system would calculate the depth points of that person to make adjustments. It would then move on to detect the amputee. Afterwards the movements of the head would allow the user to navigate and blinking would select. The system would know this by the change in the depth points. Selecting a cloth a user would be able to see himself on that apparel. The flowchart on Fig. 2 shows the workflows of the system. Here we see the individual parts of the system working as a whole. The system would make adjustments, such as, creases and folds based on the disable limb to make closest similarity with the user shown on the activity diagram in Fig. 3.

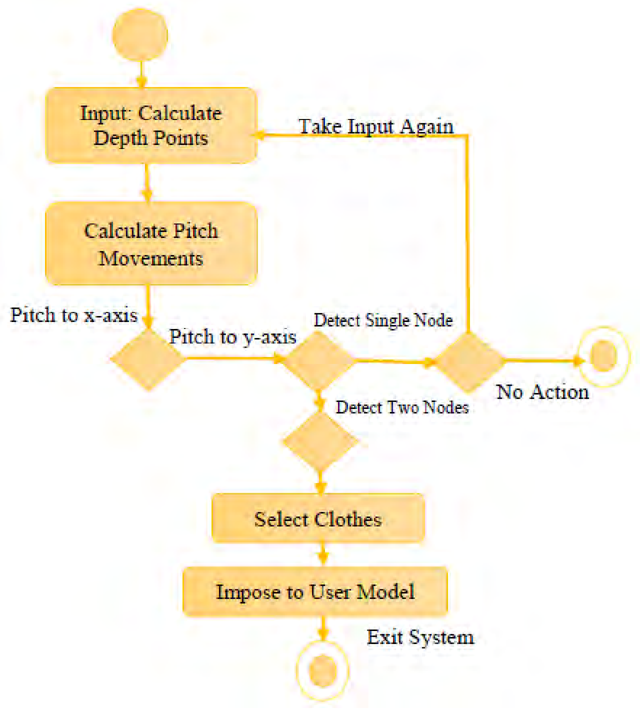


Fig 3. Activity Diagram for the Proposed System

3.2 System Components

In order to implement our system, we would need the following hardware and software – a) Kinect for Xbox 360 b) Microsoft Visual Studio c) Microsoft Kinect For Windows SDK d) NUI Skeleton API.

Using the above mentioned components we implemented the system. Setting up the Kinect for Xbox 360 with the Kinect for Windows SDK, we built an interface with Microsoft Visual Studio

2015. We plugged the Kinect to the computer's USB and powered it through an AC adapter. We used Windows 10 for the operating system.

3.2.1 Microsoft Visual Studio 2015

Microsoft Visual Studio is Microsoft's own integrated development environment (IDE). Visual Studio is used to make software for Microsoft Windows, mobile and web apps. It also uses platforms like Windows API, Windows Forms, Windows Presentation Foundation (WPF), Windows Store and Microsoft Silverlight. It has support for both native and managed code.

Its code editor has support for Intellisense and uses code refactoring. It also has a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer

Visual Studio supports 36 different programming. It supports most of the main Built-in languages include C, C++, VB.NET, C#, F# and TypeScript. Support for other languages such as Python, Ruby, Node.js, and M is also available provided the dependencies are installed. Our system was coded in C#. We used Microsoft Visual Studio 2015 as our IDE.

3.2.2 Kinect for XBOX 360

Kinect (codenamed Project Natal during development) is a line of motion sensing input devices by Microsoft for Xbox 360 and Xbox One video game consoles and Microsoft Windows PCs.[13] Based around a webcam-style add-on peripheral, it enables users to control and interact with their console/computer without the need for a game controller, through a natural user interface using gestures and spoken commands.

The first-generation Kinect was first introduced in November 2010 in an attempt to broaden Xbox 360's audience beyond its typical gamer base. A version for Microsoft Windows was released on February 1, 2012. A newer version, Kinect 2.0, was released with the Xbox One platform starting in 2013.

Microsoft released the first Beta of the Kinect software development kit for Windows 7 on June 16, 2011. This SDK was meant to allow developers to write Kinecting apps in C++/CLI, C#, or Visual Basic .NET.

The components of Kinect for Windows are mainly the following:

1. **Kinect hardware:** Including the Kinect sensor and the USB hub, through which the sensor is connected to the computer;

2. **Microsoft Kinect drivers:** Windows 10 drivers for the Kinect sensor;

3. **NUI API:** core of the Kinect for the set of Windows API, supports fundamental image and device management features like access to the Kinect sensors that are connected to the computer, access to image and depth data streams from the Kinect image sensors and delivery of a processed version of image and depth data to support skeletal tracking.

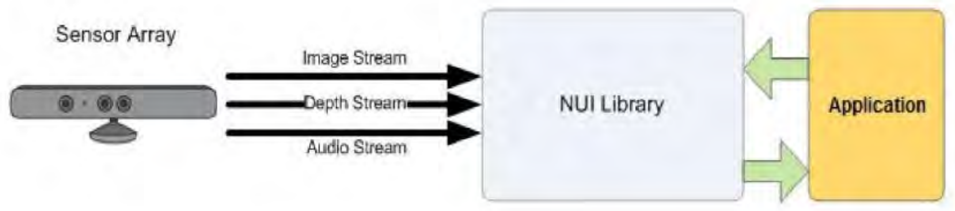


Fig 4. Hardware and software interaction with an application [11]

Kinect sensor mainly provides 3 streams: Image stream, depth stream and audio stream, with detected range from 1.2 to 3.5 meters. At this stage, the first two streams would be utilized for development of human model, cloth simulation and GUI. The middle camera is a 640×480 pixels @ 30 Hz RGB camera, providing image stream which is delivered as a succession of still-image frames for the application. The quality level of color image determines how quickly data is transferred from the Kinect sensor array to the PC which is easy for us to optimize the program on different platform. The available color formats determine whether the image data that is returned to our application code is encoded as RGB. The middle camera is a 640×480 pixels @ 30 Hz RGB camera, providing image stream which is delivered as a succession of still-image frames for the application. The quality level of color image determines how quickly data is transferred from the Kinect sensor array to the PC which is easy for us to optimize the program on different platform. The available color formats determine whether the image data that is returned to our application code is encoded as RGB. 13 The leftmost one is the IR light source with corresponding 640×480 pixels @ 30 Hz IR depthfinding camera with standard CMOS sensor on the right, which mainly provide the depth data stream. This stream provides frames in which the high 13 bits of each pixel give the distance, in millimeters, to the nearest object at that particular x and y coordinate in the depth sensor's field of view.

3.2.3 Nui skeleton API

Among NUI API, NUI Skeleton API provides information about the location of users standing in front of the Kinect sensor array, with detailed position and orientation information. Those data are provided to application code as a set of 20 point, namely skeleton position. This skeleton represents a user’s current position and pose. Our applications can therefore utilize the skeleton data for measurement of different dimension of users’ part and control for GUI. Skeleton data are

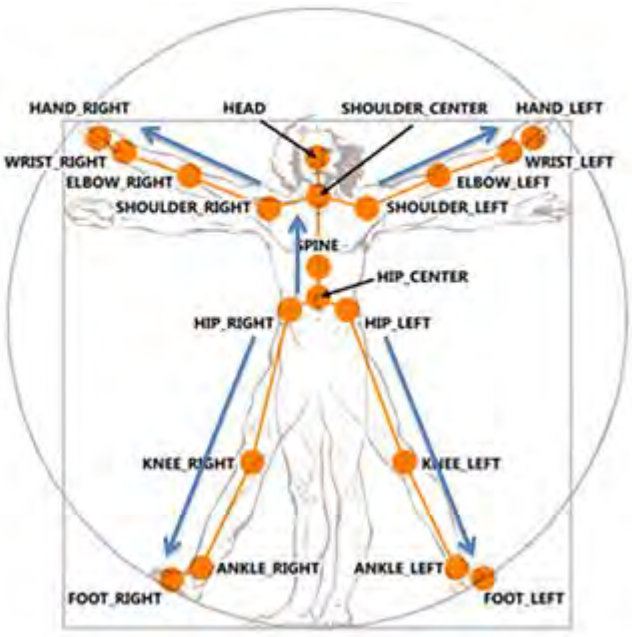


Fig 5. Skeleton joint positions relative to the human body [14]

passing 14 a buffer while our application can then use an event model by hooking an event to an event handler in order to capture the frame when a new frame of skeleton data is ready.

CHAPTER 4

VIRTUAL WARDROBE IMPLEMENTATION

4.1 Color Stream

Color image data can be used at different resolutions and formats. The format determines whether the color image data stream is encoded as RGB, YUV, or Bayer. You may use only one resolution and one format at a time.

The sensor uses a USB connection that provides a given amount of bandwidth for passing data. Your choice of resolution allows you to tune how that bandwidth is used. High-resolution images send more data per frame and update less frequently, while lower-resolution images update more frequently, with some loss in image quality due to compression.

Table I describes the formats in which color data can be used. Color formats are generally computed from the same camera data, for all data types to represent the same image.

Color Format	Description
RGB	32-bit, linear X8R8G8B8-formatted color bitmaps, in sRGB color space. To work with RGB data, specify NUI_IMAGE_TYPE_COLOR/NUI_IMAGE_TYPE_COLOR_YUV for C++ and RgbResolution640x480Fps30/RgbResolution1280x960Fps12/ YuvResolution640x480Fps15 for managed code when you

	open a color data stream.
YUV	16-bit, gamma-corrected linear UYVY-formatted color bitmaps, where the gamma correction in YUV space is equivalent to sRGB gamma in RGB space. Because the YUV stream uses 16 bits per pixel, this format uses less memory to hold bitmap data and allocates less buffer memory. To work with YUV data, specify NUI_IMAGE_TYPE_COLOR_RAW_YUV for C++ and RawYuvResolution640x480Fps15 for managed code when you open a color data stream. YUV data is available only at the 640x480 resolution and only at 15 fps.
Bayer	32-bit, linear X8R8G8B8-formatted color bitmaps, in sRGB color space. To work with Bayer data, specify NUI_IMAGE_TYPE_COLOR_RAW_BAYER for C++ and RawBayerResolution1280x960Fps12 or RawBayerResolution640x480Fps30 for managed code when you open a color data stream.

Tab I. Color data available formats [15]

We have used Color Frame class to display camera view to the screen. It contains all the image data and format. We have used RgbResolution640x480Fps30 as color format to achieve maximum output. It allows Infrared data (16 bits per pixel with the 6 least significant bits set to 0) whose resolution is 640 x 480 and frame rate is 30 frames per second. In the byte array returned for each frame, two bytes make up one pixel value. The bytes are in little-endian order, so for the first pixel, the first byte is the least significant byte (with the least significant 6 bits of this byte always set to zero), and the second byte is the most significant byte.A Writable Bitmap stores the pixel data which is stored in a pixel array and displays it through the source[2,3].

4.2 Tracking(Skeletal)

Skeletal tracking allows to recognize people and follow their actions. Using the infrared(IR) camera, Kinect can recognize up to six users in the field of view of the sensor. Of these, up to two users can be tracked in detail. An application can locate the joints of the tracked users in space and track their movements over time.

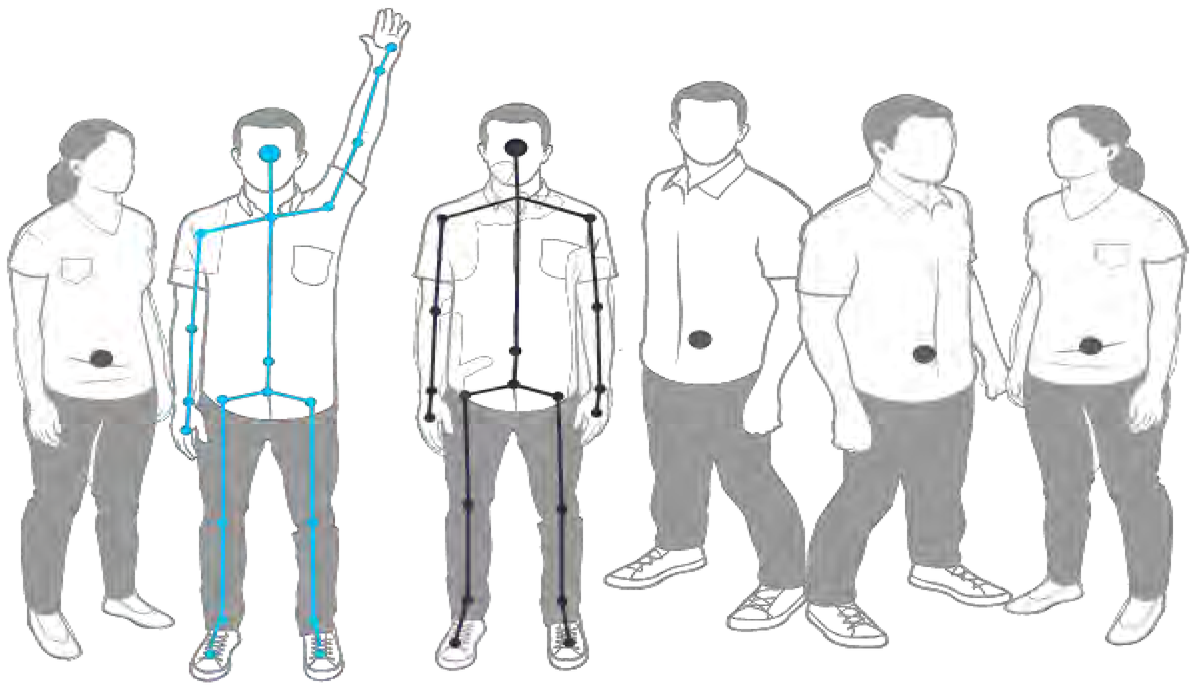


Fig 6. Kinect can recognize up to six people and track two [14]

Using the NUI API, we can track the head and the upper body of the user, no specific pose or calibration action needs to be taken for the user to be tracked.

In default range mode, Kinect can see people standing between 0.8 meters and 4.0 meters.

Using the Face Tracking SDK, together with the Kinect SDK, we can track human faces in real time. The Face Tracking SDK's face tracking engine analyzes input from a Kinect camera, deduces the head pose and facial expressions, and makes that information available to the application inreal time. FaceTrackFrame object we can detect the pitch and yaw of the head that is used for navigation [15].

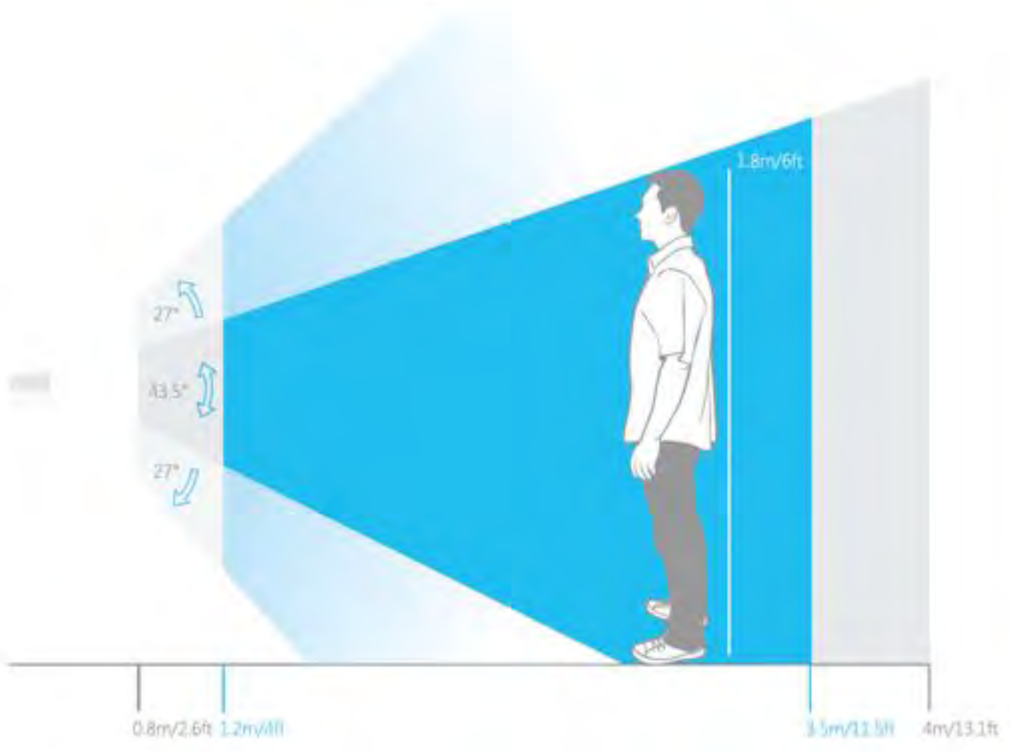


Fig 7. Kinect Vertical Field of View in default range [14]

We plan to use the FaceFrame object to detect face movements. The FaceRotationQuaternion property allows us to get the values of specific face pitching which will be used for the navigation of clothes. According to fig. 8, we will use the blink dividing the face into specific points and by implementing this equation:

Var distance = Math. Sqrt(Math.Pow(facePoints[10]. X – facePoints[35]. X , 2) +
Math. Pow(facePoints[10]. Y – facePoints[35]. Y , 2)) (1)

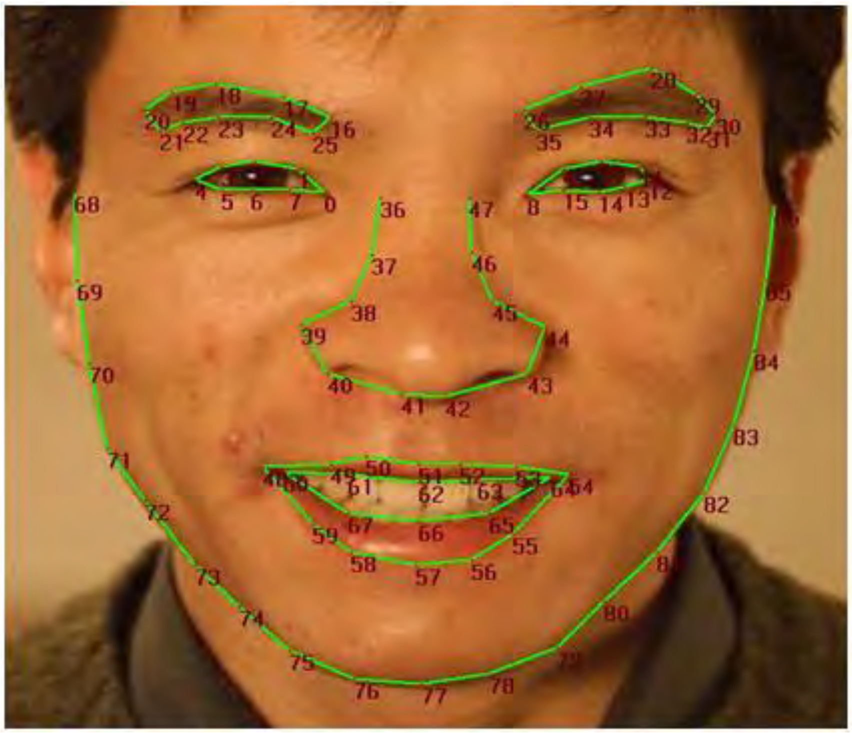


Fig 8. Camera Space using Face Tracking SDK[16]

In the new Kinect SDK 2.0, there is an ExpressionMouse for tracking blinks, which we plan to usein future implementations. This will be used to select from a set of clothes.

4.3 Amputee Extraction

Amputee extraction allows us to create an augmented reality environment to fold the cloth according to the user in real life situations. The Kinect for Windows SDK outputs a skeleton with 20 joints. In case of a missing limb it infers the limb to the screen. We are stopping the Kinect from inferring the limb and show it as it is.

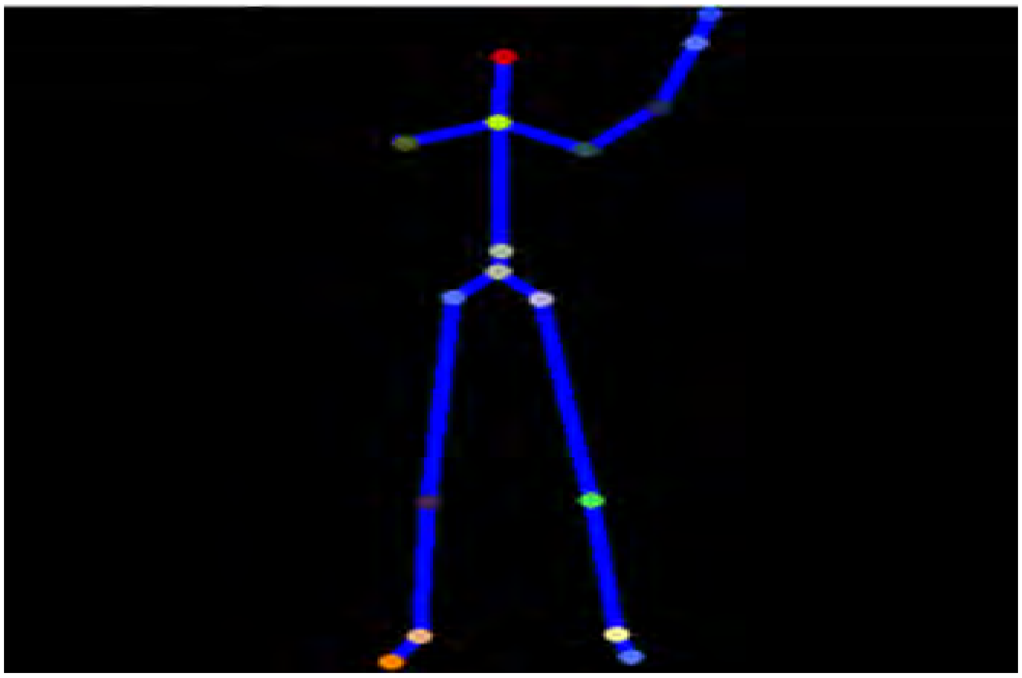


Fig 9. Skeleton Captured in the Kinect Camera

The NUI Skeleton provides full information on detailed position and orientation information up to two users. We will take the depth image and user ID provided by the Kinect SDK to detect the user and compare it to a preloaded model skeleton with help of the NUI Skeleton API to detect themissing limbs shown in Fig. 9 and 10, respectively. This will then be superimposed to the augmented version of the user's image to create a real life visualization of the user.

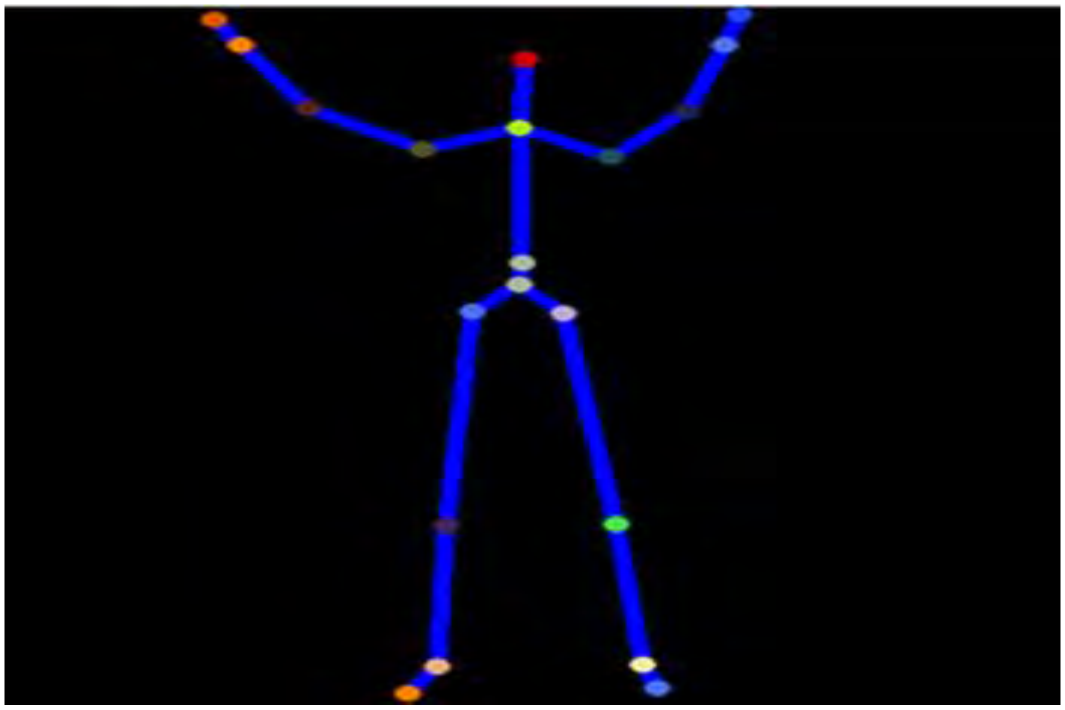


Fig 10. Model Skeleton

Comparing between these two images, the system would find out the missing limb. The GUI will contain a block where there would be information about the user's amputation. Initially, we have implemented the system for missing hands and arms.

4.4 Backgruond Removal

We used the Background Removal API to remove the background behind the user [17]. This would be important in big mall situations where the background can be distracting to test different apparels for the user.

The Background Removal API provides green screen capabilities for a single person. The tracked user is specified using their skeleton ID. This API uses various image processing techniques to improve the stability and accuracy of the player mask originally contained in each depth frame. The stream can be configured to select any single player as the foreground, and remove the rest of the color pixels from the sceneMicrosoft.Kinect.Toolkit.BackgroundRemoval.dll is a managed 64 bit DLL that we used in our system.

4.5 Database

In order to choose from a set of clothes, we have built a database for storing clothes. We have kept three types of apparel

- Shirts
- Trousers and
- T-shirts

We have created this database in MySQL and connected it to our existing C# code using "MySQL connector/NET" and "MySQL for Visual Studio". We have only built clothes for people having amputation on their arms or legs. Here is a schema for the database for our project.

Cloth table

ClothID	Name
1	T-shirt
2	Shirt
3	Trousers

Amputation table

AmputationID	AmputationName
1	No arms
2	No hands
3	Leg till knee
4	Full leg

ClothAttribute table

ClothAttributeID	Name
1	Size
2	Color

ClothAttributeValue Table

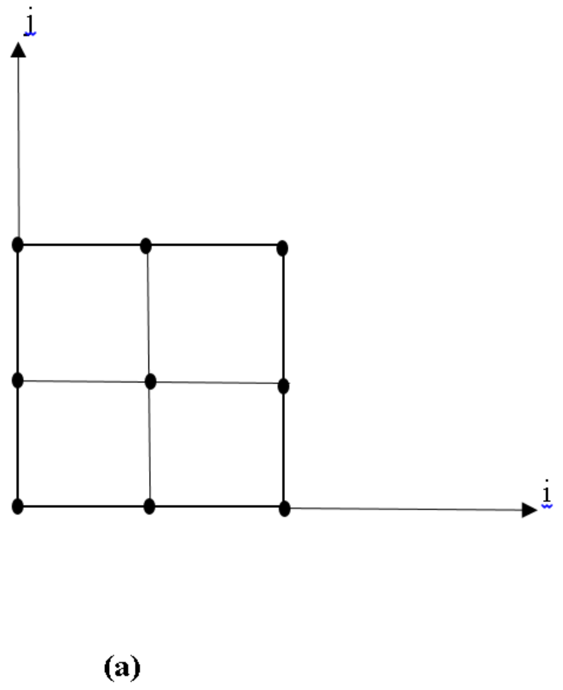
ClothAttributeValueID	AttributeID	Value
1	1	Small
2	1	Medium
3	1	Large
4	2	Green
5	2	Orange
6	2	Pink
7	2	Blue
8	2	Grey

Fig 11. Database Tables for the System

4.6 Creases and Folds

Creases and folds make the visualization realistic for people who have a missing limb. We have used Sakaguchi's version to achieve creases and folds on the body of a physically impaired [18]. Their model is called Party. The system uses a grid to represent the cloth, as shown in fig. 1. Their work is based on Newton's law of dynamics.

$$\frac{d^2r}{dt^2} = \frac{1}{m}(F_{ext} + F_{int}) \tag{2}$$



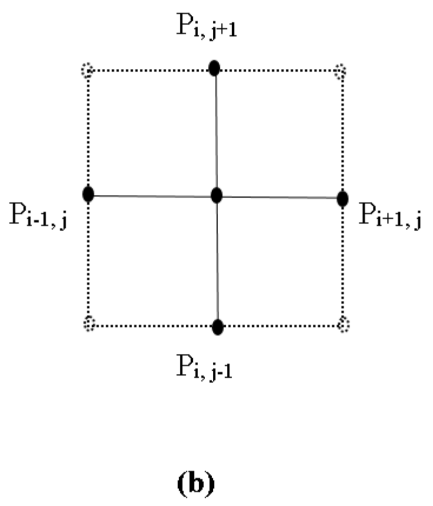


Fig 12. (a) Cloth represented as a grid and (b) Point $P_{i,j}$ with its four neighboring points

Where r is a point on the grid, m is the mass of the point, and F_{ext} and F_{int} are respectively the external and internal forces acting on the point.

The internal forces included are

$$F_{int} = F_{spring} + F_{visco} + F_{plas} \tag{3}$$

The internal forces components are according to the Terzopoulos' deformable model, where $F_{spring}$is the force due to the elongation of a spring, $F_{visco}$is the force due to viscosity and F_{plas} is the force due to plasticity.

Sakaguchi et al. used Euler's method to find out the velocity and position of the cloth. They also kept an eye on the interaction between the cloth and the underlying human body. Their model handled collision handling based on the conservation of momentum. Since the cloth and the human body are made of many vertices, collision checking would cost a lot of computational time. They increased the time using a finite bounding volume to decrease the amount of space involved in collision detection. They responded to collision detection by considering friction.

4.7 User Interface

The user interface had to be friendly as the subjects include people who may have no hands and may not interact with the computer like an average person. So, we made sure the user feels at home using this application. We have navigation by gesture and voice. Figure 13 represents the user interface that we have developed for our system.

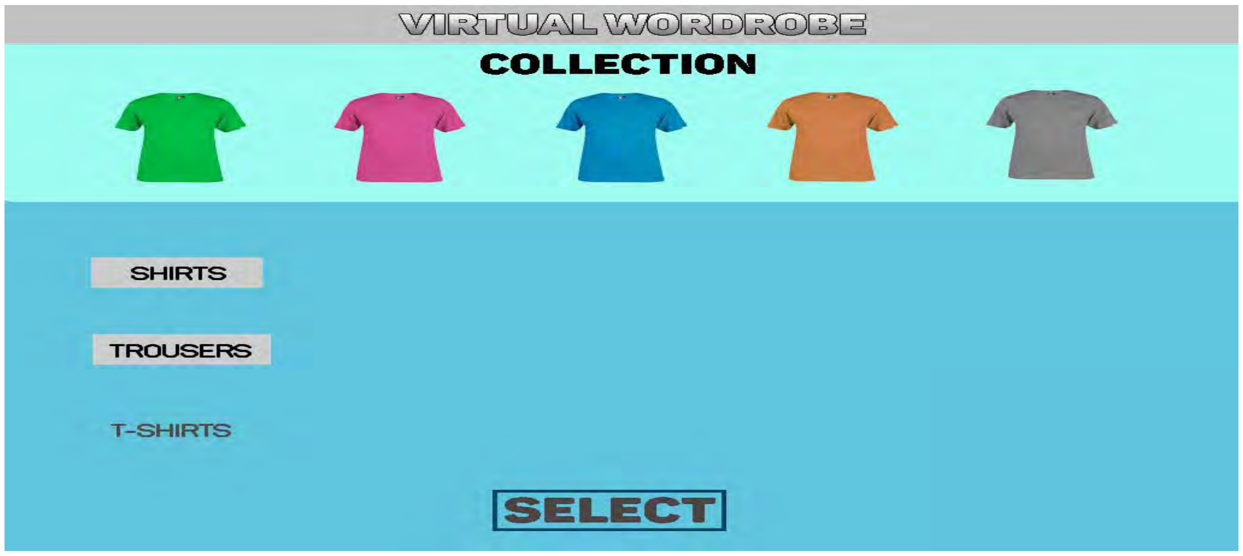


Fig 13. User Interface for the proposed system

4.7.1 Navigation by Gesture

We plan to use the FaceFrame object to detect face movements. The FaceRotationQuaternion property allows us to get the values of specific face pitching which will be used for the navigation of clothes. Using Equation (1), we would use the blink dividing the face into specific points:

This will be used to select from a set of clothes.

4.7.1 Navigation by Voice

Kinect also has a stereo microphone to enable chat and voice commands. It has a wide-field, conic audio capture.

Kinect for Xbox One has a set of preloaded voice commands for different operations. We plan to add our own custom commands implementing a custom voice recognition class using C#.

By default, the AdaptationOn flag in the speech engine is set to ON, which means that the speech engine is actively adapting its speech models to the current speaker(s). This can cause problems over time in noisy environments or where there are a great number of speakers. Therefore we will set the AdaptationOn flag OFF in our application [6].

We will be using the RcognizerInfo method for voice input creating a speech recognition engine which will be an alternate of head gestures for navigation.

4.8 Model Scaling

In order to display the appropriate size for the user, we calculated the distance of two shoulder joints to find the preferred size of the user. We have used Kinect's Skeleton position method to find the distance between the two shoulder joints.

Kinect sensor maps the skeleton in three coordinates. Monir et al. [12] has kept the Z coordinate fixed to calculate the X and Y coordinates.

$$W = \frac{Y_2 - Y_1}{X_2 - X_1} \tag{4}$$

$$W_r = \frac{Y_{ls} - Y_{cs}}{X_{ls} - X_{cs}} \tag{5}$$

$$W_l = \frac{Y_{ls} - Y_{cs}}{X_{ls} - X_{cs}} \tag{6}$$

Here,

W = Distance of two coordinate points

W_1 = Distance of left shoulder coordinate point

W_2 = Distance of right shoulder coordinate point

Y_2= next value of y-axis and Y1= previous value of y-axis

X_2= next value of x-axis and X1= previous value of x-axis

Y_{ls}= left shoulder y-axis value and Yrs= right shoulder y-axis value

X_{ls}= left shoulder x-axis value and Xrs= right shoulder x-axis value

Y_{cs}= center shoulder y-axis value and Xcs= center shoulder x-axis value

Using (4), (5) and (6) they have calculated the measurement of right shoulder by calculating the coordinate distance between right Shoulder and center Shoulder. Then the measurement of left shoulder has been determined by calculating the coordinate distance between left Shoulder and center Shoulder. [7]

CHAPTER 5

RESULTS AND DISCUSSIONS

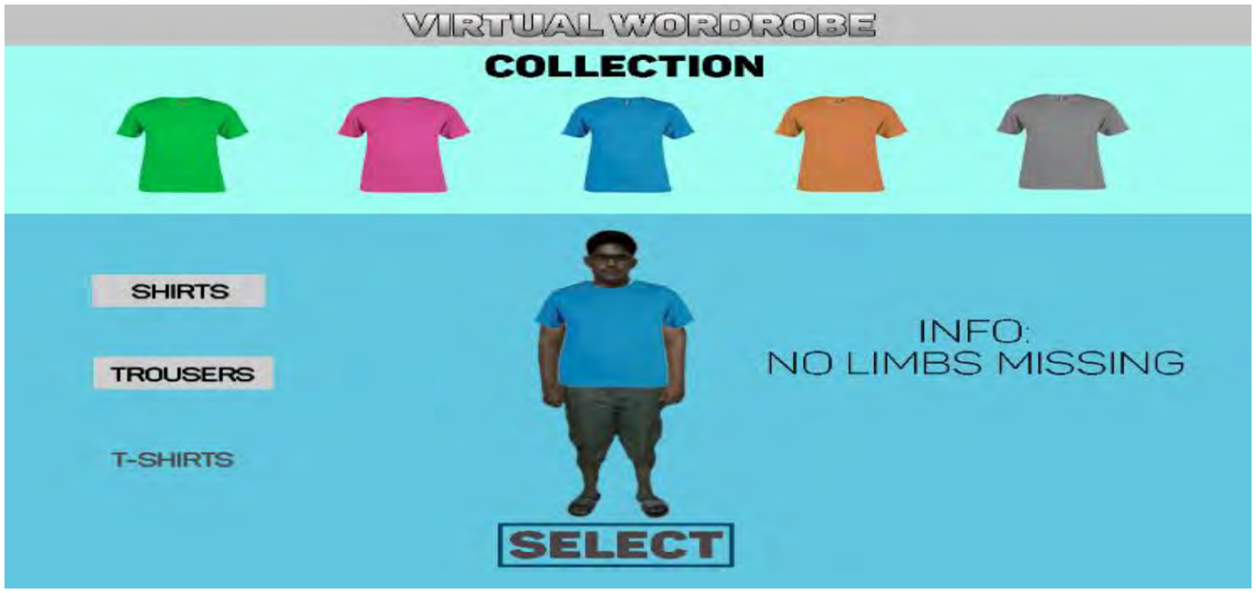
5.1 Results and Output

To get application towards real life implementation, the subjects were told to stand in front of the Kinect sensor.

- Minimum distance of 1.8 meters(5 feet)
- Maximum distance of 3.5 meters(11.5 feet)

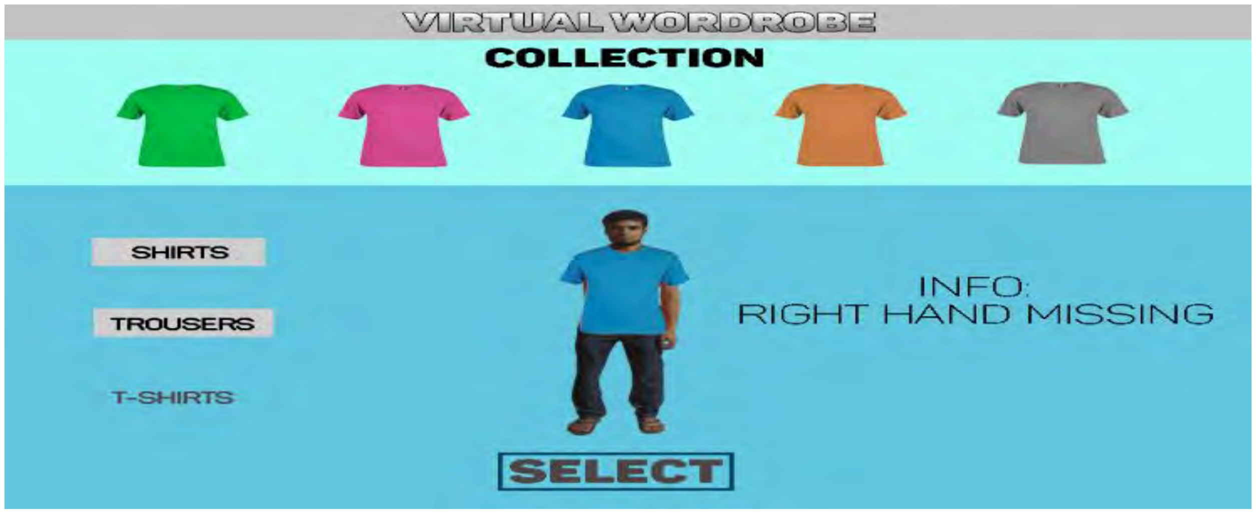
After detecting the depth points, they could able to navigate the given items with their head pitches and select by blinking. Four sample subjects are demonstrated in Fig. 14, 15, 16 and 17 respectively.

The first subject has no disabilities. The system also recognizes him for no missing limbs. Next he can select clothes navigating with head pitches left or right and select by blinking. The next test result shows the subject who has no right hands. The system identifies him and shows him options of t-shirts from which he can choose. The next subject has no right leg. The system identifies his amputation and provides a number of trousers to choose from. The last example shows the options for shirts. The subject has no right hands. His amputee was identified by the system and he was shown a number of shirts from which he can choose from head pitches and select by blinking.



INFO:
NO LIMBS MISSING

Fig 14. User with no missing limbs



INFO:
RIGHT HAND MISSING

Fig 15. User with right hand missing

further performance monitoring.

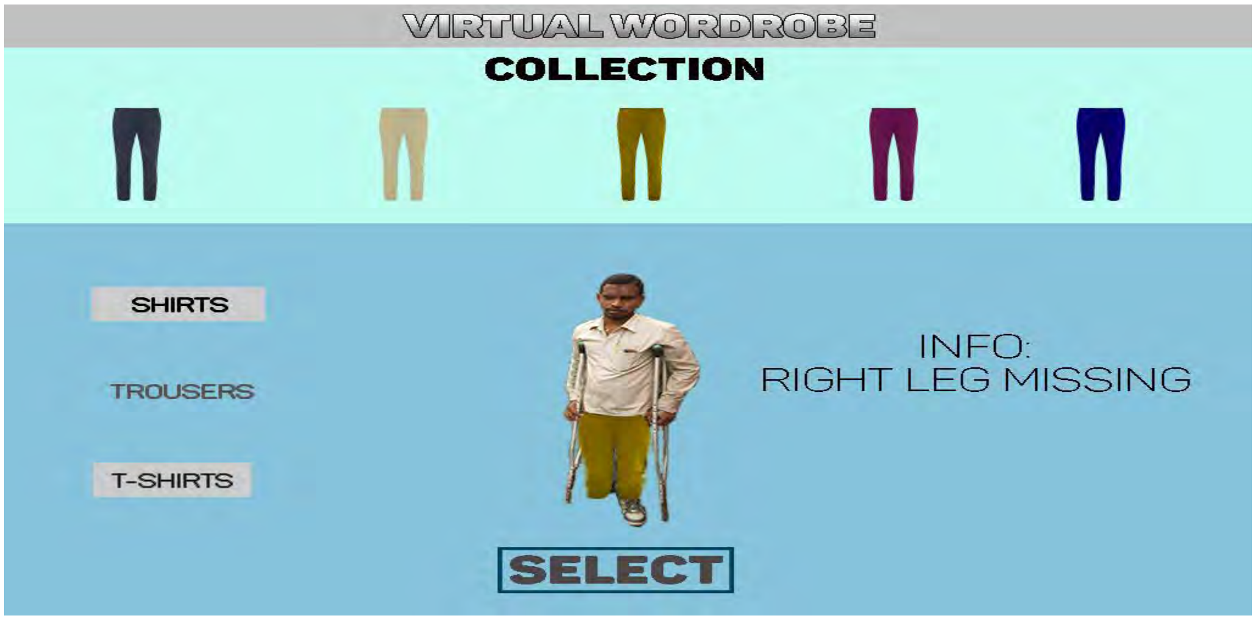


Fig 16. User with right leg missing

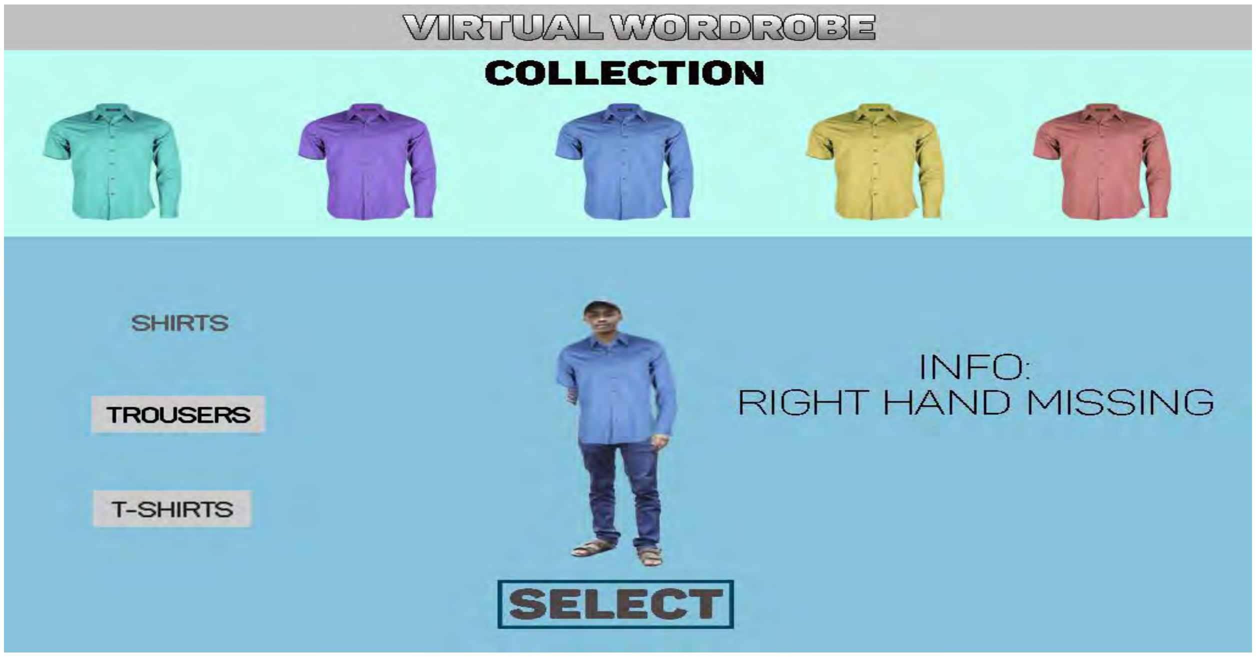


Fig 17. User with right hand missing

Table II shows the recorded names, age, preferred sizes and disabilities of the test subjects for further performance monitoring.

No.	Name	Age	Kinect Value	Size	Disability
1.	Shihab	23	624	Large	None
2.	Asgar	31	594	Medium	Right hand missing
3.	Shohid Mia	46	563	Small	Left arm missing
4	Chamily Begum	38	576	Medium	Right arm missing
5.	Md. Azim	50	584	Medium	Left arm missing
6.	Ms. Lily	26	553	Small	Left arm missing
7.	Ismail Hossain	41	617	Large	Right hand missing
8.	Shobuj	32	559	Small	Right leg missing
9.	Nusrat	21	557	Small	Left leg missing
10.	Intekhab Alam	23	582	Medium	Right hand missing

Tab II. Comparison between test subjects

5.2 Performance

As our system deals with extracting amputees and imposing apparel on people with disabilities, we evaluated the performance of our system on detecting amputee of people with different disabilities. We got an idea of the performance from the result of the ground truth model and the model we constructed. This criteria can be defined as,

$$p = \frac{Ac \cap Ag}{Ac \cup Ag} \tag{7}$$

Here Ac is the constructed model's area and Ag is the area of our proposed modelin terms of the number of pixels. A set of people with different disabilities were used as test subjects to determine the performance of the system. We represent the performance of our system on the basis of a graph imposing the width of the ground model and our proposed model. For ground truth model we have used the shoulder to feet distance of the subjects. We have compared it with the shoulder to feet distance of our system to determine the performance. The area graph below represents the results with the values of the ground truth model on the X axis and the values of our system on the Y axis.

LEGEND

▬ Shoulder to feet ratio for ground truth model

▬ Shoulder to feet ratio for our proposed mode

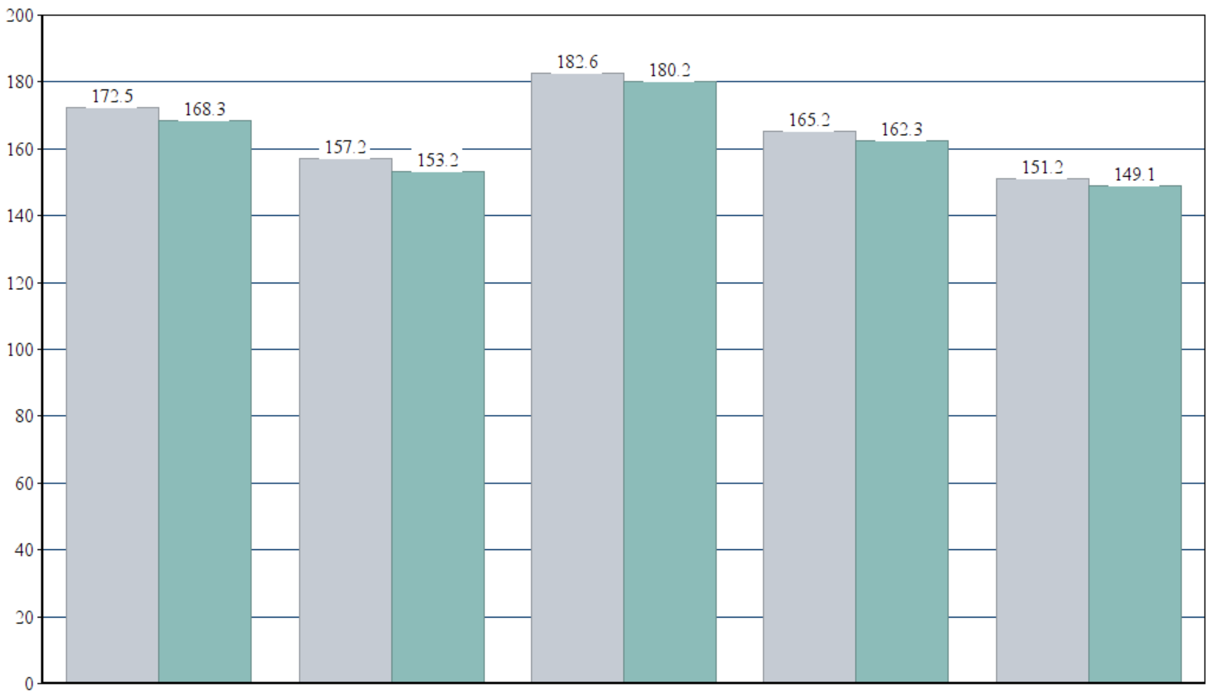


Fig 18. Shoulder to feet height comparison between ground truth model and our proposed model

Figure 18 shows the representation of the shoulder to feet height of the ground truth model with respect to the shoudler to feet height of our proposed model. We have shown the comparison between five test subjects here. The accuracy of these tests are represented below by Tab. III.

Shoulder to Feet Length for Ground Truth Model (cm)	**Shoulder to Feet Length for Our proposed Model (cm)**	**Accuracy (%)**
172.5	168.3	97.57
157.2	153.2	97.46
162.6	160.2	98.52
148.2	142.3	96.02
151.2	149.1	98.61

Tab III. Comparison between test subjects

We have also calculated the accuracy based on the values we get from the comparison between the ground truth model shoulder to feet height and the heights we get from our proposed system. Tab. IIIshows the accuracy for each experiments. The average accuracy we achieve is 97.636%.

CHAPTER 6

CONCLUSIONS

6.1 Conclusion

This system will prove very useful for people with physical disabilities. It will be very helpful for disable people to try. It uses the best of all the previous works to implement a system that is both convenient for both people with disabilities and regular people. The technology used is also state of the art and cheap which makes it easy to implement with minimum production cost. The real time implementation makes it possible for the users to select the best fit for themselves. It will also provide a fast, efficient and new way for them to trial different apparels which has long been a big problem for people with amputations. This system also has great market value for people without amputations. A real time virtual mirror would save lots of time to try out clothes. In future, we would like to implement this system within the reach of the mass where people can use this even from their smartphones.

6.2 Future works

There are many possible implementations for this system in the future. We would like to introduce this system in the market to make it available for anyone, let alone people with disabilities. Currently, there are no such system in the market for anyone. People need to try on clothes manually. This requires a lot of time and is inefficient. This system would solve this problem by saving time and helping people with disabilities to try clothes without less hassle. Some of our plans regarding the system is as follows:

- We have used a somewhat 2D model. In the future, we plan to use images from different angles to achieve a more realistic video stream. This also helps in implementing a somewhat 3D view of the system.
- The current system uses RgbResolution640x480Fps30 as color format. We intend to improve the resolution in the future using a higher resolution format. This would need an additional camera along with the Kinect to achieve a better resolution that is necessary in the 4k modern world.
- This system now works with a Kinect device connected to a camera. In the future, we plan to bring it to the internet. Later, we intend to make a version available for smartphones as well.

REFERENCES

[1] Australian National University, “Different Types of Disabilities”, retrieved from https://services.anu.edu.au/human-resources/respect-inclusion/different-types-of-disabilities

[2] Cordier, F. and Magnenat-Thalmann, N. (2005), A Data-Driven Approach for Real-Time Clothes Simulation†. Computer Graphics Forum, 24: 173–183. doi:10.1111/j.1467-8659.2005.00841.xMicrosoft, “Kinect for Windows SDK,” Available: https://msdn.microsoft.com/en-us/library/hh855347.aspx

[3] M. Yuwei, P.Y Mok, J. Xiaogang.*Interactive Virtual Try-on Clothing Design Systems[J]*, Journal Computer-Aided Design, 2010,42(4):310-321.

[4] C. –S. Cho, J. –Y. *An implementation of a garment-fitting simulation system using laser scanned 3D body data*. Computers in Industry(2010).

[5] M. W. Lee & R. Nevatia (2007, February). *Body Part Detection for Human Pose Estimation and Tracking*. Proceedings of the IEEE Workshop on Motion and VideoComputing.

[6] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, & D. Fox, (2011, September). *RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments*. Proceedings of the 13th international conference on Ubiquitous computing, Beijing, China.

[7] Y. Wu, T. S. Huang, “Vision-Based Gesture Recognition: A Review”, University of Illionois at Urbana-Champaign(2001).

[8] M. Bhuiyan, R. Piching, "A Gesture Controlled User Interface for Inclusive Design andEvaluativeStudy of Its Usability,” Journal of Software Engineering and Applications, Vol.4 No.9(2011).

[9] L. Ziquan, S. Zhao, “Augmented Reality: Virtual fitting room using Kinect,” Department of Computer Science, School of Computing, National University of Singapore, (2011).

[10] F. Isikdogan , G. Kara “A Real Time Virtual Dressing Room Application using Kinect,” Bogazici University, (2012).

[11] A.B. Habib, A. Asad, W.B. Omar, “Magic Mirror Using Kinect,” BRAC University (2015).

[12] Md. M. Monir, Md. N. H. Siddique, A. P. Tanni, Md. N. E. Hashem, “Design and Implementation of a Magic Mirror Using Kinect,” BRAC University(2016).

[13] Microsoft, “Kinect for Windows SDK,” Available:https://msdn.microsoft.com/en-us/library/hh855347.aspx

[14] Tracking Users with Kinect Skeletal Tracking, “Microsoft MSDN,” retrieved from https://msdn.microsoft.com/en-us/library/jj131025.aspx

[15] Color Stream, retrieved from https://msdn.microsoft.com/en-us/library/jj131027.aspx

[16] Face Tracking, “Microsoft MSDN,” retrieved from https://msdn.microsoft.com/en-us/library/jj130970.aspx

[17] Background Removal, “Microsoft MSDN,” retrieved from https://msdn.microsoft.com/en-us/library/dn435663.aspx

[18] Sakaguchi, Y., Minoh, M., & Ikeda, K. (1995). Party: A numerical calculation method for a dynamically deformable cloth model. Systems and computers in Japan, 26(8), 75-87.