BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NGÀNH KHOA HỌC MÁY TÍNH

Đề tài

XÂY DỰNG PHÒNG THAY ĐỒ THỰC TẾ TĂNG CƯỜNG BẰNG CÔNG NGHỆ NHẬN DẠNG CỬ CHỈ SỬ DỤNG CAMERA KINECT 2

Phân hệ nhận dạng giọng nói

Sinh viên thực hiện:

Trần Hoàng Thảo Nguyên

Mã số: B1509938

Khóa: 41

Cần Thơ, 12/2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BỘ MÔN KHOA HỌC MÁY TÍNH

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NGÀNH KHOA HỌC MÁY TÍNH

Đề tài

XÂY DỰNG PHÒNG THAY ĐỒ THỰC TẾ TĂNG CƯỜNG BẰNG CÔNG NGHỆ NHẬN DẠNG CỬ CHỈ SỬ DỤNG CAMERA KINECT 2

Phân hệ nhận dạng giọng nói

Giáo viên hướng dẫn:
ThS. Phạm Nguyên Hoàng

Sinh viên thực hiện:
Trần Hoàng Thảo Nguyên
Mã số: B1509938
Khóa: 41

Cần Thơ, 12/2019

NHẬN XÉT CỦA GIẢNG VIÊN

Báo cáo luận văn đã được chỉnh sửa theo yêu cầu của Hội đồng châm luận văn

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Cần Thơ, ngày 17 tháng 12 năm 2019

Giảng viên

Phạm Nguyên Hoàng

(Kí tên)

LỜI CẢM ƠN

Để hoàn thành tốt luận văn này, em đã nhận được sự giúp đỡ, hỗ trợ và động viên của rất nhiều cá nhân cũng như tập thể. Nhân đây, em xin tỏ lòng biết ơn đến gia đình đã quan tâm, ủng hộ về vật chất lẫn tinh thần trong quá trình tôi học tập và thực hiện luận văn.

Xin chân thành biết ơn sâu sắc đến thầy Phạm Nguyên Hoàng đã rất tận tình hướng dẫn, góp ý, truyền đạt những kiến thức và kinh nghiệm quý báu để em hoàn thành tốt luận văn này.

Đồng thời em cũng chân thành cảm ơn người bạn đã đồng hành cùng em thực hiện đề tài luận văn này - Bạn Trương Gia Huy.

Chân thành cảm ơn quý thầy cô bộ môn Khoa học máy tính - Đại học Cần Thơ đã tận tình chỉ bảo, giúp đỡ trong thời gian em thực hiện luận văn. Cuối cùng em xin cảm ơn bạn bè, anh chị khoa Công nghệ thông tin và Truyền thông đã cho em ý tưởng, hỗ trợ, động viên em trong quá trình thực hiện luận văn. Mặc dù cố gắng hoàn thiện luận văn này nhưng do kiến thức còn hạn chế nên không thể tránh những sai sót, vì thế em mong nhận được những góp ý của thầy cô và các bạn.

Trần Hoàng Thảo Nguyên

MỤC LỤC

DANH SÁCH THUẬT NGỮ TIẾNG ANH

STT	Thuật ngữ	Ý nghĩa
1	API(Application Programming Interface)	Giao diện lập trình ứng dụng
2	Augmented Reality	Thực tế tăng cường
3	Virtual Reality	Thực tế ảo
4	Computer Vision	Thị giác máy tính
5	Skeleton Detection	Nhận dạng khung xương
6	Skeleton Tracking	Quá trình nhận dạng khung xương và hiển thị khung xương theo vị trí của người dùng
7	Voice Recognition	Nhận dạng giọng nói
8	Gesture Recognition	Nhận dạng cử chỉ
9	UI (User Interface)	Giao diện người dùng

DANH SÁCH HÌNH

DANH SÁCH BẢNG

TÓM TẮT

Mua sắm, đặc biệt là quần áo luôn là một trong những nhu cầu thiết yếu của con người. Ngày nay, nhờ vào sự phát triển của khoa học kỹ thuật, việc mua sắm có thể được thực hiện dưới hai hình thức: Mua sắm ở cửa hàng và online. Khi mua trực tiếp, ta có thể dễ dàng chọn lựa và mặc thử, tuy nhiên khi cửa hàng quá đông, khách hàng phải chờ đợi rất lâu để được thử quần áo. Ngược lại, khi mua quần áo online, khách hàng có thể chọn lựa vào mua một cách nhanh chóng, tuy nhiên lại không thể mặc thử quần áo lên người.

Dựa vào thuận lợi và bất lợi này của hai hình thức mua sắm, em phát triển một **Phòng Thử Quần Áo Thực Tế Tăng Cường** để khách hàng có thể ướm thử các mẫu áo quần mà không cần phải đi vào cửa hàng cũng như chờ đợi lâu.

Hướng tiếp cận của em là sử dụng Camera Kinect Microsoft để đọc vào các data về chiều sâu, chuyển động của con người và sử dụng Unity cùng các thuật toán cần thiết để áp các mẫu quần áo 3D lên người và để xây dựng giao diện người dùng. Đồng thời, em tích hợp cả bộ nhận dạng giọng nói của Microsoft để người dùng có thể điều khiển được hệ thống thông qua việc ra lệnh bằng ngôn ngữ.

Từ khóa: Kinect Camera, nhận dạng giọng nói, nhận dạng khung xương, nhận dạng cử chỉ, thực tế tăng cường.

ABSTRACT

Shopping for clothes is one of the most basic need of human. Nowadays, thanks to the development of technology, shopping can be done both in brick-and-mortal stores and on the Internet. However, shopping offline could be frustrated because customers need to wait for using the fitting room and payment. On the other hand, online shopping is very fast and convenient, but customers can not try on the clothes in order to know if the clothes fit in their bodies or not.

To tackle shopping problems mentioned above, we have decided to implement a **Virtual Fitting Room** which allows users to "virtually" try on clothe models via a screen without going to an actual store.

Our approaching method is using Kinect Camera and Unity. Kinect Camera can provide data streams of depth image, human movement information, audio voice and also a skeleton API; while Unity is used for developing UI and to display the 3D models onto human's body. In addition, Microsoft Voice Recognition Toolkit is also embedded, so users can interact with and control the system via both gestures and verbal statements.

Keywords: Kinect Camera, voice recognition, skeleton detection, gesture recognition, augmented reality.

PHẦN GIỚI THIỆU

1. Đặt vấn đề

Việc mua sắm, đặc biệt là mua sắm quần áo là một trong những nhu cầu thiết yếu của con người. Ngày nay, với sự phát triển của khoa học kỹ thuật, ta có thể mua sắm bằng rất nhiều cách khác nhau. Trong số đó, có hai cách thức mua sắm phổ biết nhất là: mua hàng trực tiếp ở cửa hàng, mua hàng thông qua các trang web thương mại trực tuyến (mua online). Sau quá trình quan sát, em nhận thấy cả hai phương pháp này đều có những thuận lợi và hạn chế riêng. Trong đề tài này, em sẽ đề cập đến việc mua sắm quần áo.

Với trường hợp mua sắm trực tiếp tại cửa hàng, người mua có thể dễ dàng xem được chất liệu của quần áo, và thử quần áo tại cửa hàng để quyết định xem mẫu quần áo có phù hợp với vóc dáng, hay vừa với mình hay không. Tuy nhiên, khi cửa hàng có quá nhiều khách hàng, người mua có thể phải chờ đợi rất lâu để sử dụng phòng thử quần áo và để tính tiền. Việc chờ đợi này có thể dẫn đến khách hàng bị mất kiên nhẫn và bỏ đi, không còn muốn mua hàng nữa.

Với trường hợp mua sắm thông qua các trang thương mại trực tiếp, trái lại, khách hàng không cần phải chờ đợi. Mọi quá trình từ chọn lựa đến thanh toán đều được thực hiện nhanh chóng qua Internet. Tuy nhiên, khách hàng không thể thực sự mặc thử quần áo mà phải căn cứ vào hướng dẫn số đo hay ước lượng dựa vào số đo người mẫu. Quá trình này khá phức tạp và không chính sác, dẫn đến việc nhiều trường hợp hàng hóa mua trên mạng không vừa, không đúng kì vọng của người mua. Điều này có thể ảnh hưởng xấu đến cả tâm lý khách hàng và uy tín thương hiệu.

Vì các lý do kể trên nhóm em quyết định thực hiện đề tài "PHÒNG THAY ĐỒ THỰC TẾ TĂNG CƯỜNG SỬ DỤNG CAMERA KINECT 2". Đề tài này sử dụng các Camera Kinect và bộ phát triển ứng dụng của Microsoft, đi kèm với bộ phát triển game từ Unity để tạo ra một phòng thử quần áo cho phép người dùng có thể ướm lên những mẫu quần áo 3D và xem kết quả thông qua màn hình, điều khiển bằng cử chỉ và giọng nói.

2. Mục tiêu đề tài

Xây dựng hệ thống phòng thử quần áo tăng cường có chức năng mô phỏng trải nghiệm thử quần áo thật, tức hiển thị các mẫu quần áo lên cơ thể người sử dụng, đồng thời tương tác với hệ thống thông qua một màn hình.

Các dữ liệu đầu vào được thu từ Kinect camera để nhận dạng vị trí, dáng đứng, cử chỉ và giọng nói của người sử dụng. Bằng việc sử dụng camera Kinect, hệ thống không cần thêm bất cứ thiết bị đầu vào nào khác, chính vì vậy, sự tương tác giữa hệ thống và người sử dụng sẽ trở nên tự nhiên và tiện lợi hơn.

Trong hệ thống Phòng thử quần áo tăng cường, màn hình không được thiết kế và sử dụng như một User Interface bình thường. Màn hình cần phải có một khoảng trống ở giữa đủ rộng để hiển thị hình ảnh gương của người dùng (user's mirror image), và phần còn lại của màn hình ở hai bên được dùng để hiển thị các thông tin, nút chức năng, mẫu quần áo. Chi tiết về User Interface sẽ được mô tả sau.

Ngoài ra, việc tương tác trong hệ thống cũng phải được đảm bảo. Hệ thống tương tác thông qua hai nguồn chính: cử chỉ và giọng nói. Để đảm bảo cho việc tương tác xảy ra mượt, chính xác, việc lập trình cần chú ý giảm thiểu hai chỉ số là false positive và false negative. Ví dụ, sẽ có một vài cử chỉ và cụm từ nhất định được lập trình để làm lệnh điều khiển. Khi người dùng chủ ý sử dụng cử chỉ/ cụm từ đó để điều khiển, hệ thống phải nhận ra được và phản hồi đúng với lệnh này. Tuy nhiên, trong một số trường hợp, khi người dùng vô tình tạo ra/ nói ra cử chỉ/ cụm từ tương tự với các lệnh điều khiển, hệ thống cũng có thể nhận những cử chỉ đó là các cử chỉ điều khiển. Vì vậy, hệ thống này cần được lập trình kỹ lưỡng để giảm thiểu trường hợp nêu trên.

3. Lịch sử giải quyết vấn đề

Vì đồ án này bao gồm sự nghiên cứu từ nhiều nền tảng kiến thức khác nhau, nên em sẽ trình bày những nghiên cứu liên quan thành 5 phần: Thử quần áo ảo (virtual try-on), thị giác máy tính (computer vision), điều khiển bằng cử chỉ (gesture control), điều khiển bằng giọng nói (verbal control).

3.1. Thử quần áo ảo (Virtual Try-on)

Có rất nhiều hệ thống Virtual Trying tiếp cận theo hướng mô hình 3D, với nhiều mục đích khác nhau. Trong bài báo "A 3D Virtual Show Room for Online Apparel Retail Shop"[1], các nhà nghiên cứu tìm cách dựng mô hình 3D cho quần áo từ một ảnh chụp 2D. Một số nhà nghiên cứu khác lại tìm cách để cải thiện mô hình quần áo và mô hình người, để hai đối tượng này trông thật hơn và có thể tương tác với nhau, cụ thể kết quả được trình bày trong bài báo "3D Interactive Clothing Animation"[2]. Ngoài ra, còn có một vài nghiên cứu xây dựng hệ thống thử quần áo cho một mô hình 3D của người sử dụng, tức là áp mẫu quần áo lên mô hình người như bài báo "Virtual Dressing Room Application with Virtual Human"[3].

3.2. Thị giác máy tính (Computer Vision)

Trước khi camera Kinect được phát minh, một hoặc nhiều camera được sử dụng kết hợp với nhau trong các hệ thống thị giác máy tính (Poppe .R)[4], sử dụng thị giác máy tính để track chuyển động của cơ thể con người cũng đã và đang được nghiên cứu từ năm 1997 cho đến ngày nay (Wren, Azarbayejani, Trevor, & Pentland)[5]. Có rất nhiều nhà khoa học tìm cách tạo ra hoặc cải tiến thuật toán thị giác máy tính để máy tính có thể nhận dạng được cơ thể con người một cách tốt hơn (Lee & Nevatia)[6], trong khi đó một vài nhà khoa học khác đã giới thiệu các hệ thống nhận dạng hình ảnh chiều sâu (depth value image) tương tự với Kinect (Du, et al.)[7].

3.3. Điều khiển bằng cử chỉ (Gesture Control)

Một số nghiên cứu về nhận dạng cử chỉ của nhiều ngón tay (multi-finger hands gesture) đã được thực hiện (Malik & Laszlo)[8] và cả nhận dạng một chuỗi các cử chỉ của cơ thể người (sequence of gesture) (Kim & Kim)[9] .Ngoài ra, một số dự án khác nghiên cứu cách sử dụng cử chỉ để điều khiển mô hình người ảo trong một không gian ảo (Huang & Kallmann)[10]. Mục đích chính của mô hình này là định nghĩa một tập hợp các cử chỉ dễ học cho tất cả mọi người (bao gồm người khuyết tật và người già) để họ có thể điều khiển được các user interface (Bhuiyan & Picking, Gesture Controlled User Interfaces for elderly and Disabled)[11].

3.4. Nhận dạng bằng giọng nói

Hiện nay, có khá nhiều nghiên cứu được thực hiện để tạo ra các hệ thống có thể được điều khiển bằng giọng nói như nhận dạng và chuyển từ thành giọng nói (text-to-speech) (Das, Prerana)[12]. Một vài nghiên cứu khác tập trung vào việc sử dụng giọng nói để điều khiển trong các hệ thống nhà thông minh (Ali, Awa) [13]. Trong những năm gần đây, việc sử dụng Speech Recognition API cũng khá phổ biến và được sử dụng trong một số nghiên cứu như “Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx)"[14] - một nghiên cứu thực hiện so sánh các speech API của các công ty công nghệ lớn trên thế giới.

4. Đối tượng và phạm vi nghiên cứu

4.1. Đối tượng nghiên cứu

Dựa theo những trình bày ở các mục trước, đối tượng nghiên cứu của đề tài được xác định gồm 4 thành phần như sau:

- Camera Kinect và lập trình với Kinect: nghiên cứu, tìm hiểu về camera Kinect, nguyên lý hoạt động của camera và cách làm việc với camera. Cụ thể, phần này bao gồm cấu tạo của camera, đặc điểm của các cảm biến mà kinect có, cách sử dụng bộ phát triên ứng dụng của Microsoft đi kèm với Kinect camera.

- Điều khiển với cử chỉ (gesture control): nghiên cứu, tìm hiểu cơ sở lý thuyết về đồ họa, cách nhận dạng cử chỉ của camera Kinect. Cụ thể là đối với dữ liệu nhận được từ cảm biến, thông qua lập trình, quá trình nhận dạng sẽ được diễn ra như thế nào.

- Điều khiển với giọng nói (voice control): nghiên cứu, tìm hiểu cơ sở lý thuyết về audio streams, cách nhận dạng được giọng nói, cách trích xuất được audio streams để đưa vào quá trình xử lý. Tiếp theo là sử dụng dữ liệu đã trích xuất để điều khiển hệ thống theo yêu cầu.

4.2. Phạm vi nghiên cứu

Lập trình ứng dụng với Kinect và Unity có thể ứng dụng vào nhiều lĩnh vực khác nhau với những yêu cầu khác nhau. Trong phạm vi đề tài này, em sẽ tập trung vào lĩnh vực nhận diện cử chỉ, nhận diện giọng nói và xây dựng không gian thử đồ 3D trong Unity.

Sau khi tìm hiểu những sơ lược cơ sở lý thuyết và các đề tài liên quan đã được thực hiện, phạm vi nghiên cứu của đề tài sẽ xoay quanh các công việc sau:

- Nghiên cứu các cảm biến có trong camera Kinect và bộ công cụ phát triển phần mềm với Kinect do Microsoft cung cấp để nhận dạng cử chỉ và nhận dạng giọng nói (Cả hai thực hiện).
- Nghiên cứu lĩnh vực đồ họa máy tính để hiểu về hệ tọa độ không gian ba chiều được sử dụng trong Unity 3D, thông qua đó tính toán các thông số (Cả hai thực hiện).
- Tìm hiểu về Unity, các thuộc tính, các thành phần của chúng để tạo giao diện người dùng, cũng cách nhập và quản lý các mẫu quần áo 3D (Cả hai thực hiện).
- Tìm hiểu về cách nhận diện cử chỉ bằng camera Kinect và ứng dụng nó vào phần mềm (do Trương Gia Huy thực hiện).
- Tìm hiểu về cách nhận diện giọng nói và ứng dụng nó vào phần mềm (do Trần Hoàng Thảo Nguyên - em thực hiện)

5. Phương pháp nghiên cứu

5.1. Về lý thuyết

Đề tài này cần tổng hợp kiến thức từ lĩnh vực đồ họa máy tính, về Unity và camera Kinect. Bên cạnh đó là ngôn ngữ lập trình C#, là ngôn ngữ lập trình được dùng trong bộ công cụ phát triển phần mềm dành cho camera Kinect của Microsoft. Do đó, về mặt lý thuyết có những việc sau:

- Tìm hiểu về Unity, thông qua các bài viết/video hướng dẫn.
- Tìm hiểu về camera Microsoft Kinect và bộ công cụ phát triển phần mềm của nó.

- Tìm hiểu về ngôn ngữ C# và cách sử dụng các hàm được định nghĩa sẵn trong bộ công cụ Kinect.
- Tìm hiểu, nghiên cứu các đề tài có sử dụng camera Kinect trước đây để tìm ra hướng đi phù hợp bằng cách đọc các bài báo, luận văn trước đó.
- Thảo luận nhóm với bạn thực hiện chung đề tài mỗi tuần để giải quyết những vấn đề khó khăn gặp phải.
- Thảo luận, tham khảo ý kiến của giáo viên cố vấn về những khó khăn khi thực hiện đề tài.

5.2. Về thực hành

- Tạo những project đơn giản để làm quen với cách hoạt động, cũng như hệ tọa độ và các thành phần khác trong Unity
- Tìm các mẫu quần áo 3D có sẵn và tiến hành chỉnh sửa để phù hợp với hệ tọa độ của Unity. Sau đó, nhập những mẫu quần áo 3D vào Unity và định nghĩa cách quản lý chúng.
- Tìm các mẫu hoa văn dành cho áo, tạo các màu đơn sắc nhằm giúp người dùng có thể tùy chọn được loại hoa văn/màu sắc họ mong muốn trên cùng một mẫu quần áo.
- Tiến hành lập trình bằng ngôn ngữ C#.
- Kiểm thử xem hệ thống đã thực hiện đúng chức năng yêu cầu chưa.

Như vậy, phương pháp nghiên cứu của đề tài là đi từ thực tiễn => giải pháp và để đưa ra được giải pháp tối ưu nhất, em thực hiện nghiên cứu sâu về cơ sở lý thuyết cần thiết, các cách tiếp cận của những nghiên cứu tương tự đã thực hiện trước đó. Tiếp theo, em tiến hành phân tích, chọn lựa hướng tiếp cận phù hợp nhất với trình độ và với yêu cầu đề tài.

Cuối cùng, em tiến hành lập trình các chức năng của đề tài, kết hợp kiểm thử. Hai quá trình này được thực hiện xoay vòng cho đến khi có được sản phẩm phù hợp, tối ưu nhất.

6. Kết quả đạt được

Kết quả đạt được của đề tài là lập trình thiết kế thành công hệ thống thử quần áo ảo (Virtual Fitting Room) dưới dạng thực tế tăng cường (augmented reality). Về phần các mẫu quần áo, yêu cầu là các mẫu 3D phải đa dạng, có thể thay đổi màu sắc và chất liệu (materials). Giao diện người dùng (UI) cần được thiết kế đẹp mắt, gọn gàng và dễ hiểu để người sử dụng có thể tiếp cận được sản phẩm một cách dễ dàng nhất.

Về mặt điều khiển, các lệnh điều khiển bằng giọng nói được lập trình bằng tiếng anh, nên các lệnh phải đơn giản, dễ nói và dễ hiểu. Tương tự, các cử chỉ cơ thể, tay dùng để điều khiển bằng cử chỉ cũng phải đơn giản, dễ thực hiện.

Hệ thống Phòng thay đồ thực tế tăng cường(Augmented Fitting Room) yêu cầu chạy mượt, các tương tác với người dùng diễn ra tự nhiên, hệ thống tiếp nhận yêu cầu điều khiển và phản hồi chính xác, nhanh (quick and accurate response).

7. Bố cục luận văn

Phần 1: Giới thiệu

- Giới thiệu tổng quan về đề tài.

Phần 2: Nội dung

- Mô tả chi tiết bài toán, các cơ sở lý thuyết đã áp dụng.
- Thiết kế và xây dựng phần mềm.
- Kiểm thử và đánh giá.

Phần 3: Kết luận

- Kết quả đạt được và hướng phát triển phần mềm.

PHẦN NỘI DUNG

CHƯƠNG 1: MÔ TẢ BÀI TOÁN

1. Mô tả chi tiết bài toán

Phòng thử quần áo tăng cường với camera Kinect v2 và Unity là một phần mềm chạy trên Windows với mục đích giúp người dùng có thể thử những mẫu quần áo khác nhau mà không phải trực tiếp mặc vào.

Người dùng bắt đầu sử dụng phần mềm bằng cách đứng trước camera Kinect và hệ thống sẽ tự nhận diện cơ thể của họ. Để bắt đầu sử dụng ứng dụng thì người dùng cần phải tiến hành chọn lựa giới tính. Việc chọn lựa này có thể được tiến hành bằng cử chỉ hoặc giọng nói.

Sau khi người dùng lựa chọn giới tính xong, hệ thống sẽ căn cứ vào giới tính mà hiển thị ra những mẫu quần áo thích hợp trên một menu chọn lựa hình chữ nhật nằm ở phía tay trái người dùng. Người dùng sẽ tiếp tục chọn mẫu áo, lúc này, hệ thống sẽ tính toán độ dài từ cổ đến hông người dùng, sau đó căn chỉnh mẫu áo và hiển thị lên màn hình tại vị trí người dùng đang đứng. Người dùng có thể trượt thanh hiển thị quần áo bằng cách nắm tay lại ở nửa trên màn hình để trượt lên, và ở nửa dưới màn hình để trượt xuống để chọn lựa các mẫu quần áo khác.

Những mẫu áo sẽ có số lượng chất liệu nhất định phù hợp với chúng, người dùng cũng tương tác với hệ thống như lúc chọn mẫu áo. Các chất liệu được hiển thị về phía bên tay phải người dùng. Bên cạnh việc chọn chất liệu qua con trỏ, người dùng còn có thể chọn chúng thông qua giọng nói. (*Xem hình 1).*

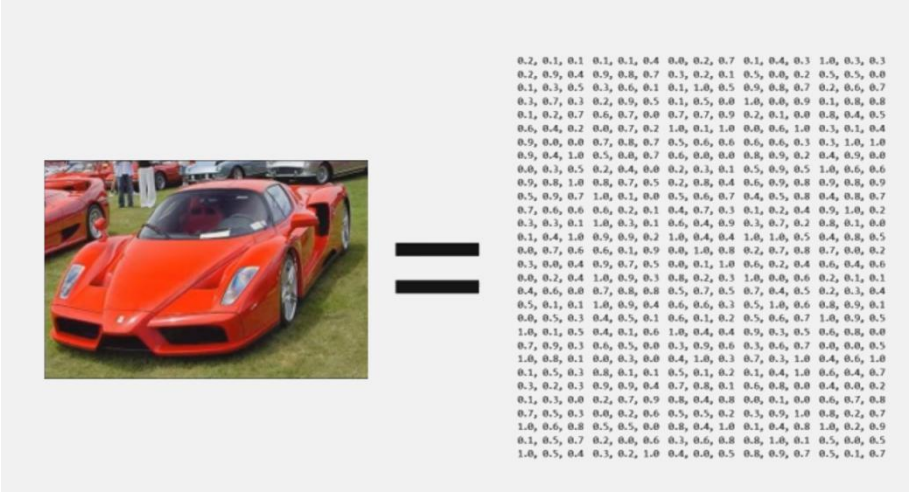
Hình 1. Hệ thống thử quần áo ở cửa hàng Topshop - Mỹ

2. Vấn đề và giải pháp liên quan đến bài toán

2.1. Bài toán xác định khung xương của người sử dụng (Skeleton Detection)

Xác định khung xương nói riêng, hay còn gọi là nhận diện vật thể nói chung, là một vấn đề mà thị giác máy tính đã tìm kiếm hướng giải quyết trong một thời

gian dài. Khởi đầu của việc nhận diện vật thể khá khó khăn, vì hình ảnh đối với máy tính chỉ là một ma trận điểm ảnh chứa giá trị của các kênh màu RGB. Như vậy, vấn đề của việc nhận dạng vật thể là chúng ta cần tìm cách để biến những giá trị RGB đó thành một thứ thực sự có ý nghĩa.



Hình 2. Biến hình ảnh vật thể thành thông tin

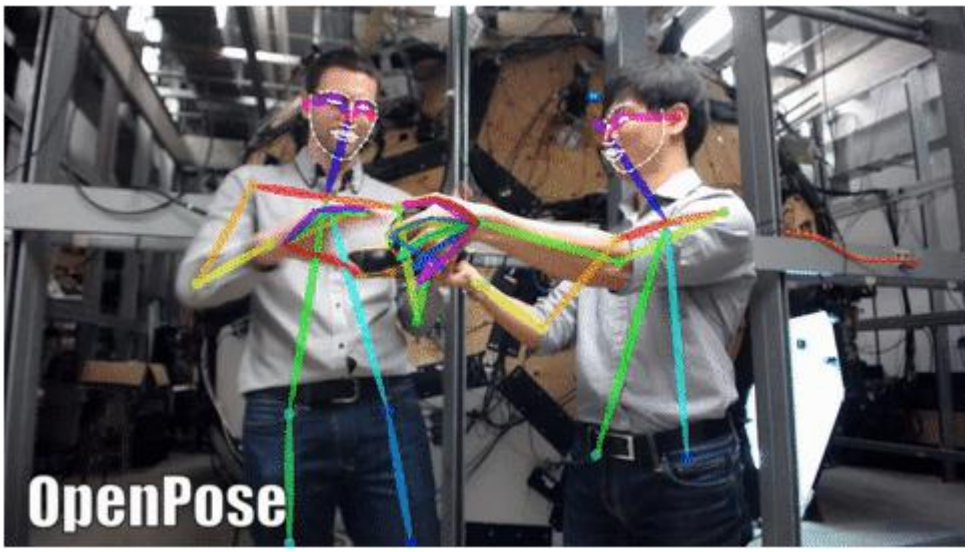
Bên cạnh đó, việc thay đổi góc nhìn, hoặc sự trùng lặp màu sắc so với cảnh nền cũng có thể đưa ra những kết quả khác nhau. Như vậy, việc lập trình để giải quyết toàn bộ các vấn đề khách quan nói trên gần như bất khả thi.

Hình 3. Khó khăn trong việc nhận dạng vật thể

Do đó, những nhà nghiên cứu đã dần dần chuyển sang xu hướng sử dụng máy học để nhận diện cơ thể người. Và từ đây cũng hình thành nên hai hướng đi khác nhau cho việc xác định khung xương, đó là: xác định khung xương với ảnh màu, và xác định khung xương với ảnh độ sâu.

A) *Xác định khung xương với ảnh màu - hướng tiếp cận phần mềm*

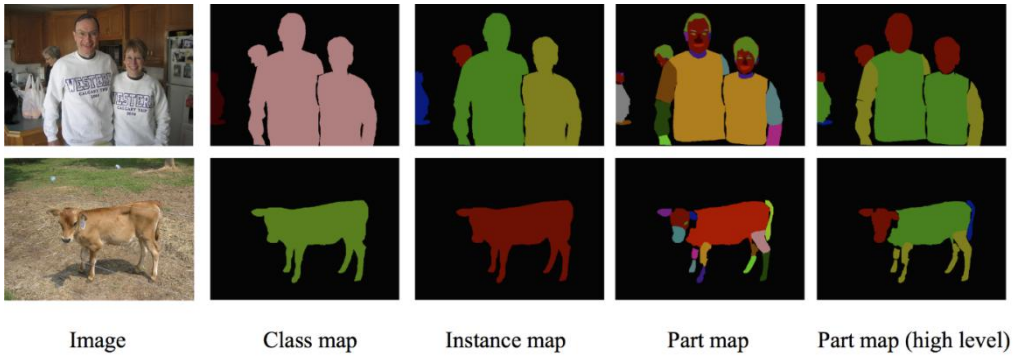
Xác định khung xương với ảnh màu phương pháp xác định khung xương dựa trên luồng video từ camera bình thường. Một trong những dự án nổi bật nhất của hướng đi này chính là dự án OpenPose[15]. OpenPose là một thư viện về thị giác máy tính được được viết trên ngôn ngữ C++, sử dụng OpenCV và Caffe. OpenPose có thể nhận diện khung xương con người theo thời gian thực. Với dữ liệu đầu vào là hình ảnh, video, luồng video từ webcam, ... OpenPose sau đó sẽ xử lý và xuất dữ liệu đầu ra tùy theo nhu cầu của người dùng như: hình ảnh/video thể hiện các khớp xương, hoặc tập dữ liệu dưới dạng văn bản như JSON, XML.



Hình 4. Thư viện OpenPose

Về cơ bản, chức năng của OpenPose là giống với camera Kinect. Tuy nhiên, OpenPose yêu cầu người dùng có hiểu biết nhất định về máy tính để có thể cài đặt và sử dụng nó theo nhu cầu. Bên cạnh đó, OpenPose cũng sẽ tiêu tốn không ít tài nguyên máy tính do số tác vụ cần làm trong một khung hình lớn hơn so với Kinect, điều này cũng dẫn đến tốc độ xử lý cũng sẽ chậm hơn.

Do sử dụng ảnh màu thông thường (không có thông tin về độ sâu), nên đầu tiên OpenPose sẽ phân ngưỡng ảnh để tìm ra cơ thể con người ở đâu trong bức ảnh, sau đó mới tiến hành nhận diện khung xương.

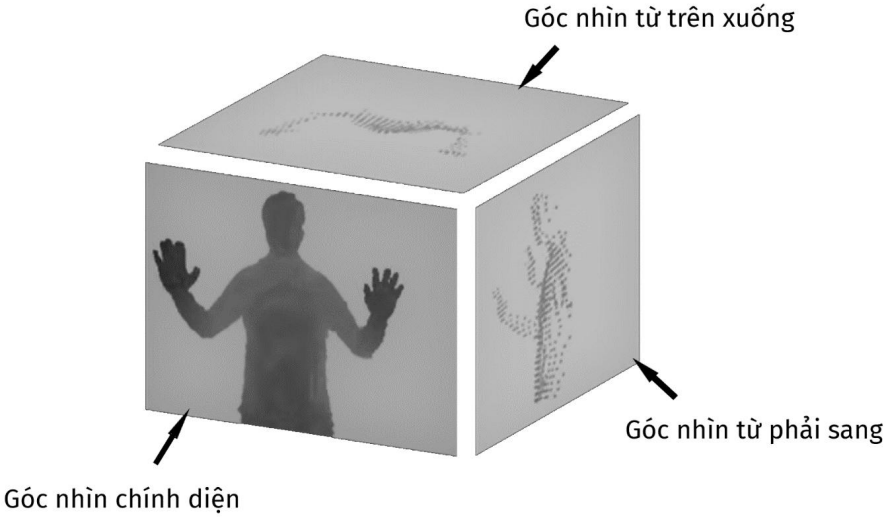


Hình 5. Quá trình phân ngưỡng tách chủ thể ra khỏi nền OpenPose

B) Xác định khung xương với ảnh độ sâu - hướng tiếp cận phần cứng

Ở đề tài này, chúng em sử dụng thiết bị phần cứng: Camera Kinect của Microsoft để nhận dạng khung xương người. Xác định khung xương với camera Kinect là quá trình nhận diện cơ thể con người thông qua các ảnh độ sâu. Kinect sẽ không phải thực hiện tách chủ thể bằng ảnh phẳng và nhờ vào một đám mây các điểm trong không gian ba chiều thu được từ cụm cảm biến độ sâu. Mô hình này giúp Kinect tiết kiệm một khoảng thời gian khá lớn ở việc phân ngưỡng vì chỉ cần loại bỏ các điểm có khoảng cách nhất định so với camera. Sau đó, Kinect sẽ đưa đám mây các điểm ảnh đã phân ngưỡng đó qua một mô hình đã được Microsoft huấn luyện sẵn được tích hợp bên trong, kết quả thu được chính là luồng dữ liệu BodyStream. Vì vậy, thời gian thực hiện cũng nhanh hơn.

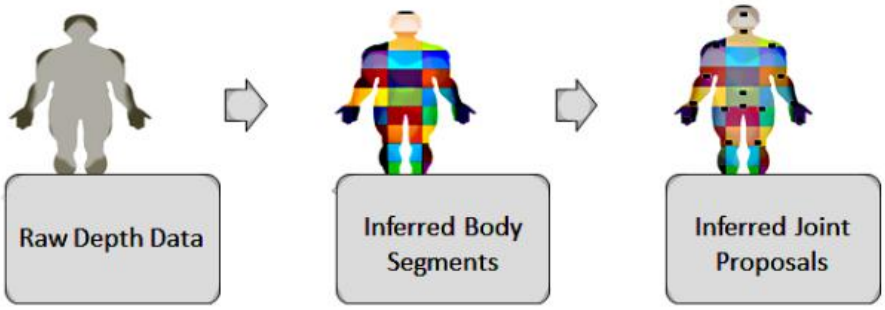
Hình 6. Hình ảnh thu được từ bộ cảm biến độ sâu camera Kinect



Hình 7. Không gian ba chiều tạo ra từ bộ cảm biến độ sâu của camera Kinect

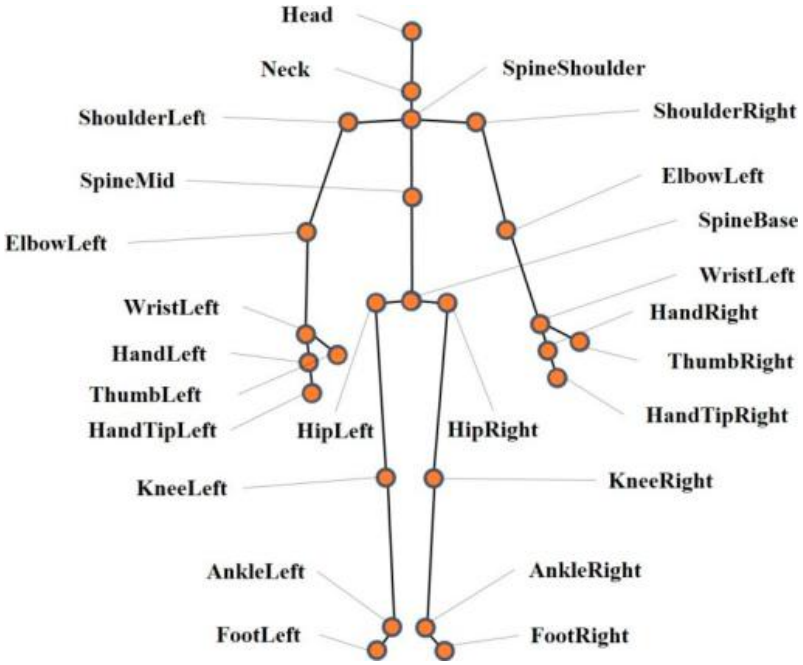
Bước đầu tiên trong quá trình Nhận dạng khung xương của Kinect Sensor là chỉ ra (identify) một vật thể là cơ thể người. Thông tin của quá trình này đơn giản chỉ là hình ảnh mức xám thô của DepthStream. Tiếp theo là quá trình gán nhãn

cho các bộ phận khác nhau của cơ thể người - quá trình này chia cơ thể người ra thành nhiều phần nhỏ khác nhau. Kinect sử dụng thuật toán Cây Quyết Định (Decision Tree) để thực hiện công việc gán nhãn này. Sau khi các bộ phận của cơ thể đã được xác định, Kinect định vị các khớp xương vào vị trí với độ chính xác cao nhất có thể. Hình dưới đây mô tả quá trình nhận dạng khung xương nói trên.



Hình 8. Quá trình nhận dạng khung xương của Kinect 2

Với camera Kinect V2, khung xương bao gồm 25 khớp, mỗi khớp xương được phân biệt và gọi để sử dụng bằng tên (head, shoulders, elbows, wrists, arms, spine, hips, knees, ankles, etc). Hình dưới đây là 25 khớp xương này cùng với tên gọi của chúng.



Hình 9. Các khớp xương nhận diện được bằng camera Kinect

2.2. Giới thiệu Camera Kinect

Camera Kinect là một thiết bị đầu vào ở dạng cảm biến chuyển động (movement sensor) do hãng Microsoft sản xuất dành cho Xbox 360 và máy tính Windows. Khởi nguồn từ dự án "**Project Natal**" được bắt đầu vào năm 2008, đến cuối năm 2010, Microsoft đã cho ra mắt chiếc camera Kinect v1.

Kinect hiện nay có hai phiên bản: Xbox 360 và V2. Trong đề tài luận văn này, em sẽ sử dụng Kinect Camera V2 .Vì vậy, em sẽ chỉ tập trung trình bày cấu tạo và thông tin cần thiết về Camera Kinect V2.

Cấu tạo của Camera Kinect:

Hình 10. Camera Kinect V2

Camera Kinect V2 gồm các thành phần như sau:

- **RGB Camera (Camera màu RGB):** chức năng của camera này là nhận biết ba màu cơ bản đỏ (red - R), xanh lá cây (green - G) và xanh dương (blue - B). Khi thu hình ảnh bằng camera này, ta sẽ có ảnh màu - là sự kết hợp của ba ảnh Đỏ, Xanh lá và Xanh dương. Độ phân giải của camera là 1920 x 1080, tốc độ frame là 30fps.

- **3D Depth Sensor (Cảm biến chiều sâu 3D):** cảm biến chiều sâu gồm có hai bộ phận là Bộ phát hồng ngoại (IR Emitter) và cảm biến chiều sâu. Độ phân giải của Cảm biến chiều sâu 3D là 512 x 424, với tốc độ 30fps, đây cũng là độ phân giải của DepthStream - luồng dữ liệu nhận được từ cảm biến chiều sâu.

- **Microphone Array (Mảng các microphone):** Một Microsoft Array là một microphone được cấu tạo từ một mảng nhiều microphone, đặt ở nhiều vị trí khác nhau. Microphone Array của Kinect v2 gồm 4 microphone quản lý 4 kênh âm thanh.

Bảng dưới đây thể hiện rõ các thông số kỹ thuật của Kinect v2:

Chức năng	Kinect V2
Độ phân giải ảnh màu	1920 x 1080 30fps
Độ phân giải ảnh hồng ngoại (IR Image Resolution)	512 x 424 30fps
Độ phân giải ảnh chiều sâu	512 x 424 30fps
Vùng nhìn thấy (Field of View)	84° ngang x 54° dọc

camera RGB	
Vùng nhìn thấy (Field of View) cảm biến chiều sâu	70° ngang x 60° dọc
Phạm vi độ sâu (Depth Sensing Range)	0.5m - 4.5m
Nhận và theo dõi khung xương (Skeleton Tracking)	Tối đa 6 vật thể, 25 khớp điểm (joint) /khung xương.
Cử chỉ có sẵn (Built-in Gestures)	Trạng thái bàn tay (đóng, mở) và con trỏ bàn tay.
Chuẩn USB	3.0

Bảng 1. Thông số kỹ thuật của Kinect v2

Các luồng dữ liệu của camera Kinect:

a) **Dữ liệu màu (ColorStream):** Ảnh màu của camera Kinect V2 được tạo ra từ camera màu RGB. Một ảnh màu được tạo ra bằng cách kết hợp hình ảnh từ ba kênh màu Đỏ-Xanh lá-Xa dương từ camera này. Ảnh kết quả là một ảnh màu bình thường có độ phân giải 1920 x 1080.

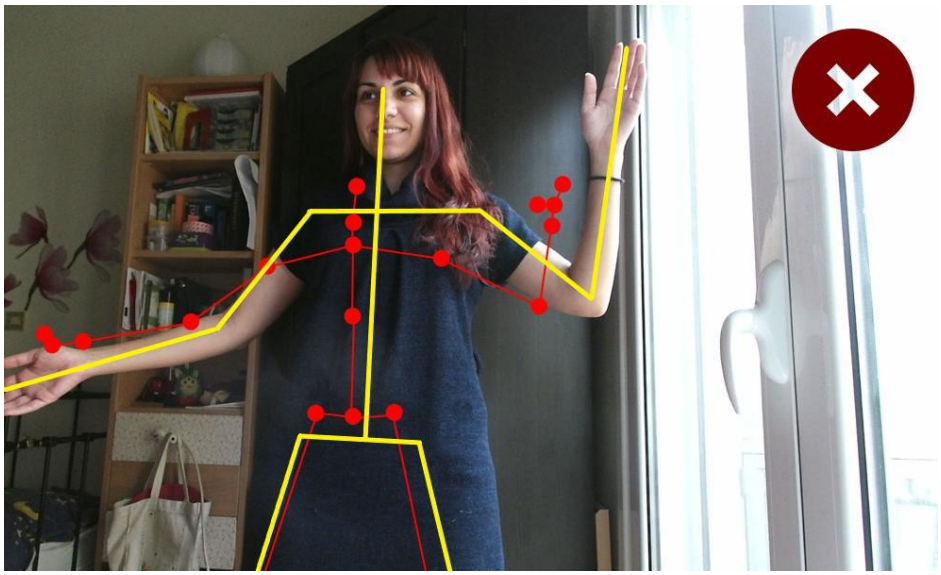
b) **Dữ liệu chiều sâu (DepthStream):** Hình ảnh chiều sâu có độ phân giải 512 x 424 và là ảnh mức xám 16 bit. Dựa vào DepthStream, Kinect có thể tính toán và cho ra BodyStream và thực hiện nhận dạng khung xương. Hình ảnh thu được là một ảnh trắng đen, vật thể càng gần thì điểm ảnh càng sáng, càng xa thì điểm ảnh càng tối.

c) **Âm thanh (AudioStream):** AudioStream là luồng dữ liệu âm thanh thu được từ microphone array của camera Kinect, có kiểu data 16-bit và tần số là 16 kHz. Dữ liệu này được thu vào dưới dạng sóng analog, và được microphone của Kinect chuyển đổi thành dạng tín hiệu điện tử để xử lý bằng máy vi tính.

d) **Dữ liệu cơ thể người (BodyStream):** BodyStream là luồng dữ liệu bao gồm thông tin tracking real-time của tất cả người đang trong tầm nhìn của cảm biến Kinect. Trong BodyStream còn bao gồm cả dữ liệu về khớp điểm và khung xương (Skeleton Joints), hướng (orientation), trạng thái bàn tay (hand states) cho tối đa 6 người. Dữ liệu ở dạng không gian 3 chiều (X,Y,Z) với X,Y là giá trị X,Y của Hình ảnh chiều sâu (Depth Image) và Z là khoảng cách từ từng khớp xương (joint) đến camera.

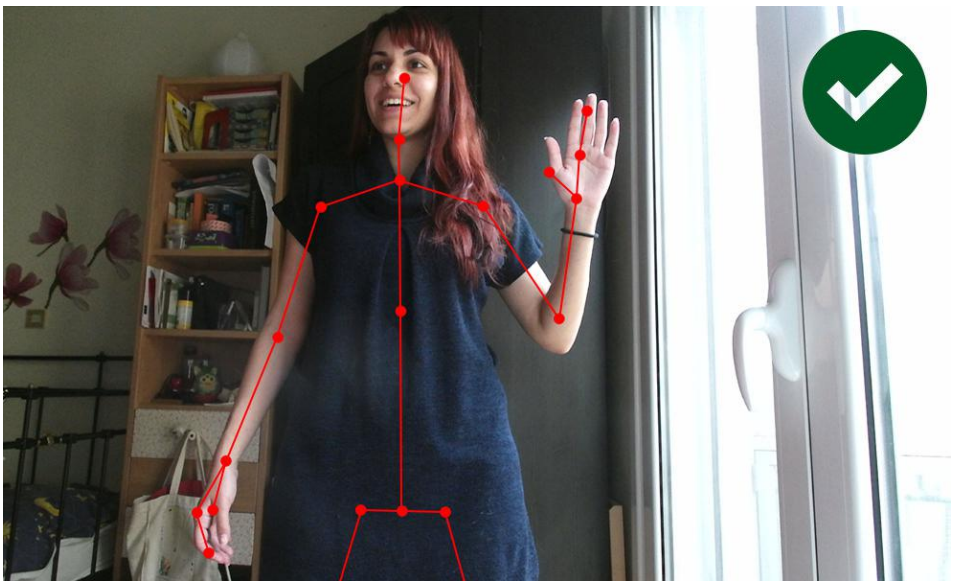
2.3. Chuyển đổi vị trí khung xương từ BodyStream sang ColorStream

Do sự khác biệt về độ phân giải giữa ColorStream và BodyStream, các khung xương nếu được dựng cùng độ phân giải với luồng ảnh màu sẽ cho ra kết quả không được chính xác. Vì vậy, chúng ta phải thực hiện một vài phép toán để có thể hiển thị các khớp xương đúng vị trí.



Hình 11. Khung xương nhận diện được nhưng chưa đúng vị trí

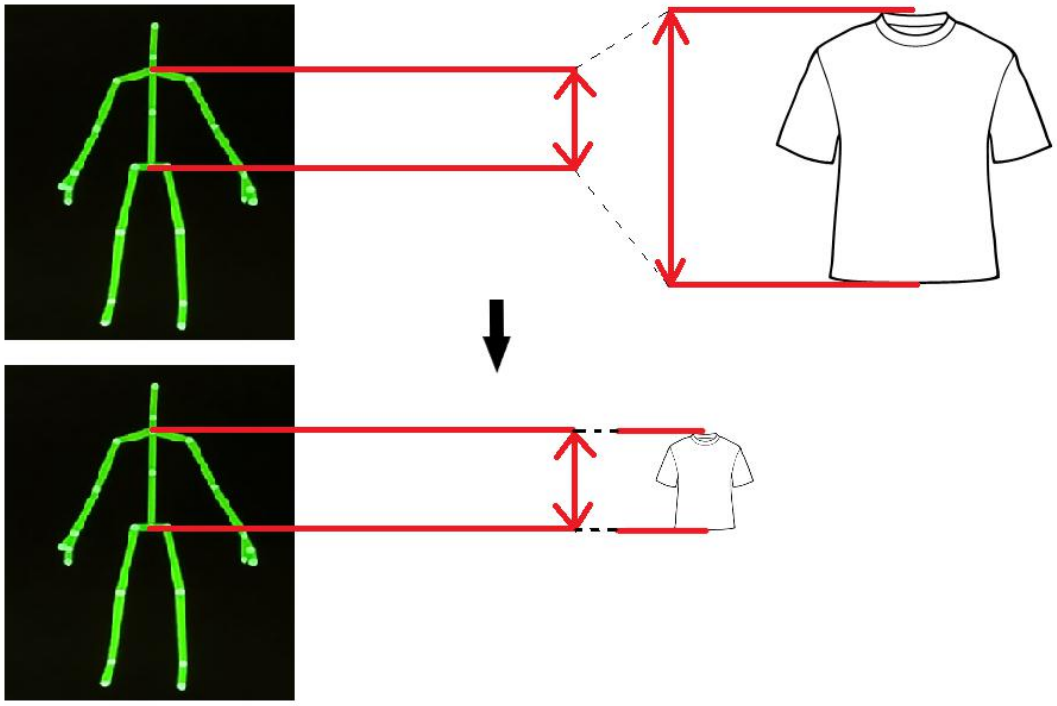
Tuy nhiên, việc chuyển đổi vị trí này có thể được thực hiện bằng một hàm có sẵn trong gói Kinect SDK gọi là CoordinateMapper. CoordinateMapper sẽ giúp người dùng chuyển đổi qua lại giữa các hệ tọa độ của ColorStream và BodyStream. Kết quả sau khi thực hiện chuyển đổi sẽ như hình sau:



Hình 12. Khung xương nhận diện được sau khi thực hiện Coordinate Mapping

2.4. Bài toán căng chỉnh kích cỡ mẫu áo

Với các mẫu áo khác nhau thì kèm theo đó, kích thước của chúng cũng khác nhau. Điều này cũng tương tự đối với mô hình 3D của nó. Cho nên, chúng ta cần phải điều chỉnh kích cỡ của chúng sao cho phù hợp với từng khách thử đồ một cách tự động. Hướng tiếp cận của em chính là sử dụng tỉ lệ chiều dài cơ thể chia cho chiều dài mô hình áo, và tiếng hành thu phóng trên cả ba chiều X, Y, Z bằng tỉ lệ đó. Điều này sẽ hạn chế việc mô hình áo bị vỡ, bị biến dạng do việc thu phóng không đều.



Hình 13. Căng chỉnh chiều dài áo theo người dùng

2.5. Bài toán áp các mô hình quần áo 3D

Hệ thống cần phải hiển thị đúng các mẫu quần áo 3D lên đúng vị trí trên cơ thể người và các mẫu quần áo này phải có thể di chuyển, xoay được theo chuyển động của người dùng. Nếu người dùng lùi ra xa/ tiến lại gần cảm biến Kinect, mẫu quần áo cũng phải trở lên nhỏ đi/ to xa cho phù hợp.

Giải pháp để giải quyết vấn đề này chính là việc nhận dạng được khung xương. Với việc truy xuất được các khớp khác nhau của khung xương người, kết hợp với các điểm tựa trên mẫu quần áo 3D, hệ thống có thể hiển thị chính xác vị trí áo/ quần lên cơ thể người. Ngoài ra, vì chức năng nhận dạng khung xương của Kinect còn có thể lưu trữ được dữ liệu về chuyển động và chiều sâu của hình ảnh, nên bằng cách truy xuất các thông số này, ta có thể làm cho các mẫu quần áo đi theo và thay đổi kích thước theo vị trí của người sử dụng.

Một vấn đề khác của bài toán này là đối với các mẫu quần áo tải trên mạng có thể có kích thước quá lớn so với khung xương được hiển thị. Bằng cách lập tỉ lệ

giữa khoảng cách các khớp và khoảng cách các điểm tựa của mẫu quần áo, hệ thống có thể tính toán để scale - thu nhỏ/ phóng to mẫu cho vừa với cơ thể người dùng.

2.6. Các vấn đề về điều khiển

Để người dùng có thể thực hiện được các thao tác điều khiển như chọn mẫu quần áo, chọn các mẫu cho phù hợp với giới tính, tạm dừng hệ thống,… ta có hai hướng giải quyết:

2.6.1. Sử dụng phân hệ nhận dạng cử chỉ:

Nhận dạng cử chỉ là giúp máy tính hiểu được ý nghĩa những cử động của con người, từ đó mô phỏng lại những cử động đó ở các mô hình ba chiều, hoặc xây dựng nên những ứng dụng tương tác với người dùng mà không cần phải chạm vào các thiết bị ngoại vi. Nhận dạng cử chỉ thường được áp dụng đối với khuôn mặt và bàn tay vì hai vị trí đó có nhiều hành động, biểu cảm khác nhau. Nhận dạng cử chỉ cũng là nền móng để chúng ta có thể phát triển công nghệ thực tế ảo, thực tế tăng cường, theo dõi chuyển động để dựng phim, phát hiện nụ cười để chụp ảnh ở các máy ảnh...

Có hai cách tiếp cận đối với vấn đề này chính là sử dụng phần cứng và sử dụng phần mềm.

Đối với hướng tiếp cận sử dụng phần cứng, chúng ta sẽ thực hiện dán các cảm biến trực tiếp lên cơ thể con người và từ đó dựng nên các mô hình ba chiều trực tiếp từ dữ liệu của các cảm biến đó.. Điển hình chính là các trường quay/dựng phim của hãng phim Hollywood, hoặc các hãng làm trò chơi thể thao như Electronic Art, 2K. Các diễn viên sẽ được trang bị một bộ cảm biến giúp tái tạo lại những hành động của họ trên máy tính, và các hành động đó sẽ được ghép vào các nhân vật trong phim, trò chơi của hãng. Hướng tiếp cận này thường được biết đến với tên **Motion Capture**.

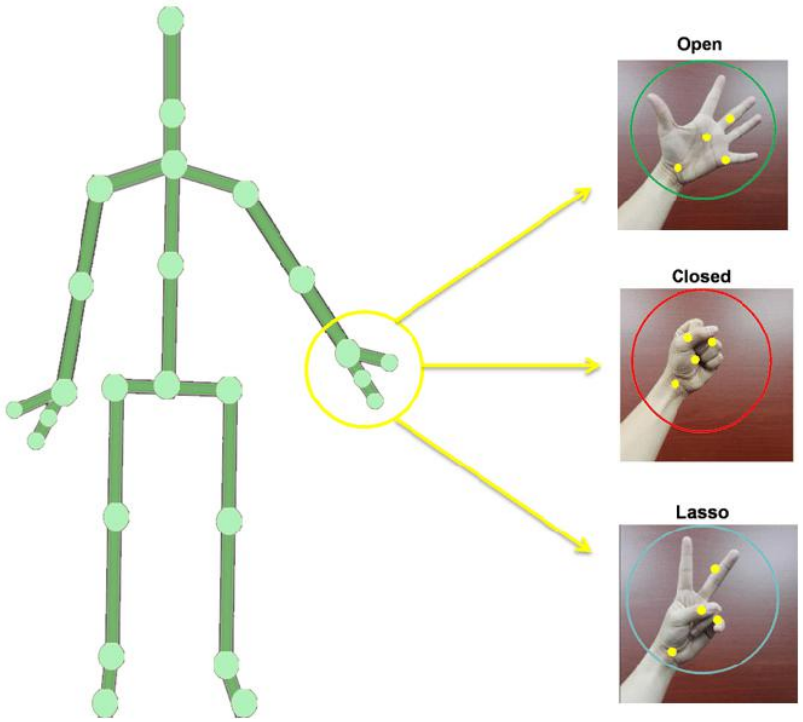
Hình 14. Bộ đồ gắn các cảm biến trong các studio game, studio làm phim

Hình 15. Hình ảnh thu được từ bộ đồ cảm biến

Mặc dù chất lượng đầu ra của hướng tiếp cận trên là cực kì tốt, từ hình ảnh đến số khung hình trên giây. Tuy nhiên, nó chỉ thực sự phù hợp với các hãng làm phim, trò chơi vì việc ghi nhận chuyển động trực tiếp đó tốn khá nhiều công sức cũng như diện tích. Sẽ rất khó để có thể bày biện mọi thứ như thế trong một không gian nhỏ như phòng thử đồ. Do vậy, em sẽ sử dụng hướng tiếp cận phần mềm để nhận dạng cử chỉ con người.

Đối với hướng tiếp cận phần mềm, chúng ta sẽ thực hiện nhận dạng cử chỉ thông qua các phép toán từ các dữ liệu thu được của các cảm biến hoặc máy ảnh. Camera Kinect giúp chúng ta dễ dàng đạt được điều này với mô hình đã được Microsoft huấn luyện sẵn. Ngoài việc xác định được khung xương con người, camera Kinect còn hỗ trợ theo dõi các cử chỉ của bàn tay. Theo như cấu trúc dữ liệu **HandState** được định dạng sẵn trong Kinect, dùng để diễn tả trạng thái của bàn tay, thì có tất cả 5 trạng thái:

- **Closed:** Bàn tay đang nắm thành nắm đấm.
- **Lasso:** Bàn tay đang ở trạng thái lasso.
- **NotTracked:** Bàn tay chưa được theo dõi.
- **Open:** Bàn tay đang mở tự nhiên.
- **Unknown:** Không nhận diện được trạng thái bàn tay.



Hình 16. Các trạng thái của bàn tay do camera Kinect ghi nhận

Ở đề tài này, em sẽ sử dụng trạng thái **Closed** của bàn tay để thực hiện thao tác cuộn thanh hiển thị áo.

2.6.2. Sử dụng phân hệ nhận dạng giọng nói:

Người dùng nói ra/phát âm các lệnh điều khiển tương ứng với các thao tác. Các lệnh âm thanh này được thu vào bằng hệ thống microphone của camera Kinect. Sau đó, hệ thống phải nhận dạng và hiểu được lệnh này yêu cầu thực hiện chức năng điều khiển nào. Đây là chức năng do em phụ trách.

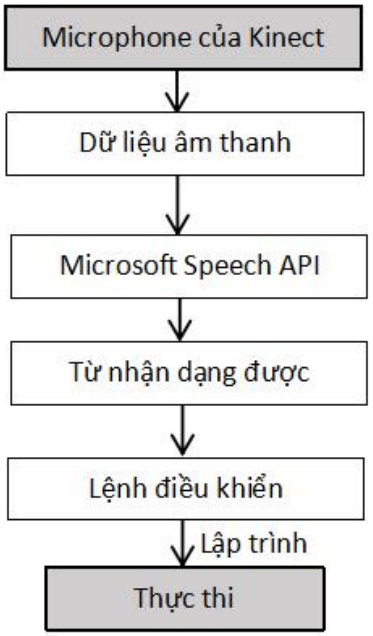
Để làm được điều này, em sử dụng bộ nhận dạng cụm từ của Unity: Unity Phrase Recognition. Đây là hệ thống được tích hợp trong thư viện của Unity, có

chức năng quản lý các bộ nhận dạng cụm từ và các sự kiện được gọi kèm theo khi các bộ nhận dạng này được gọi.

Hệ thống sẽ thu âm thanh từ audio stream của Kinect, thông qua lập trình để chuyển đổi kiểu dữ liệu cho phù hợp với dữ liệu trong Unity. Sau đó, dữ liệu âm thanh sẽ được truyền qua một hệ thống nhận diện giọng nói. Giải pháp của vấn đề này là API của Microsoft, thư viện Windows Speech. Các lớp (class) âm thanh của Unity và Kinect đều kế thừa từ Windows.Speech. Các hệ thống nhận dạng giọng nói thường được chia là hai loại: Command Recognition (Nhận dạng dòng lệnh) và Free-form Dictation (Nhận dạng chính tả).

- Free-form dictation: người sử dụng cần huấn luyện cho hệ thống bằng cách phát âm/ nói đi nói lại một cụm các câu/ từ nhiều lần để hệ thống "quen" được với giọng nói của người sử dụng.
- Command Recognition: người dùng không cần huấn luyện. Lập trình viên định nghĩa sẵn một bộ các từ vựng mà người dùng có thể sử dụng để tương tác với một hệ thống. API nhận dạng sẽ chia nhỏ tín hiệu âm thanh thu được ra thành các "từ" và tìm kiếm các từ này trong bột từ vựng có sẵn. Nếu tìm thấy, API sẽ xem như quá trình nhận dạng thành công.

Thư viện Unity cung cấp lớp PhraseRecognition và KeywordRecognizer để "lắng nghe" các từ khóa từ tín hiệu âm thanh và tiến hành so sánh chúng với bộ từ vựng. Kinect chỉ đóng vai trò là một thiết bị thu thanh đầu vào, quá trình nhận dạng trong đề tài được thực hiện bởi hai lớp này của Unity và thư viện có sẵn của hệ thống : Windows.Speech.



Hình 17. Quá trình điều khiển bằng giọng nói

2.7. Các công cụ lập trình và thư viện hỗ trợ

2.7.1 Ngôn ngữ lập trình C#

C# hay C-sharp là ngôn ngữ lập trình hướng đối tượng (Object Oriented Programming Language) được phát triển bởi Microsoft. C# được phát triển dựa trên hai ngôn ngữ lập trình khác là Java và C++ và được phát hành lần đầu trên vào năm 2000. Phiên bản C# gần đây nhất là C# 8.0, được phát hành năm 2019 cùng với Visual Studio 2019 phiên bản 16.3.

Đặc điểm của ngôn ngữ lập trình C#:

- C# là ngôn ngữ đơn giản: C# có cú pháp đơn giản như Java và C++, nhưng đã được cải tiến để trở nên thân thiện hơn, có ít từ khóa hơn.
- C# là ngôn ngữ lập trình hướng đối tượng: chính vì vậy nó có đầy đủ các tính chất của loại ngôn ngữ lập trình này. Đó là tính trừu tượng (Abstraction), tính đóng gói (Encapsulation), tính đa hình (Polymorphism) và tính kế thừa (Inheritance).

Trong đề tài luận văn này, em lựa chọn ngôn ngữ lập trình C# để phát triển ứng dụng.

2.7.2 Visual Studio Code

Visual Studio Code (VS Code) là một phần mềm chỉnh sửa văn bản do Microsoft phát hành cho cả ba nền tảng: Windows, Linux và MacOS. Sở hữu khả năng tùy chỉnh mạnh mẽ bằng việc cài đặc các tiện ích mở rộng, VS Code gần như hỗ trợ toàn bộ ngôn ngữ lập trình đang có hiện nay. Các tiện ích mở rộng về ngôn ngữ lập trình sẽ giúp VS Code gợi ý các lệnh/hàm cho người dùng, cũng như tự động hoàn thành câu lệnh. Ngoài ra, VS Code còn cho phép người dùng tùy chỉnh font, kích cỡ chữ, và chủ đề của ứng dụng.

2.7.3 Bộ phát triển ứng dụng Microsoft Kinect SDK

Cuối năm 2011, Microsoft cho ra mắt bộ phát triển ứng dụng dành cho cảm biến Kinect (Microsoft Kinect Software Development Kit - gọi tắt là Kinect SDK). Kinect SDK cho phép nhà phát triển tạo ra các ứng dụng máy tính tương tác với chuyển động cơ thể người, sử dụng 2 ngôn ngữ lập trình là C++ và C#. Hiện nay Kinect SDK có thể được tải miễn phí từ trang chủ của Microsoft và có nhiều phiên bản khác nhau.

Đối với đề tài này, để phù hợp với camera Kinect v2, chúng em sử dụng Kinect SDK 2.0. Cụ thể, bộ phát triển này bao gồm:

- Các drivers cần thiết để sử dụng cảm biến Kinect dành cho máy tính chạy hệ điều này Windows 8 (x64), Windows 8.1 (x64) và Windows Embedded Standard 8 (x64).

- Giao diện lập trình ứng dụng (Application Programming Interface - API) và các giao thức của thiết bị.
- Các ví dụ về các làm việc, khai khác camera Kinect, cách lập trình với Kinect SDK (sample codes).

2.7.4 Unity

Unity là một môi trường lập trình/thiết kế trò chơi được phát triển bởi công ty Unity Technologies có trụ sở tại San Francisco, California.

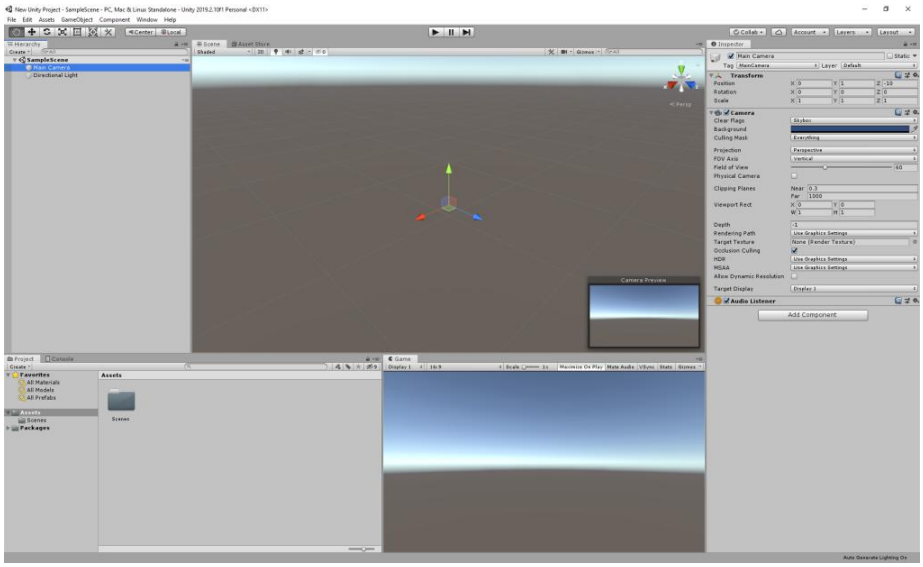
Unity cho phép người dùng tạo ra những trò chơi trong không gian hai chiều, ba chiều hoặc thậm chí là thực tế ảo, thực tế tăng cường. Ngoài ra, bên cạnh những vật thể, hình ảnh có sẵn, người dùng có thể thêm vào dự án của họ những tập tin mới bất kì lúc nào. Với phần mềm sử dụng không gian ba chiều, Unity hỗ trợ những định dạng như *.fbx, *.obj, ...

Thế mạnh của Unity là nó cho phép người dùng lựa chọn giữa việc kéo thả vật thể và điều chỉnh các thông số của chúng. Cùng với đó là Unity Script sử dụng ngôn ngữ lập trình C# cho phép người dùng tạo những event, thiết lập hành vi cho vật thể. Ngoài ra, Unity còn hỗ trợ đóng gói ứng dụng đa nền tảng, giúp người dùng có thể dễ dàng xuất dự án của họ thành tập tin thực thi để chạy trên các máy tính khác nhau.

Các vật thể chính trong một ứng dụng Unity là: Scene, GameObject, MainCamera, UI, Light, Script.

A) Scene:

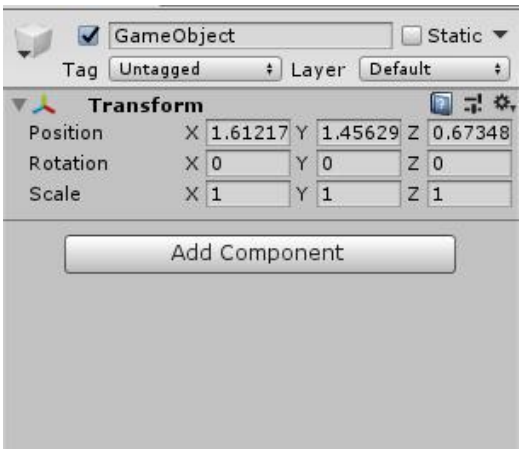
Scene (cảnh/màn chơi) là vật chứa đựng các GameObject, các nút, các bảng tùy chọn. Người dùng có thể thỏa sức chỉnh sửa, thiết kế cảnh bằng các chướng ngại vật, ánh sáng, môi trường. Một ứng dụng có thể có nhiều cảnh, ví dụ như những trò chơi có nhiều cấp độ khác nhau, khi người chơi hoàn thành cảnh x thì sẽ được chuyển sang cảnh y.



Hình 18. Scene trong Unity

B) GameObject:

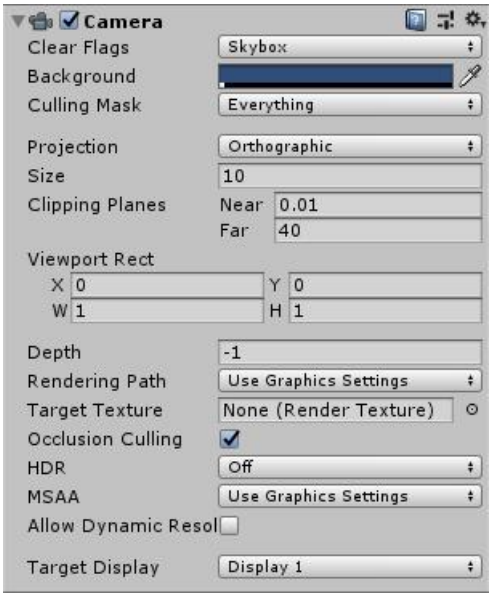
GameObject là lớp vật thể cơ bản nhất trong Unity. Nó không có chức năng nào ngoài việc xác định vị trí của vật thể trong không gian và làm vật chứa đựng những thành phần khác. Người dùng có thể thêm các thành phần (components) khác nhau để tạo ra những GameObject mang chức năng riêng biệt. Có thể kể đến các thành phần như: Light, Skinned Mesh Renderer, C# Script, Video Player, ...



Hình 19. GameObject trong Unity

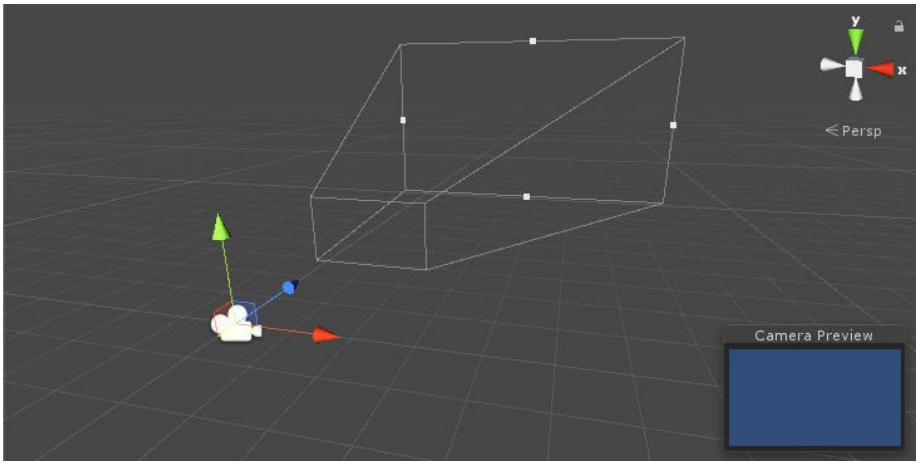
C) Main Camera:

Main Camera là một GameObject có thành phần Camera. Nó được dùng để hiển thị một phần không gian trong cảnh đang chứa nó cho người chơi. Số lượng camera được dùng trong Unity là không giới hạn, tuy nhiên camera thể hiện màn hình chính, hay các màn chơi chính sẽ được gán với tag Main Camera. Camera có hai kiểu góc nhìn khác nhau đó là Perspective và Orthographic.



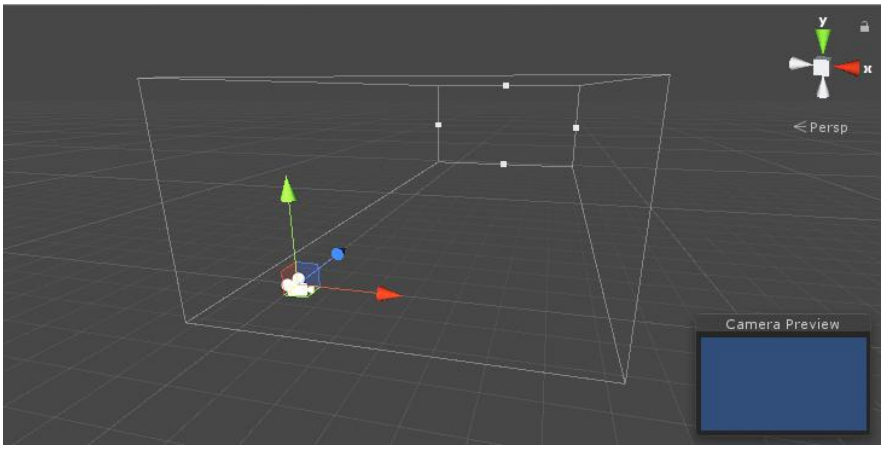
Hình 20. Main Camera trong Unity

Với kiểu Perspective, camera sẽ nhìn không gian giống như mắt của con người. Càng gần camera thì góc nhìn càng hẹp, càng xa camera thì góc nhìn càng rộng. Đối với vật thể, càng gần camera thì càng lớn, càng xa camera thì càng nhỏ.



Hình 21. Perspective Camera

Với kiểu góc nhìn Orthographic, camera sẽ nhìn không gian với góc nhìn đồng bộ. Tức là góc nhìn sẽ không thay đổi ở bất kì vị trí nào. Vật thể cũng sẽ giữ nguyên kích thước dù khoảng cách đến camera có thay đổi.



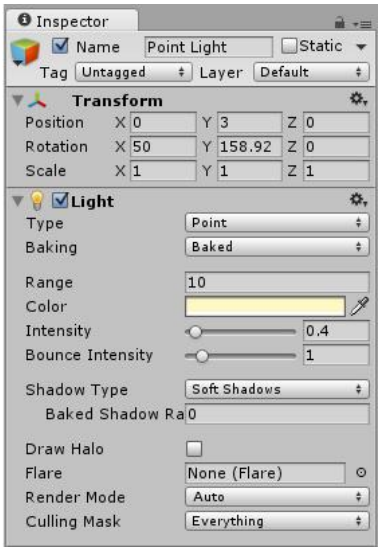
Hình 22. Orthographic Camera

D) UI

Lớp UI bao gồm thành phần dùng để tạo ra giao diện người dùng trong Unity. Có thể kể đến: Button (tạo nút bấm), Canvas (dùng để hiển thị các hình ảnh),...Các thành phần của UI sẽ có thuộc tính RectTransform để xác định vị trí, trọng tâm, cũng như kích thước của chúng thay vì thuộc tính Transform như GameObject.

E) Light

Light (ánh sáng) là một GameObject có thành phần Light được sử dụng để chiếu sáng không gian trong Unity.

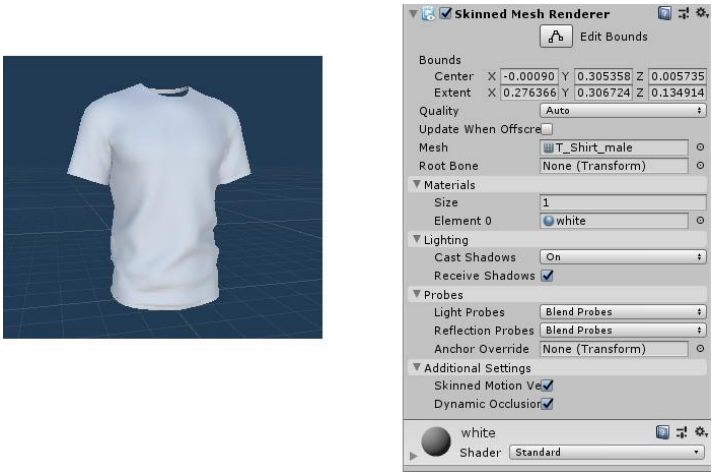


Hình 23. Light (ánh sáng) trong Unity

Chúng ta sử dụng ánh sáng để chiếu sáng cảnh và các vật thể khác, cũng như tạo hiệu ứng đổ bóng, phản chiếu cho vật thể, giúp tạo độ chân thực cho ứng dụng. Thành phần này không có tác dụng đối với các vật thể thuộc lớp UI.

F) Skinned Mesh Renderer

Skinned Mesh Renderer là một thành phần dùng để dựng vật thể ba chiều trong Unity. Nó bao gồm thuộc tính **mesh** và **material**, cho phép người dùng lựa chọn hình dạng vật thể và màu sắc, chất liệu của vật thể đó. Đây là thành phần chính dùng để tạo các mẫu áo được sử dụng trong đề tài này.



Hình 24. Mesh trong Unity

G) Script

Script, hay còn gọi là mã điều khiển hành vi (behavior script), là một phần không thể thiếu trong các ứng dụng của Unity. Hầu hết các ứng dụng đều cần script để có thể phản ứng lại với những dữ liệu đầu vào từ phía người dùng, hoặc để tạo ra các sự kiện và mốc thời gian diễn ra của chúng. Ngoài ra, script còn có thể tạo các hiệu ứng chuyển cảnh, kiểm soát cũng như quản lí các vật thể có trong Unity. Script sử dụng ngôn ngữ C#, và có thể sử dụng bất kì trình soạn thảo văn bản nào để chỉnh sửa.

CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT

1. Phân tích hệ thống

Từ mục tiêu đề ra ở chương 1, hệ thống phải có những chức năng sau đây:

- Hiển thị hình ảnh thu được từ camera Kinect ra màn hình chính
- Hiển thị các mẫu áo, mẫu hoa văn, màu sắc để người dùng lựa chọn.
- Xác định được kích cỡ áo theo chiều dọc và căn chỉnh, sau đó ốp mẫu áo vào người dùng
- Track được vị trí và chuyển động của người dùng để mẫu quần áo di chuyển/ thay đổi kích thước theo.
- Thay đổi được chất liệu mẫu áo.
- Nhận diện được khung xương, cử chỉ người dùng.
- Nhận diện được những từ khóa mà người dùng nói.

Hệ thống sẽ bắt đầu bằng một menu cho phép người dùng lựa chọn giới tính với hình ảnh từ camera RGB của Kinect sẽ được hiển thị phía sau. Các con trỏ sẽ được hiển thị khi camera Kinect đã nhận diện được người dùng. Ngoài ra, họ cũng có thể sử dụng giọng nói của mình để chọn giới tính.

Sau đó, hệ thống sẽ hiển thị các mẫu áo cho người dùng lựa chọn. Khi người dùng chọn mẫu áo, hệ thống sẽ hiển thị thêm các mẫu hoa văn, màu sắc phù hợp với mẫu áo đang hiển thị. Mẫu áo sẽ được căn chỉnh tùy theo chiều cao và khoảng cách đến camera của người dùng. Các mẫu hoa văn, màu sắc cũng sẽ được gắn từ khóa giúp người dùng lựa chọn dễ dàng hơn bằng giọng nói.

2. Thiết kế hệ thống

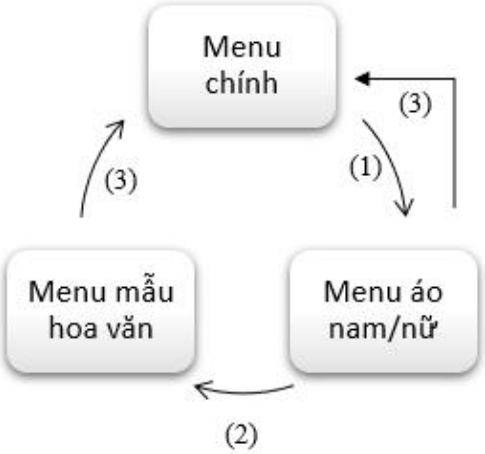
2.1. Giao diện người dùng (User Interface)

Giao diện người dùng sẽ được thiết kế đơn giản nhằm giảm thiểu các hành động mà người dùng phải làm. Màn hình chính sẽ chỉ có 3 nút, bao gồm: “Male” và “Female” để người dùng chọn giới tính và “Exit” để thoát khỏi ứng dụng.

Menu hiển thị các mẫu áo sẽ được đặt bên trái của màn hình, và ở phía đối diện là thanh hiển thị mẫu hoa văn, màu sắc.

Nút “Pause” được đặt cạnh thanh hiển thị bên phải, giúp người dùng có thể chuyển trở về màn hình chính chọn giới tính.

Lưu đồ hiển thị các menu như sau:



Hình 25.Menu hệ thống

Chú thích:

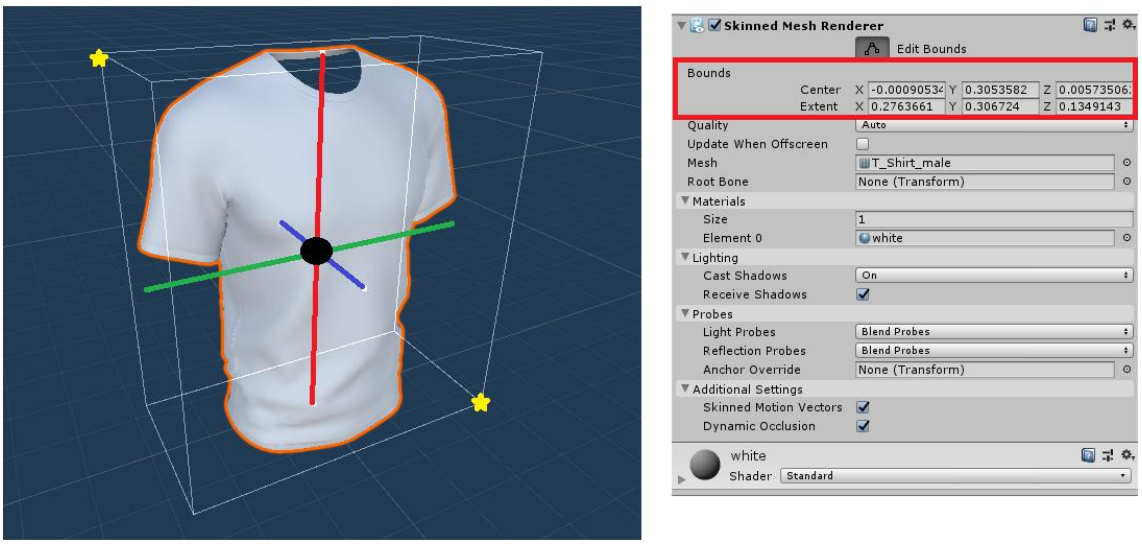
- (1) Người dùng chọn giới tính của bản thân.
- (2) Người dùng chọn/đổi mẫu áo muốn thử.
- (3) Người dùng chọn tạm dừng.

2.2. Mô hình áo/ quần 3D

Các mẫu áo là những mô hình 3D được thiết kế sẵn được lưu dưới định dạng *.fbx. Mỗi mô hình sau khi nhập vào Unity sẽ cung cấp cho ta hai vật thể là: lưới (mesh) và chất liệu tạo thành (material). Ở đề tài này em chỉ sử dụng mesh của mô hình để tạo nên nên các mẫu áo cơ bản với vật liệu là mặc định màu trắng. Những mẫu áo được tạo ra được Unity gọi là các prefab. Các prefab này có thể được dùng trong chương trình thông qua lệnh **Instantiate** mà không phải có mặt trong cảnh từ trước. Instantiate sẽ tạo ra một phiên bản của prefab đó, mỗi khi prefab thay đổi thì những vật thể được tạo thông qua lệnh Instantiate cũng sẽ thay đổi theo.

Những mô hình áo khác nhau sẽ có độ dài khác nhau, vì vậy, việc căng chỉnh mẫu áo cho phù hợp với dáng người dùng là vô cùng cần thiết. Các mẫu áo được tạo ra sẽ có những thông số chính cần được lưu tâm đó là:

- **Scale** (độ thu phóng): mặc định là 1, dùng để thu phóng áo.
- **Center** (trọng tâm): tọa độ trọng tâm của một chiếc hộp bao quanh mẫu áo. Được thể hiện bằng hình tròn màu đen.
- **Extend** (tầm rộng): khoảng cách từ trọng tâm đến 3 mặt phẳng X, Y, Z bao quanh mẫu áo. Thể hiện bằng hình hộp chữ nhật bao quanh áo.
- **Min** (điểm nhỏ nhất): tọa độ điểm này bằng tọa độ **Center** trừ cho tọa độ **Extend**. Được đánh dấu bằng ngôi sao vàng góc dưới bên phải.
- **Max** (điểm lớn nhất): tọa đổ điểm này bằng tọa độ **Center** cộng với tọa độ **Extend**. Được đánh dấu bằng ngôi sao vàng góc trên bên trái.



Hình 26. Mô hình áo 3D

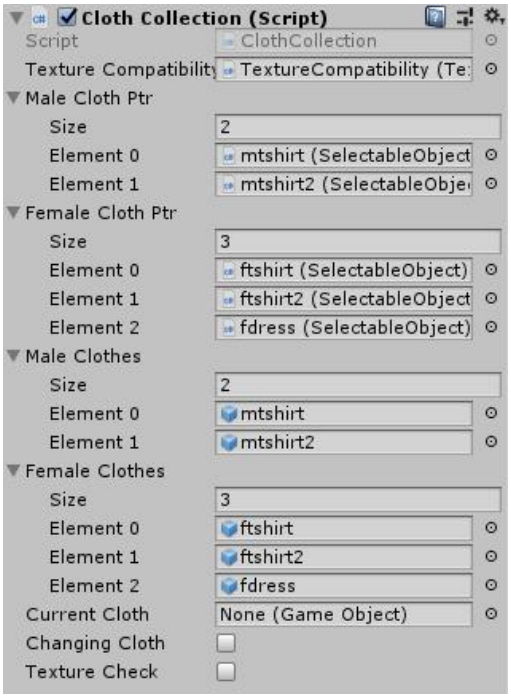
Chiều dài áo được xác định bằng tọa độ y của điểm lớn nhất trừ cho tọa độ y của điểm nhỏ nhất, hoặc bằng 2 lần độ lớn phương y của tầm rộng. Chiều dài áo được thể hiện bằng đoạn thẳng màu đỏ ở hình trên. Sau khi người dùng chọn mẫu áo, hệ thống sẽ tính toán độ dài từ cổ (tương ứng với khớp xương Neck) đến hông (tương ứng với khớp xương SpineBase) của người dùng rồi tiến hành căng chỉnh mẫu áo cho phù hợp. Độ thu phóng sẽ được điều chỉnh theo tỉ lệ chiều dài cơ thể chia cho chiều dài mẫu áo.

2.3 Tạo cấu trúc, hoa văn

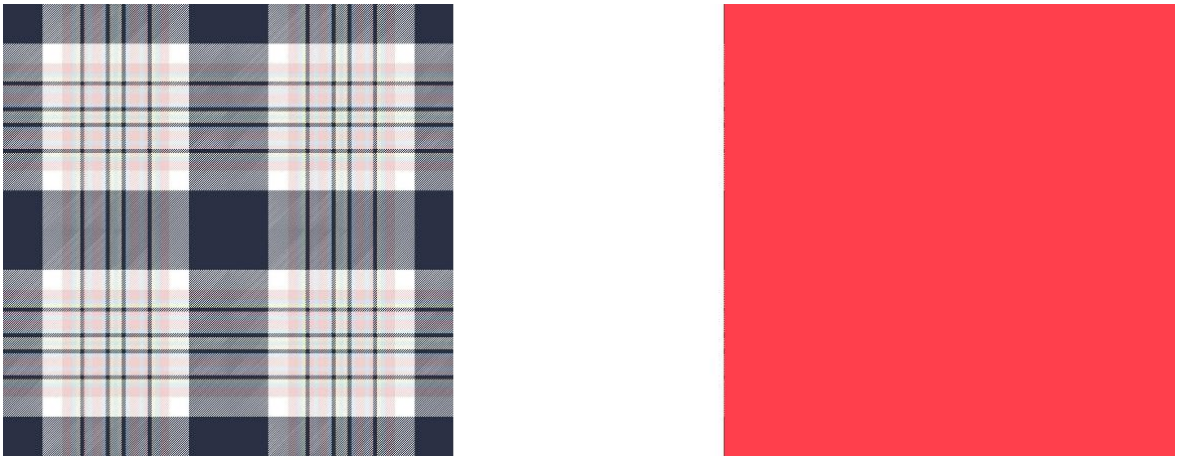
Vấn đề tiếp theo cần xử lý trong đề tài này là việc tạo ra một thư viện mẫu áo, hoa văn để gọi và sử dụng trong lúc lập trình, cũng như để quản lý số lượng. Em sẽ định nghĩa một lớp gọi là **ClothCollection** chứa các mảng một chiều chỉ đến các mẫu áo của nam và nữ, cùng các mảng có kiểu bool tương ứng trong lớp **TextureCompatibility** dùng để xác định loại hoa văn phù hợp.

Trong lớp này sẽ có một hàm gọi là **SetCloth** dùng để thực hiện gán vật thể được chọn cho biến **currentCloth**. Biến bool **changingCloth** được dùng để xác định xem người dùng có đổi áo chưa, và sẽ được đặt là **True** mỗi khi người dùng chọn mẫu áo. Biến changingCloth giúp chương trình biết khi nào người dùng đã chọn mẫu áo mới, thông qua đó hủy đi mẫu áo cũ và tạo mẫu áo mới ốp lên người dùng. Hàm tiếp theo trong lớp ClothCollection là hàm **EnableTexturePanel**. Hàm này sẽ bật, tắt các menu hoa văn/màu sắc tùy thuộc vào mảng biến bool **TextureCheck** được gán giá trị từ lớp TextureCompatibility của mẫu áo tương ứng.

Lớp TextureCompatibility được định nghĩa để tiến hành gán các giá trị True/False cho các mẫu áo nam/nữ. Có hai loại hoa văn đó là **RealTexture** và **NormalTexture**, với RealTexture thể hiện các mẫu hoa văn thật, còn NormalTexture thể hiện những màu sắc đơn.



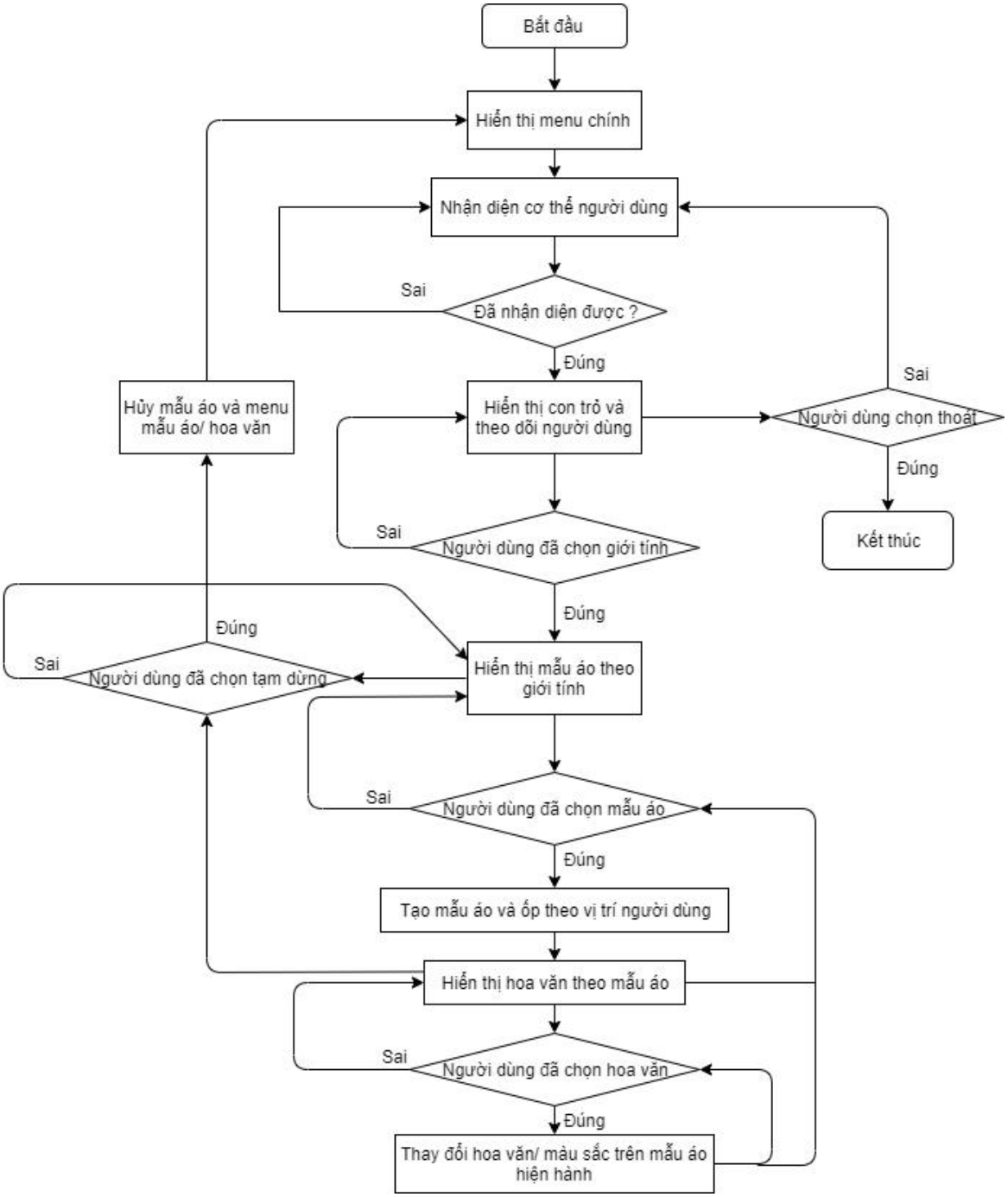
Hình 27. Script quản lý các mẫu áo



Hình 28. Real Texture (bên trái) và Normal Texture (bên phải)

3. Lưu đồ giải thuật của hệ thống

Dưới đây là lue đồ thể hiện giải thuật của đề tài luận văn:



Hình 29. Lưu đồ giải thuật của hệ thống

4. Nhận dạng giọng nói

Chức năng nhận dạng giọng nói của đề tài luận văn được thực hiện ở dạng Command Mode - Chế độ câu lệnh. Tức là người sử dụng nói một câu lệnh ngắn, ví dụ "Start" hoặc "Stop" thì hệ thống sẽ lắng nghe câu lệnh này để thực thi chức năng cần thiết.

4.1 Unity Phrase Recognition Engine

Hệ thống nhận dạng giọng nói của Unity có hai module chính:

- Acoustic Model
- Language Model

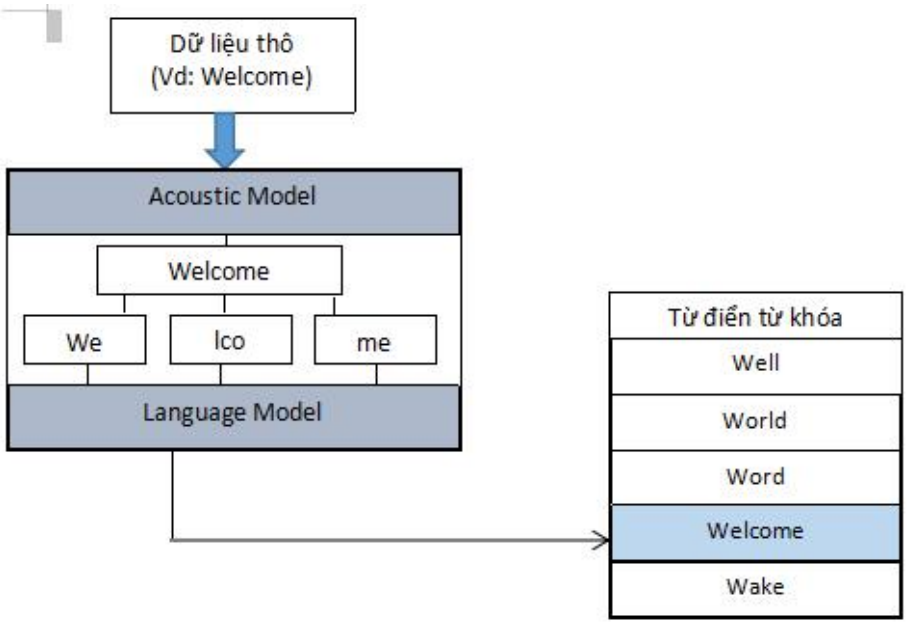
Quá trình nhận dạng giọng nói được diễn ra như sau:

- Microphone nhận vào các tín hiệu âm thanh vào chuyển đổi âm thanh từ dạng analog thành sóng kỹ thuật số để máy tính có thể hiểu được. Ở bước này, microphone của Kinect đã được tích hợp và thực hiện được.
- Sau đó, tín hiệu audio đưsợc gửi đến hệ thống Recognition Engine để nhận dạng.
- Thành phần acoustic model của hệ thống nhận dạng có tác dụng chia nhỏ tín hiệu audio này nhiều thành phần âm thanh riêng lẽ. Sau đó, thành phần language model của hệ thống sẽ phân tích các thành phần này, so sánh từng từ với từ điển được định nghĩa để nhận ra và hiểu được lệnh yêu cầu từ người dùng.
- Engine tính toán và so sánh các từ nhận được và ra quyết định xem những từ này có trong từ điển từ khóa cần nhận diện không. Nếu không, nó sẽ bỏ qua tín hiệu âm thành này và không làm gì cả.
- Cuối cùng hệ thống sẽ chuyển các lệnh này thành các hành động điều khiển.

Engine của Unity là một hệ thống nhận dạng nhạy cảm (sensitve system). Nên Engine có thể giảm thiểu được việc nhận dạng sai các từ có cách phát âm gần giống nhau như "their" và "there".

Hiện nay, Engine này của Unity hỗ trợ được 10 ngôn ngữ trên thế giới (không có tiếng Việt). Vì vậy để dễ tiếp cận, em lựa chọn lập trình với ngôn ngữ tiếng anh.

Dưới đây là sơ đồ khối miêu tả quá trình nhận dạng giọng nói:



Hình 30. Quá trình nhận dạng giọng nói

4.2 Cài đặt Phrase Recognition Engine

Để có thể sử dụng được Engine, ta cần thêm namespace UnityEngine.Windows.Speech vào trước chương trình.

Sau khi chương trình được khởi động, hệ thống cũng đồng thời khởi động keywordRecognizer - đây là hàm luôn luôn được ở trạng thái "bật" để luôn sẵn sàng lắng nghe câu lệnh điều khiển từ người sử dụng.

Sau khi một lệnh điều khiển đã được nhận dạng, hệ thống sẽ tiến hành xử lý nó bằng các sự kiện điều khiển do em định nghĩa.

4.3 Định nghĩa các lệnh điều khiển

Để hệ thống có thể hiểu được giọng nói của người sử dụng, em định nghĩa ra các lệnh điều khiển có sẵn - gồm một tập hợp các từ khóa mang ý nghĩa điều khiển. Khi người dùng nói một câu có các từ khóa (keyword) này, hệ thống sẽ chia nhỏ câu đó ra và tiến hành tìm kiếm xem từ khóa có trong câu không. Nếu có, hệ thống sẽ truyền từ khóa nhận được cho hàm xử lý sự kiện, nếu không, hệ thống sẽ bỏ qua không làm gì cả. Trong khuôn khổ đề tài luận văn, ta có hai loại điều khiển:

1. **Điều khiển chọn Menu**:

Đây là khi người sử dụng được hỏi xem muốn hê thống hiển thị ra các mẫu quần áo cho nam hay cho nữa. Menu bao gồm 3 nút là Female, Male và Stop. Vì vậy, các keyword được dùng để định nghĩa cho chức năng này là:

- Female: Khi người dùng nói "Female", hệ thống sẽ hiển thị các mẫu quần áo của phụ nữ.
- Male: Khi người dùng nói "Male", hệ thống sẽ hiển thị các mẫu quần áo của nam giới.
- Stop: Khi người dùng nói "Stop", hệ thống sẽ dừng và trở về menu chính.
- Exit: Khi người dùng nói "Quit", hệ thống sẽ dừng và thoát ứng dụng.

2. **Điều khiển chọn màu sắc và chất liệu mẫu áo:**

Đây là chức năng điều khiển khi người sử dụng đang trong trạng thái "thử" quần áo. Các keyword được sử dụng cho chức năng chọn màu sắc là:

- Color black: chuyển màu quần áo sang màu đen
- Color blue: chuyển màu quần áo sang màu xanh dương
- Color green: chuyển màu quần áo sang màu xanh lá cây
- Color red: chuyển màu quần áo sang màu đỏ
- Color white: chuyển màu quần áo sang màu trắng.

Các keyword được sử dụng cho chức năng chọn chất liệu là:

- Texture jeans: chuyển chất liệu quần áo sang jeans
- Texture one: Chuyển chất liệu quần áo sang loại caro 1.
- Texture two: chuyển chất liệu quần áo sang loại caro 2.

Các lệnh điều khiển này được định nghĩa dưới dạng một Dictionary, với key là các từ khóa, và value là các sự kiệu điều khiển được gọi khi hệ thống "nghe" được các từ khóa này.

Không giống với hệ thống điều khiển bằng cử chỉ, người dùng khi đang ở menu quần áo của Female phải chọn Stop để trở về Menu chính, và từ đó đổi sang menu quần áo của Male; thì ở hệ thống điều khiển bằng giọng nói, người dùng có thể chuyển đổi trực tiếp từ Menu chính - Menu Female - Menu Male bằng cách sử dụng giọng lệnh.

4.4 Định nghĩa các sự kiện điều khiển

Các sự kiện điều khiển thực chất là các hàm. Các hàm này có chức năng tác động lên các thành phần của giao diện người dùng để chuyển đổi các Menu cho phù hợp.

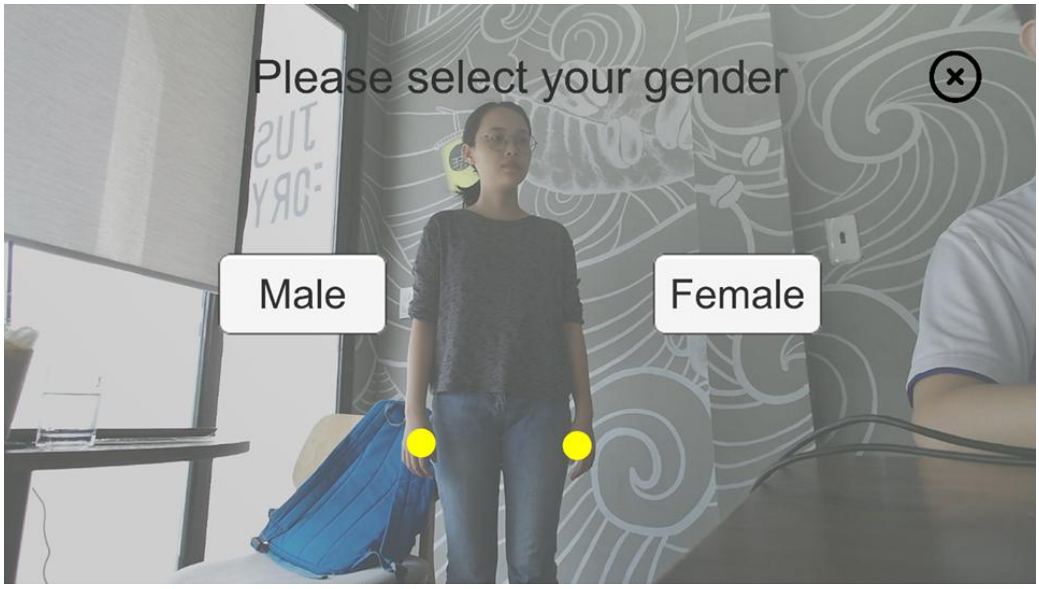
- Đối với chức năng điều khiển Menu chính: Các sự kiện điều khiển là các hàm có chức năng trigger sự kiện click của các nút Female, Male và Stop. Sau khi các nút này được click, giao diện người dùng sẽ được chuyển đổi sang Menu quần áo nam/nữ hoặc quay về menu chính (nếu gọi Stop).

- Đối với chức năng chọn màu sắc và chất liệu: Các sự khiện điều khiển hàm với chức năng gán giá trị mới cho thành phần Real Texture. Khi một chất liệu hoặc màu sắc mới được chọn, chất liệu cũ sẽ được “hide” - giấu đi.
- Riêng đối với chức năng thoát (Exit): Hàm xử lý sẽ gọi thẳng thủ tục thoát ứng dụng.

CHƯƠNG 3: KIỂM THỬ VÀ ĐÁNH GIÁ

1. Giới thiệu giao diện hệ thống

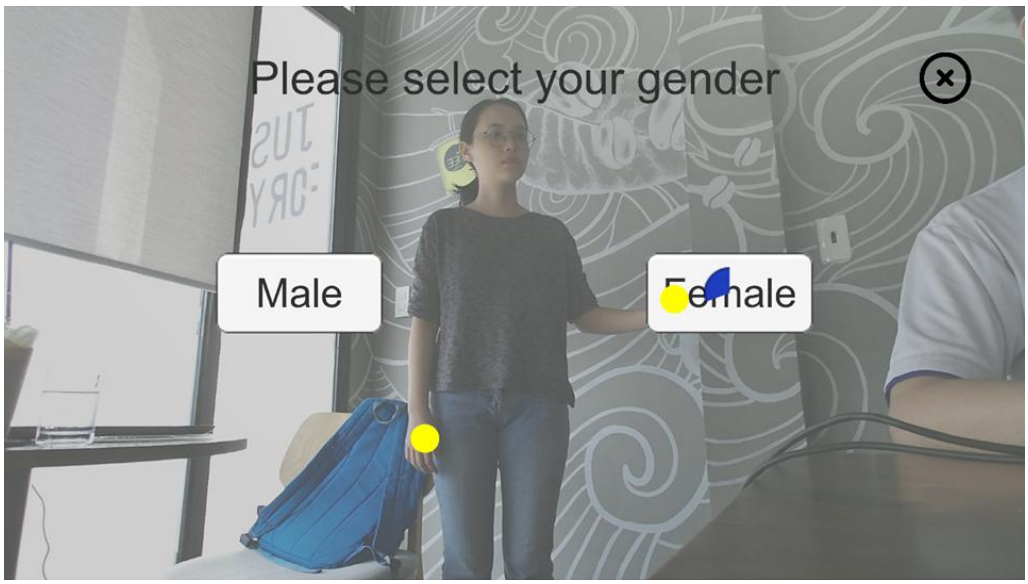
1.1 Giao diện chính



Hình 31. Giao diện chính

Khi hệ thống được khởi động, màn hình chính sẽ được hiển thị đầu tiên. Ở màn hình này, hệ thống hỏi giới tính người dùng để hiển thị trang phục phù hợp. Các thành phần gồm:

- (1) Male, hiển thị các mẫu áo nam.
- (2) Female, hiển thị các mẫu áo nữ.
- (3) Exit, thoát chương trình.
- (4) Text box.
- (5) Kinect displaying plane, hiển thị luồng video từ camera RGB của Kinect.



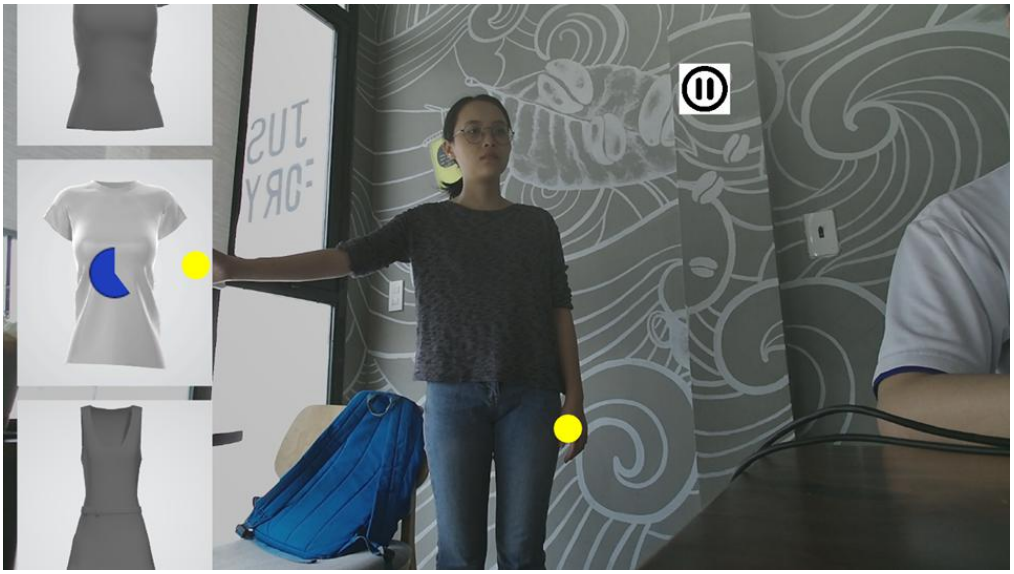
Hình 32. Nút Female được chọn

1.2 Giao diện mẫu trang phục nam và nữ (Female Menu)

Sau khi trả lời giới tính là nam (Male) hay nữ (Female), hệ thống sẽ hiển thị giao diện trang phục.

Giao diện gồm có:

- Danh sách các mẫu trang phục dạng thanh cuộn: Người dùng chọn trang phục bằng các đưa tay vào vùng hình ảnh của trang phục đó trên màn hình.
- Nút Pause: Khi chọn nút này, hệ thống trở về màn hình chính. Có thể chọn bằng con trỏ bàn tay hoặc nói "Stop".



Hình 33. Người dùng chọn mẫu áo

Menu áo nữ được hiển thị tương ứng, người dùng tiếp tục chọn mẫu áo

1.3 Giao diện màu sắc/ chất liệu

Khi người dùng chọn một trang phục, giao diện màu sắc hoặc chất liệu mà trang phục đó có sẽ hiện ra. Giao diện là danh sách các màu sắc / chất liệu được trình bày bằng hình ảnh.

Người dùng có thể chọn một màu sắc/ chất liệu nào đó bằng con trỏ bàn tay (đưa tay đến vùng có hình ảnh của màu sắc/ chất liệu đó) hoặc nói tên của màu sắc / chất liệu.

Đối với màu sắc, người dùng có thể nói: black (đen), blue (xanh dương), green (xanh lá), red (đỏ), white (trắng).

Đối với chất liệu, người dùng có thể nói: jean (vải jean), mode one (vải sọc bé, chéo), mode two (vải sọc to).

Hình 34. Mẫu áo đã được chọn

Hình 35. Mẫu áo được chọn và đổi chất liệu (Texture)

Hình 36. Đổi màu cho áo

2. Đánh giá kết quả kiểm thử

2.1 Giao diện

Sau khi chạy thử hệ thống, chúng em nhận thấy giao diện được thiết kế hợp lý, đơn giản. Người dùng có thể dễ dàng làm quen để điều khiển hệ thống. Kết quả áp mẫu quần áo được hiển thị trực quan, đủ to để người dùng có thể dễ dàng quan sát.

Hệ thống cử chỉ và câu lệnh dùng cho điều khiển dễ phát âm và dễ thuộc nên người sử dụng cũng tiếp cận hai cách điều khiển này nhanh chóng và tương tác với hệ thống tự nhiên.

Về chương trình, chương trình chạy mượt trên nền Unity, thao tác chọn lựa mẫu quần áo và các nút được xử lý nhanh chóng.

2.2 Nhận dạng khung xương

Với phần kiểm thử chức năng nhận diện khung xương, em sẽ kiểm tra khả năng nhận diện và theo dõi hai bàn tay của người dùng.

Con trỏ	Số lần thực hiện	Số lần phát hiện đúng	Tỉ lệ
Tay trái	50	45	90%
Tay phải	50	46	92%

Bảng 2. Kiểm thử nhận dạng khung xương

Thao tác/cử động nhẹ với tư thế bình thường:

Con trỏ	Số lần cử động	Số lần đúng vị trí	Tỉ lệ
Tay trái	50	47	94%
Tay phải	50	47	94%

Bảng 3. Thao tác/cử động nhẹ với tư thế bình thường

Thao tác với tư thế lạ như bắt chéo tay, khoanh tay...:

Con trỏ	Số lần cử động	Số lần đúng vị trí	Tỉ lệ
Tay trái	20	5	25%
Tay phải	20	6	30%

Bảng 4. Thao tác với tư thế lạ

2.3 Nhận dạng giọng nói

Vì quá trình nhận dạng sử dụng microphone từ Kinect Camera nên chất lượng tín hiệu bị ảnh hưởng đáng kể, dẫn đến kết quả nhận dạng thay đổi nhiều khi người dùng đứng gần và đứng xa camera.

Dưới đây là kết quả kiểm thử chức năng nhận dạng giọng nói khi người dùng đứng ở hai vị trí có khoảng cách khác nhau:

Khoảng cách	**Số lần thử**	**Độ chính xác**	**Nội dung**
Dưới 1m	50	92.12%	Female
1m đến 1.5m	50	70.5%	
Dưới 1m	50	90%	Male
1m đến 1.5m	50	66.6%	
Dưới 1m	50	82%	Hai nút Stop/Exit
1m đến 1.5m	50	73%	
Dưới 1m	50	82.3%	Màu sắc/ chất liệu
1m đến 1.5m	50	65.5%	

Bảng 5. Nhận diện giọng nói

Về tốc độ xử lý, nhìn chung tốc độ xử lý nhanh, tốc độ trung bình là khoảng 0.9s. Dưới đây là thống kê về tốc độ xử lý sau 10 lần thử:

Thời gian xử lý trung bình	Số lần thử	Nội dung
0.96s	20	Nút Stop / Exit
0.92s	20	Nút Female
0.82s	20	Nút Male
0.9s	20	Màu sắc / Chất liệu

Bảng 6. Thời gian nhận diện giọng nói

Như vậy, quá trình nhận dạng diễn ra khá chính xác (ở cả hai trường hợp khoảng cách, độ chính xác đều trên 60%). Tuy nhiên, khi ở khoảng cách xa, độ chính xác bị giảm đáng kể so với khi ở khoảng cách gần. Đồng thời, khi ở khoảng cách xa, một số trường hợp camera không thể nghe được giọng nói. Đặc biệt, khi kiểm thử ở môi trường có nhiều tiếng ồn như quán cafe, camera cũng không thể nghe được và phân biệt được giọng nói người dùng và tạp âm. (Mặc dù đã sử dụng hệ thống Microphone Array của Kinect, nhưng tín hiệu thu được vẫn khá yếu). Tuy nhiên trong một môi trường tương đối yên tĩnh và ở khoảng cách thích hợp (từ 0.9 đến 1.2m), quá trình nhận dạng giọng nói diễn ra nhanh và chính xác.

PHẦN KẾT LUẬN

1. Kết luận

1.1 Kết quả đạt được

Về phần kết quả đề tài, ứng dụng "Phòng thay đồ thực tế tăng cường sử dụng camera Kinect 2" đã thành công xây dựng được một hệ thống Phòng thử quần áo ảo, thực hiện được đúng các chức năng được đề ra trong mục tiêu ban đầu của đề tài với độ chính xác tương đối cao và thời gian tính toán nhanh. Cụ thể, hệ thống hiện tại có các ưu điểm sau:

- Nhận dạng đúng khung xương ngườ.
- Ốp đúng vị trí mẫu quần áo vào khung xương.
- Nhận dạng được cử chỉ và giọng nói của người dùng để điều khiển hệ thống.
- Mẫu quần áo có thể xoay theo hành động người dùng, có thể thay đổi kích thước to/nhỏ khi người dùng tiến gần/ đi ra xa so với camera.
- Trượt được menu, tạo được các nút trên menu và chọn được các nút bằng cử chỉ và giọng nói.
- Thời gian phản hồi và tính toán nhanh.

1.2 Hạn chế

Đề tài còn một số hạn chế sau do thời gian thực hiện hạn chế:

- Chỉ cho phép một người dùng tại một thời điểm
- Quản lý mô hình thủ công, chưa tự động.
- Ứng dụng chỉ mới ốp và xoay được mẫu áo chứ chưa cử động tay áo theo tay người dùng.
- Số lượng mẫu áo còn hạn chế do việc tạo ra mô hình ba chiều khá khó khăn, chưa thử được mẫu quần.
- Tín hiệu âm thanh thu được từ camera Kinect khá yếu, gây khó khăn trong quá trình nhận dạng giọng nói.
- Chưa thực hiện nhận dạng giọng nói bằng Tiếng Việt.

2. Hướng phát triển

Ứng dụng "Phòng thay đồ thực tế tăng cường sử dụng camera Kinect 2" đã thành công trong việc đáp ứng các yêu cầu cơ bản đề ra ở chương 1, tuy nhiên vẫn còn một số hạn chế nêu trên. Trong tương lai, chúng em sẽ tiếp tục tìm hiểu

và khắc phục những điểm yếu này, đồng thời tích hợp thêm chức năng thêm mẫu quần áo tự động, chụp ảnh với phông nền tùy chọn, có thêm giỏ hàng và quét mã QR để thanh toán cũng như chia sẻ hình ảnh, nghiên cứu để chức năng nhận dạng giọng nói có thể được thực hiện bằng tiếng Việt.. Thêm vào đó, có thể áp dụng kiến thức 3D để làm cho các mẫu quần áo thêm sinh động.

TÀI LIỆU THAM KHẢO

[1] Cheng, Ching-I., et al. "A 3D Virtual Show Room for Online Apparel Retail Shop." Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference. Asia-Pacific Signal and Information Processing Association.

[2] Min, Shi, Mao Tianlu, and Wang Zhaoqi. "3D interactive clothing animation." 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). Vol. 11. IEEE, 2010.

[3] Kotan, Muhammed, and Cemil Öz. "Virtual dressing room application with virtual human using Kinect sensor." Journal of Mechanics Engineering and Automation 5 (2015): 322-326.

[4] Poppe, Ronald. "Vision-based human motion analysis: An overview." Computer vision and image understanding 108.1-2 (2007): 4-18.

[5] Wren, C. R., Azarbayejani, A., Trevor, D., & Pentland, A. P. (1997, 7). "Pfinder: Real-Time Tracking of the Human Body". IEEE Transactions on Pattern Analysis and Machine Intelligence.

[6] Lee, M. W., & Nevatia, R. (2007, 2). "Body Part Detection for Human Pose Estimation and Tracking". Proceedings of the IEEE Workshop on Motion and Video Computing.

[7] Du, H., Henry, P., Ren, X., Cheng, M., Goldman, D. B., Seitz, S. M., & Fox, D. (2011, 9). "RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments". Proceedings of the 13th international conference on Ubiquitous computing.

[8] Malik, Shahzad, and Joe Laszlo. "Visual touchpad: a two-handed gestural input device." Proceedings of the 6th international conference on Multimodal interfaces. ACM, 2004.

[9] Yoon, Juyoung, et al. "Imidazolium receptors for the recognition of anions." Chemical Society Reviews 35.4 (2006): 355-360.

[10] Huang, Yazhou, and Marcelo Kallmann. "Interactive demonstration of pointing gestures for virtual trainers." International Conference on Human-Computer Interaction. Springer, Berlin, Heidelberg, 2009.

[11] Bhuiyan, Moniruzzaman, and Rich Picking. "Gesture-controlled user interfaces, what have we done and what's next." Proceedings of the Fifth Collaborative Research Symposium on Security, E-Learning, Internet and Networking (SEIN 2009), Darmstadt, Germany. 2009.

[12] Das, Prerana & Acharjee, Kakali & Das, Pranab & Prasad, Vijay. (2015). "Voice Speech Recognition System: Text-To-Speech". Journal of Applied and Fundamental Sciences. 1. 2395-5562.

[13] Ali, Awadalla & Eltayeb, Eisa & Abusail, Esra. (2017). "Voice Recognition Based Smart Home Control System. International Journal of Engineering Inventions". Volume 6.

[14] Këpuska, Veton, and Gamal Bohouta. "Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx)." Int. J. Eng. Res. Appl 7.03 (2017): 20-24.

[15] CAO, Zhe, et al. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. arXiv preprint arXiv:1812.08008, 2018.