

3D character and overlay image movement by human body tracking using unity.

by

Ashraful Azim  
16101220

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science

Department of Computer Science and Engineering  
Brac University  
April 2020

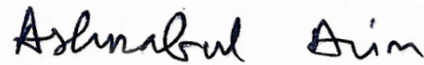
© 2020. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

A handwritten signature in black ink that reads "Ashraf Azim". The signature is written in a cursive style and is centered within a light gray rectangular box.

---

Ashraf Azim  
16101220

# Approval

The thesis/project titled “3D character and overlay image movement by human body tracking using unity.” submitted by

1. Ashraful Azim (16101220)

Of Spring, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on April 18, 2020.

## Examining Committee:

Supervisor:  
(Member)



---

Md. Golam Rabiul Alam, PhD  
Associate Professor  
Department of Computer Science and Engineering  
BRAC University

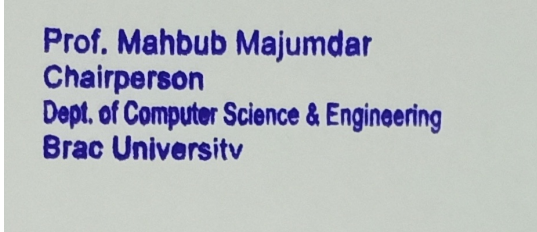
Program Coordinator:  
(Member)



---

Md. Golam Rabiul Alam, PhD  
Designation  
Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)



**Prof. Mahbub Majumdar**  
**Chairperson**  
**Dept. of Computer Science & Engineering**  
**Brac University**

---

Mahbubul Alam Majumdar, PhD  
Professor  
Department of Computer Science and Engineering  
Brac University

## Ethics Statement

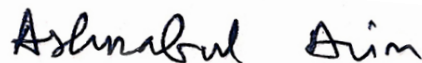
Hereby, I, Ashraful Azim consciously assure that for the "3D character and overlay image movement by human body tracking using unity" the following is fulfilled:

1. This paper is the authors' own original work, which has not been previously published elsewhere.
2. The paper is not currently being considered for publication elsewhere.
3. The paper reflects the authors' own research and analysis in a truthful and complete manner.
4. The paper properly credits the meaningful contributions of co-authors and co-researchers.
5. The results are appropriately placed in the context of prior and existing research.
6. All sources used are properly disclosed (correct citation). Literally copying of text must be indicated as such by using quotation marks and giving proper reference.
7. All authors have been personally and actively involved in substantial work leading to the paper and will take public responsibility for its content.

The violation of the Ethical Statement rules may result in severe consequences.

We agree with the above statements and declare that this submission follows the policies of BRAC UNIVERSITY as outlined in the Guide for Authors and in the Ethical Statement.

**Corresponding Author's Full Name Signature:**



---

Ashraful Azim  
16101220



## Abstract

Technology has made our life very easy and we have the touch of technology everywhere. By using technology, we are solving various difficult problems and making our life more convenient. In our daily life, we buy different clothing and ornaments from different shopping malls. In order to know if a dress is appropriate for the person properly, he or she needs to try the product before buying the product. But the process of trying a product in a big busy shopping mall is time-consuming and hectic for the customers. Often times, people find the trial room occupied and there are big queues in front of the trial rooms. So, we are planning to make the customer's shopping experience hassle-free and convenient. For this reason, we come up with an idea of a smart mirror which will use overlay 3D image fitting techniques that fits overlay 3D images like dress, ornament or other user defined images within optimal space for various human body shape and type. For this, we are using human body tracking technology to first track the skeleton of the body and then take the coordination of the skeleton to move the 3D dress or object accordingly. So, the customer can select a dress and virtually try all the products without any hassle. This system will save the customer's valuable time and they do not need to face the hassles of traditional trialing system. We also know that online shopping is growing its popularity day by day. By having this device at home, one can easily try dress they are about to buy from online. Before buying the dress he or she can easily check if the dress is appropriate for him or her by trying the product virtually without trying it physically. The shopkeepers too will be benefited from this product. During seasons like various religious vacations, shopkeepers are unable to provide enough trial room for the customers. Online shopkeeper can increase customer satisfaction as the customer can now try the product virtually from their residence. We believe that it is a great innovation that brings revolution in the customer's shopping experience. This body tracking technology can be used in various other purposes also. 3D object movement is used in many fields. In gaming and film industry some games require tracking of human movement to move the 3D object accordingly.

**Keywords:** Human Body Tracking, Smart Dressing Mirror, Smart Shopping Mirror, 3D object movement using body tracking.

## **Dedication**

I dedicate my work to my parents. This research would have been impossible without their support and love. Through every difficulty I had them as my guardian angel all the time.

Then, I dedicate my work to Brac University for providing me with all the facilities that I need and facilitating me with the state of the earth computer labs, printing lab etc. I would like to thank our honorable department chair for ensuring we get the best support and service from the University.

Finally, I would dedicate my work to my honorable supervisor, Md. Golam Rabiul Alam, PhD. Without his constant supervision this would not have come to life. In every step of my work I got huge inspiration from him.

## Acknowledgement

All praise belongs to the Almighty Allah, who gave me the strength to do this research and giving me the knowledge and patience required for my work. I thank the Almighty for giving me good health so that I could continue my work. I acknowledge the support of my beloved faculty, Iftexharul Mobin, PhD. for inspiring me to commence this research in the first place. I learned a lot from him and his initial supervision helped me a lot to understand the basics of the research and how I should work to successfully complete this work.

I strongly acknowledge the support of my supervisor Md. Golam Rabiul Alam, PhD. for his immense help and mentoring thought the whole research period for which I could complete the research on time. I learned a lot from him in this time while conducting the research. I learned a lot of new technical thing, tools and technologies which helped me to easily complete my research. I learned the math and science behind my work which was very important for me to conduct the research. I also learned from his integrity, his passion and dedication for his work, and his sincerity for his research. Without his guidance this research would not have been in this position.

I also acknowledge all of my faculty members who helped me time to time for this research and I also acknowledge all of my peers to help my out whenever it was necessary. I acknowledge all the support from the university for facilitating me with top of the class computer labs and other research materials. I am thankful to my parents to support me all the time for this research.

# Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x
Nomenclature	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Challenges . . . . .	2
1.3 Goals . . . . .	3
1.4 Research Methodology . . . . .	3
<b>2 Human Body Tracking</b>	<b>4</b>
2.1 Pose estimation . . . . .	4
2.2 Top-Down Approach . . . . .	4
2.3 Bottom-Up Approach . . . . .	5
2.4 Part Affinity Fields . . . . .	5
2.5 Architecture . . . . .	5
2.6 From Body Parts to Limb . . . . .	7
2.7 Limb to Body Model . . . . .	7
<b>3 Video Input Setup</b>	<b>9</b>
3.1 Unity Environment . . . . .	9
3.2 Video Input . . . . .	9
3.3 Using Web Cam . . . . .	10

<b>4</b>	<b>3D object movement</b>	<b>11</b>
4.1	3D Character Movement . . . . .	11
4.2	3D Dress fitting . . . . .	14
4.3	Simulations . . . . .	15
<b>5</b>	<b>Result</b>	<b>16</b>
5.1	Accuracy . . . . .	16
5.2	Performance . . . . .	17
<b>6</b>	<b>Future Work</b>	<b>18</b>
6.1	Facial recognition . . . . .	18
6.2	Real Time Hair Simulation . . . . .	18
6.3	Real Time Cloth Simulations . . . . .	18
	<b>Bibliography</b>	<b>21</b>

# List of Figures

2.1	Architecture . . . . .	6
2.2	Demonstration of real picture inference . . . . .	6
2.3	Skeleton . . . . .	7
3.1	WebCam Code . . . . .	10
4.1	Position 1 . . . . .	12
4.2	Position 2 . . . . .	13
4.3	Position 3 . . . . .	13
4.4	Position 4 . . . . .	14
4.5	Dress Fit . . . . .	15
5.1	Missing Points . . . . .	16

# List of Tables

5.1	Performance Table in FPS . . . . .	17
5.2	Performance Table in Resource Usage . . . . .	17

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

<i>CPU</i>	Central Processing Unit
<i>DSLR</i>	Digital Single-Lens Reflex
<i>GPU</i>	Graphics Processing Unit
<i>ONNX</i>	Open Neural Network Exchange
3D	3 Dimension
3D-TOF	3D time of flight
CNN	Convolutional Neural Network
FPS	Frames Per Second
mp4	MPEG-4(Moving Picture Experts Group)
MTCNN	Multi-task Cascaded Convolutional Networks
NN	Neural Network



# Chapter 1

## Introduction

### 1.1 Background

The modern age with vast population forced humanity to face many challenges that were not an issue for our previous generation. To cope up with this vast population we have to invent newer technology that can help us to make our life convenient and hassle free. Technology has brought the shopping experience of customers to a different level. By the blessing of technology, our life has been changed. Likewise, the smart mirror is a new innovation of technology, which provides a new way to try on clothing without getting undressed.[9] Nowadays, the sellers give more importance to the customer's choice and they want to make the customer's shopping experience more convenient. In a trial room, the customers find many hassles to check for different products. Even, sometimes they choose many products so that it is tough to trial all the things. Moreover, in our over-populated country, we always need to maintain a long queue to trial a dress. As a result, it is a time-consuming matter for every customer. In addition to that, most of the showrooms are small in size. So, many showroom owners do not like to keep trial rooms in their tiny showrooms. They think that in the space of trial rooms, they can display more products. So, my researched system could detect and track the body of a human being and get the coordinates of different parts of his body. Then, it could place an overlay 3D object mostly a dress or ornament on the tracked human body that will look like the person is wearing the dress virtually. The customer will stand in front of the mirror. Then, the camera can capture their image and set it on their chosen products. By this way, the customers can virtually trial many products in a very short time. I tried to keep our mirror user-friendly and simple to use. I believe my idea will bring a huge change in the taste of customer's shopping.

In this paper I proposed a system where the device will take live video of the user and detect his face/eyes/neck or shoulder and place a chosen dress on appropriate position automatically as if the user is wearing the dress. 3D body tracking can be used in other purposes also. Currently, for big budget motion pictures the film makers use different human body tracking technologies. Motion capture is a very popular technology used in military, entertainment, sports and medical application. Usually, a motion capture device or suit is worn by the respective talent and he or she does the acting wearing it. The device takes the coordinates of different body parts and apply the same movement to a 3D model via computer device. Similar kind of technology is also used in gaming machines. Kinect is a very popular device

for motion capturing. We analyze Kinect as a 3D measuring device, experimentally investigate depth measurement resolution and error properties, and make a quantitative comparison of Kinect accuracy with stereo reconstruction from SLR cameras and a 3D-TOF camera.[4] My proposed device can capture the human movement using a single inexpensive webcam or DSLR camera and track the human using onnx and apply the tracking data to a respective 3D model that will move exactly like the person. In medical simulation also this device will be very helpful to simulate a surgery situation or human anatomy or various dissecting practice.

## 1.2 Challenges

The research for any image processing, machine learning or 3D modeling or movement is computationally very expensive. Some models never get enough train to give any data that is feasible enough to use in a real system. For image processing the researcher has to deal with huge amount of data to handle. 3D modelling and simulation is one of the most computationally expensive tasks for any computer scientist. Often times it requires very powerful GPU and CPU to run the algorithms required to complete the simulation. Some simulations require supercomputers to run it. For instance, realistic lighting and shadowing for 3D images often require tremendous amount of computational power to render. For Monster University Pixar Studio had double the size of data server it had in past. that would be considered one of the top 25 supercomputers in the world. The 2,000 computers have more than 24,000 cores. The data center is like the beating heart behind the movie's technology. Even with all of that computing might, it still takes 29 hours to render a single frame of Monsters University, according to supervising technical director Sanjay Bakshi .[5] Using Kinect is a great way to track the human body movement which uses a depth sensor to track different points of human body. Kinect is relatively expensive and the popularity of Kinect is shrinking down day by day because of the advancement of computer vision technology. My system is using a single web cam or DSLR camera to capture the image and process the data to figure out different points in human body like hands and face etc. to apply the coordinates to a 3D object, character or dress that will move according to the coordinates.

Computer vision is not perfect. In some conditions it fails to identify the intended object properly. In my research, I have found while tracking a human if the background of the person is not simple for example it has multiple colors and designs then the machine has a very hard time calculating the coordinates properly even in some case gives wrong coordinates also.

The performance of the system drops dramatically if the 3D model has hair simulation, cloth simulation or other kind of complex simulation. With the increase of the resolution the frame rate also drops a lot. It seems, a very powerful computational machine is required to run the system properly. But, as technology advances very computationally powerful machine are becoming available which is also inexpensive and cost effective.

For the tracking of human body, I am using a neural network model that can track different human body parts from a given video. The model that I am using can track simultaneously 1 and 1 person only. From my research I have found that tracking of human body and applying it to a 3D model is already computationally expensive enough let alone 2 or 3 bodies. So I have decided to stick to this model for this

research.

## 1.3 Goals

As population is growing more and more shopping malls are trying to use technology to get rid of trial rooms and use virtual dressing room instead of physical dressing room. Using 3D models in entertainment business is also booming. Walt Disney Animation Studios which began as the feature animation department of Walt Disney Productions, producing its first feature-length animated film Snow White and the Seven Dwarfs in 1937 and as of 2020 has produced a total of 58 feature films. Conventional films are also using 3D models that needs human body tracking data. Games with human body tracking like Dance Central is also growing in popularity. In this paper we will be researching on how to use onnx data for human body tracking to use it in unity, use that body coordinates to move a 3D object like dress or 3D character, the study of how much computational power is needed to move 3D object in real time using this body tracking data, the study of the effect of different background and feasibility of using web cam for human body tracking.

## 1.4 Research Methodology

The task has basically two section, detection of human body, tracking human body and moving the 3D object according to the tracking. For human body tracking various methods are available. The Open Neural Network Exchange Format (ONNX) is a new standard for exchanging deep learning models. It promises to make deep learning models portable thus preventing vendor lock in. Let's look at why that matter for the modern ML/AI developer. I am using an onnx neural network model for this research. While importing this model as asset to unity it will be automatically converted to a NN model to work with. This NN model will give me the required human body coordinates that is required. By using this coordinate, I can move the 3D models accordingly. As a sample 3D character, I am using Unity Chan which is available at the Unity asset store. Unity Chan has its in built hair simulation and cloth simulation applied to it. There are two methods for collecting video data. I can either use a pre-recorded .mp4 file with a human character moving on it, or I can use my web cam or any other external video source. I have used both the options for this research.

# Chapter 2

## Human Body Tracking

### 2.1 Pose estimation

Human body tracking or Pose estimation is one of the most important computer vision problems. Pose estimation is usually used to predict what a person is doing for instance is he or she seating or running or dancing etc. In our case we are using pose estimation technique to get the coordination of different part of the human body and apply that to move a 3D character or object. Pose estimation has interested researchers for a long time. If we know the pose of a human, we can further train machine learning models to automatically infer relative positions of the limbs and generate a pose model that can be used to perform smart surveillance with abnormal behavior detection, analyze mythologies in medical practices, control 3D model motion in realistic animations, and a lot more .[10] Previously, pose estimation has been done by attaching a bright point on the different limbs of human body that can be easily tracked by the camera. If that marker is not possible the pose estimation becomes a bit harder problem. The problem can be simplified in different ways. Using 3D cameras that can detect depth is one of the simplest ways. Using infrared or Kinect also makes it simpler. Detecting pose from still image vs video has different type of complexity. In this research we are using single web cam that can detect human pose from a video.

### 2.2 Top-Down Approach

Pose estimation from an image is relatively an easy task. It is due to the lack of hints from other channels. Different viewpoints from 2 or 3 camera adds additional complexity to the process. As human is concern a same pose sometimes gives different appearance. If the picture has partially occluded scene it is very hard for the model to detect the limbs correctly. In this research we are using pose estimation for a single person as it is enough for our research and less computationally complex gives quite good results in pose estimation.

Pose estimation for multiple person is a very hard challenge. For occluded and human interacting with other human top-down is a very popular approach. In this technique user first train an object detector model which detect all the human being in the scene and then run the pose estimation on each person. Though it seems a good solution but very often the detector fails to detect the human in crowded places and places where several people appear in a single box. If such happens the

whole model can fail easily. Moreover, the complexity increases drastically as the number of persons increases.

## 2.3 Bottom-Up Approach

In the bottom up approach the body pose is recognized by pixel level image evidence. This method solves the occluded limb problem and more than one person in the same box problem. As user have information from the whole picture it can easily differentiate between people. It can also assign different limb to respective person. This method can also become computationally very complex. In worst case it can become NP-hard problem also. But the silver lining here is it should work in any given situation unlike the top down approach.

## 2.4 Part Affinity Fields

This method uses PAF for 2D pose estimation. The approach uses a non-parametric representation, which we refer to as Part Affinity Fields (PAFs), to learn to associate body parts with individuals in the image.[7] Human 2D pose estimation—the problem of localizing anatomical key points or “parts”—has largely focused on finding body parts of individuals [7]. PAF is a bottom up approach which is a set of 2D vector fields encode the orientation and position of the limbs. If we could already detect the body parts like leg, face, hands etc. to generate a pose from it we must connect the two points of the limbs. In each part there are many elements competing to form a limb. There could be more than one people in that image and there could be lots of false positives. For the association between each body parts detection a feature representation called Part Affinity Fields is used that consists data about location and orientation in the area of support of the limb.

In short PAF is a set of vectors encoding direction of one part of the limb to another limb. In case the point lies on a limb then its PAF value is a unit vector directing from starting joint to the ending joint of the limb. The value of the PAF will be zero if it is outside the limb.

## 2.5 Architecture

The main intention here is to concurrently predict detection confidence map and affinity fields. It utilizes a feed-forward network as feature extractor.

This is architecture of the two-branch cnn. Every stage in the upper branch predicts confidence map  $S$  and each stage in the bottom branch predicts PAF  $L$ . Then they concentrated with image features  $F$ . This will be utilized as input for the next stage. As it is divided into two parts, the top one will predict the detection confidence maps and the lower one is for affinity fields. They are organized as an iterative prediction architecture that will improve the success in each stage. Before sending input to this network the method uses auxiliary CNN to extract an input feature map  $F$ .

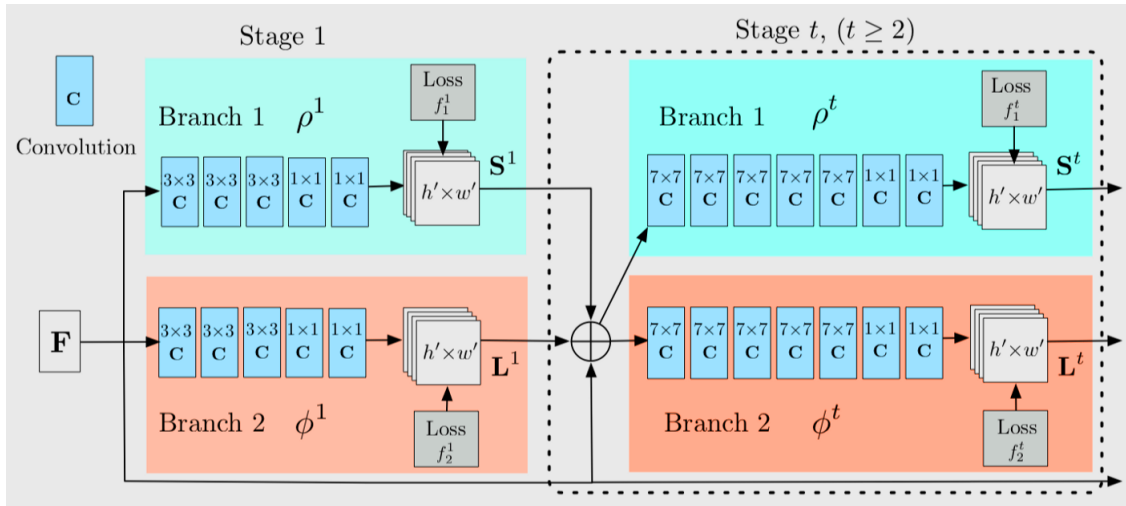


Figure 2.1: Architecture

This prediction is processed by all the branches, and their predictions concatenated with first  $F$  are utilized as input for the next stage. The process is re-run again and again in every stage to get the optimal result.

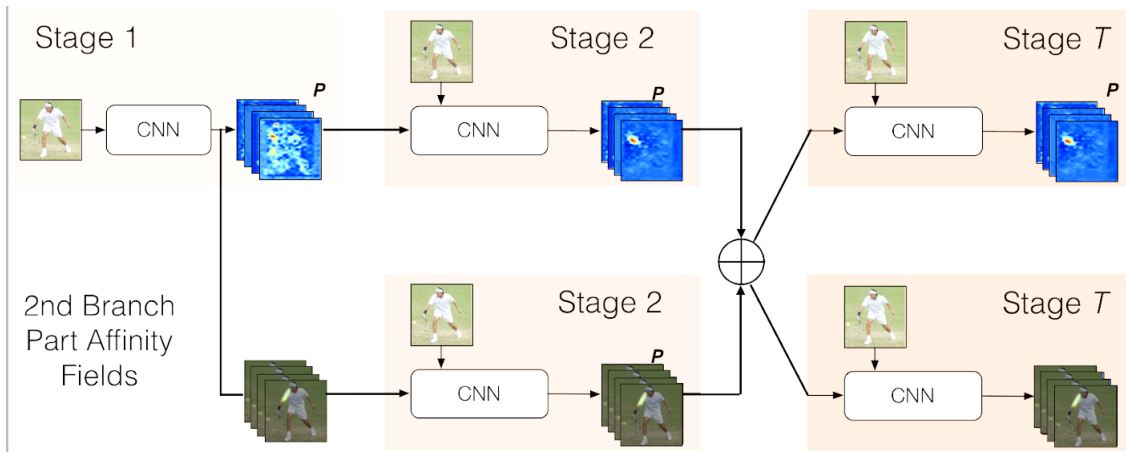


Figure 2.2: Demonstration of real picture inference

## 2.6 From Body Parts to Limb

The Confined maps and PAFs may have confusions in earlier stages but it correct itself in later stages. When each stages ends, the related loss function is used to each branch to guide the whole network. In the upper branch, every confidence map is 2D representation of the confidence that a pixel belongs to a certain organ like elbow or wrist etc. For the body part selection regions, we define confidence maps for different beings. Then the system will perform the non-maximum suppression for obtaining a particular set of parts location.

In that inference, the system computes line integrals of all PAFs in the line segments of pairs of selected body parts. The PAF will consider it as a limb only if the selected limb formed by the connection of defined pair of point aligned with related PAF.

## 2.7 Limb to Body Model

Till now we can understand how the algorithm finds different limbs of the body on the picture between two points. For the pose estimation a full body model is needed. Here, the algorithm was able to only identify the body parts but a logical connection among the body parts is needed to get a meaningful structure from the image. This problem is seen as k-partite graph matching problem. Here, nodes are the different body parts detected and edges are all possible combination of these body parts. k-partite matching implies the vertices could be partitioned into k groups of nodes with no connections inside each group. Edges are weighted with part affinities.



Figure 2.3: Skeleton

A direct solution of this problem is computationally very complex. The proposed model here is relaxation where initial k-partite graph is decomposed into a set of bipartite graphs. Hence, the connecting is now relatively less complex. The decomposition is based on the kind of task it needs to solve. The basic idea is we know

how the body should be connected for example a head can not connect directly the hip. So, it first connects to the shoulder and then shoulder to the hip.



# Chapter 3

## Video Input Setup

### 3.1 Unity Environment

Unity is originally used for game development and augmented reality project. Inside Unity there are two types of view for the developer, one is scene view where the developer can place different 3D objects in the 3D place and apply different physics rule on these objects. There is a camera in the 3D space which implies the viewpoint or in simply the eye of the user or the person playing the game or using the app. The camera can be placed anywhere by the developer. He or she can also apply some camera movement rules so that the use can move the camera which in terms moves the viewpoint of the user. After the scene view there is game view. If switched to game view the developer will see what the user will see using the program. It is simply the view that can be seen by the camera.

### 3.2 Video Input

There is no native video player UI in the Unity engine. However, as some cases playing a video in some scene is necessary there is a video player component to render video as a dynamic texture. By default, the Material Property of a Video Player component is set to MainTex, which means that when the Video Player component is attached to a GameObject that has a Renderer, it automatically assigns itself to the Texture on that Renderer (because this is the main Texture for the GameObject) For creating the render object to play the video on first we create the render texture on the scene mode in the 3D space. Here, we can set the resolution of the vide in the inspector. Then we create the UI canvas, and inside the canvas we create the UI panel. Then inside the panel we create a raw image. I make the raw image size close to the size of the canvas. In the panel we add video player. In the inspector we drag our intended video to the video clip tab. In the target section we drag the original texture that we created. In the raw image we give the texture of the video. When we click the debug icon, we can see in the game view our intended video is playing inside the canvas. In this video we will apply our trained NN model to track the human body and get the coordinates which will be applied to our 3D object.

### 3.3 Using Web Cam

Capturing video from webcam is a bit different from playing video in the canvas. To access the device camera whether it is a webcam, camera connected through USB or phones camera (in case the final product is running on a phone) certain script has to be written. For this we first create a C script. We can edit the script using Microsoft Visual Studio.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraScript : MonoBehaviour
{
    static WebCamTexture backCam;

    void Start()
    {
        if (backCam == null)
            backCam = new WebCamTexture();

        GetComponent<Renderer>().material.mainTexture = backCam;

        if (!backCam.isPlaying)
            backCam.Play();
    }

    void Update()
    {
    }
}
```

Figure 3.1: WebCam Code

The main thing to look here is the WebCamTexture class, we are achieving this with the help of this class. We are making it static and if my object is not null, I am going to initialize it and I am getting a texture over which the script is applied and I am applying my camera texture to that selected texture. If my webcam texture meaning my camera is not already started then start it as shown in the script.

We can then apply the script to any 3D object then we can see the webcam video is playing on that 3D object. Depending on the orientation of the camera the video sometimes has to be flipped or moved upside down. In the inspector view we can do it easily by changing the X or Y or Z position to certain degree. In this project we will also be able to use webcam data to apply our NN model in it to get all the human tracking data and apply it to move the 3D object accordingly.

# Chapter 4

## 3D object movement

### 4.1 3D Character Movement

The main object here is to move a 3D character or dress or any other object according to the tracked human body data. It implies if the human gives a certain pose the character will mimic that pose for instance, if he moves his hand the character will move his hand etc. As discussed earlier to track a human we at first need to train the NN model. The more accurate the model is the more precisely it will track the body and hence move 3D object accordingly. We are using a pretrained (.onnx) file to use as our NN model. We download the latest onnx file from [http://digital-standard.com/threedpose/models/Resnet34\\_3inputs\\_448x448\\_20200212.onnx](http://digital-standard.com/threedpose/models/Resnet34_3inputs_448x448_20200212.onnx). We open the file in Unity which will convert it to NN model. For 3D character I am using Unity Chan which is available at the unity store. There are 24 points that we will be tracking in the Human tracking using Barracuda. These points are listed below-

1. rShldrBend
2. rForearmBend
3. rHand
4. rThumb2
5. rMid1
6. lShldrBend
7. lForearmBend
8. lHand
9. lThumb2
10. lMid1
11. lEar
12. lEye

13. rEar
14. rEye
15. Nose
16. rThighBend
17. rShin
18. rFoot
19. rToe
20. lThighBend
21. lShin
22. lFoot
23. lToe
24. abdomenUpper

These 24 points are tracking points from our human being. This is useful for the skeleton creation as shown in Figure 2. The position of hip, head, neck, spine is already calculated in the NN model. For the rest we need to transform different joint points to arm face and legs. For the character movement we are transforming the position index from the tracking data to the joint points of the 3D character. For example, to get the right shoulder movement position from bone transform of human body bones right upper arm.

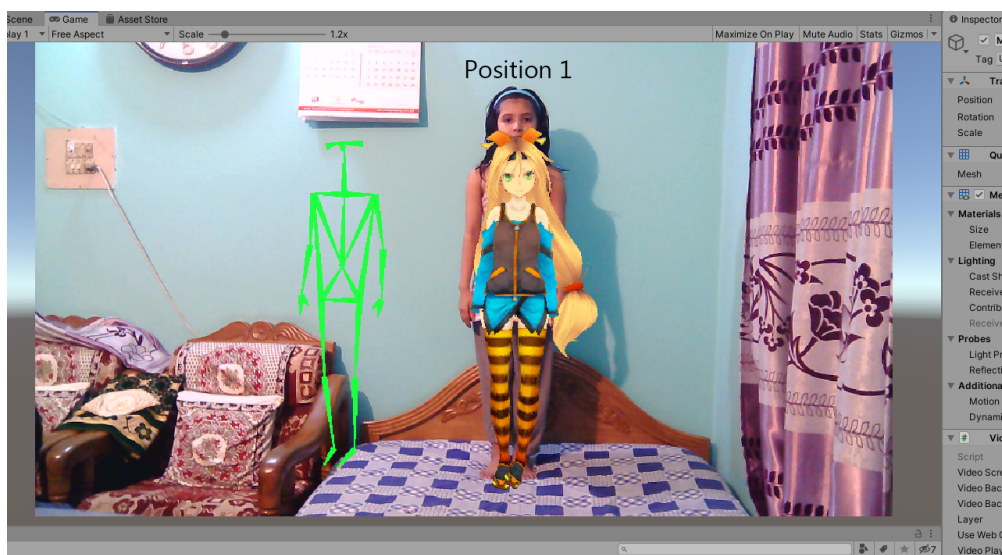


Figure 4.1: Position 1

Here, in figure 4 we can see that as the human is standing straight the 3D character aka the Unity Chan is also standing straight. In figure 5, the hands of the human

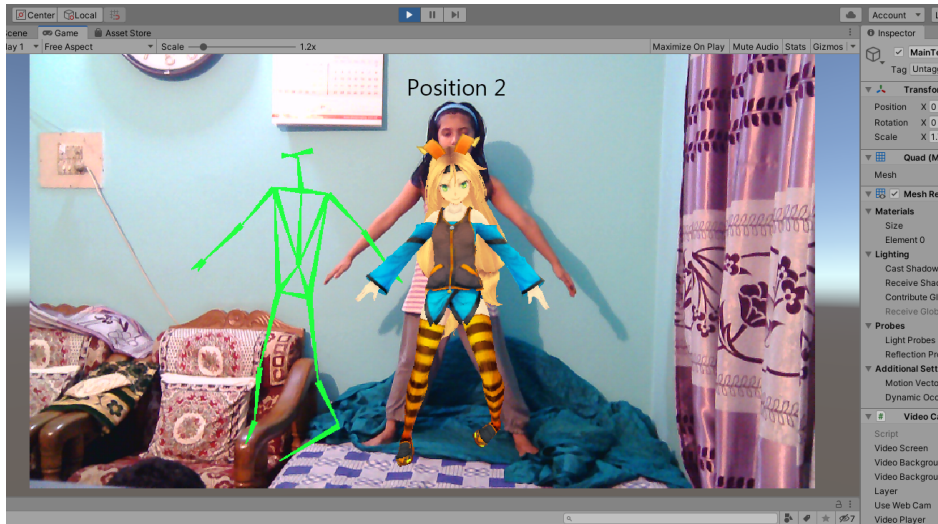


Figure 4.2: Position 2

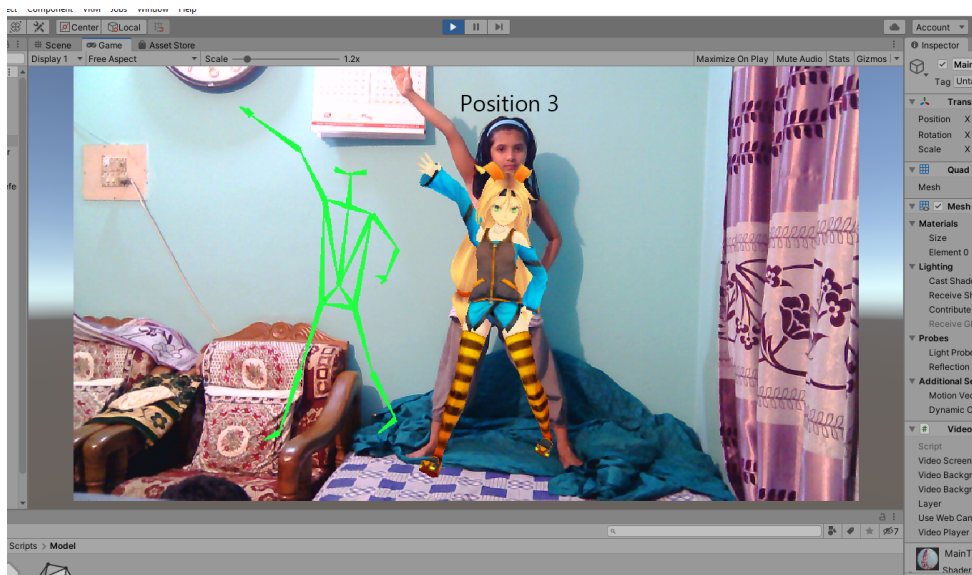


Figure 4.3: Position 3

is a bit wide open and the character is also following that. In figure 6 one arm is raised up so the character also raised up his arm. In figure 7, the human is sitting with wide open arm and the character is also sitting with wide open arm. One thing to be noted here is the purpose here is to move the 3D character according to the pose estimation, it does not have to be fitted appropriately with the person in the pictures which is needed for dress fitting.



Figure 4.4: Position 4

## 4.2 3D Dress fitting

3D dress fitting is different than 3D character movement. For character movement it does not have to be fitted with the picture of the person as it not the intention here. But, for the dress fitting, it is important to that the dress to some extent fits with the picture of the person and also it should move as the person moves his hand or leg etc. Moving the dress according to the human movement has already been done in the 3D character movement. But the dress fitting according to the persons size is a challenge.

When selecting a 3D object in the unity inspector one can easily change the position of the object in X, Y, Z coordinate. The scaling or size of the 3D dress can also be changed. The easiest way to scale the background is to change the value in 'Video Background Scale' The default value here is 1. It could be set from .1 to 1 as per requirement. In this research we tried to match the size of the 3D dresses as close as possible.

As the system of 3D character movement and 3D dress fitting is pretty similar, we are just using 3D dresses instead of character and resized the dress appearance a little bit to get the optimum result.



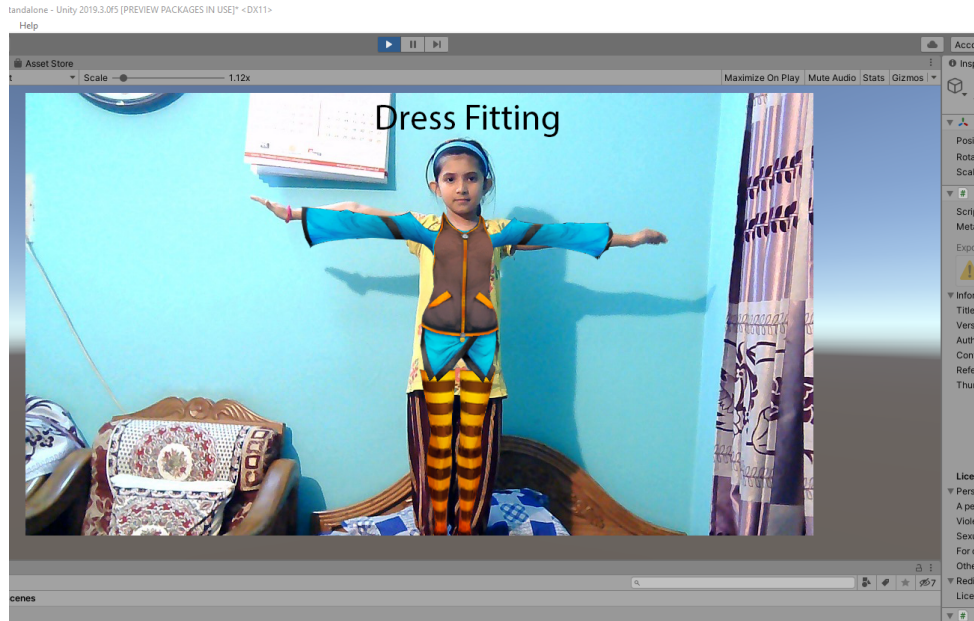


Figure 4.5: Dress Fit

### 4.3 Simulations

Different types of simulations in 3D characters and dresses is computationally very challenging task. The more advanced the simulation is the more it is realistic and the more it becomes computationally expensive. As it is a vast research area, in this research we are using in built hair simulation for the Unity Chan Character and inbuilt dress simulation for both Unity Chan and the 3D dress. For lighting and shadowing we are also using the default Unity light system. Here, we can select the position of the light source which will affect the brightness and shadow condition of the 3D object.

The proposed system is already demanding decent amount of computational power so we are avoiding using additional hair, cloth or lighting simulations.

# Chapter 5

## Result

### 5.1 Accuracy

The accuracy of the system varies upon different types of poses any predominantly on the background of the human body. If the background is complex for instance has multiple color pattern or designs the NN model will have very hard time detecting the pose correctly. Our proposed system can only detect one human being at a time. For our project detecting one person is enough.

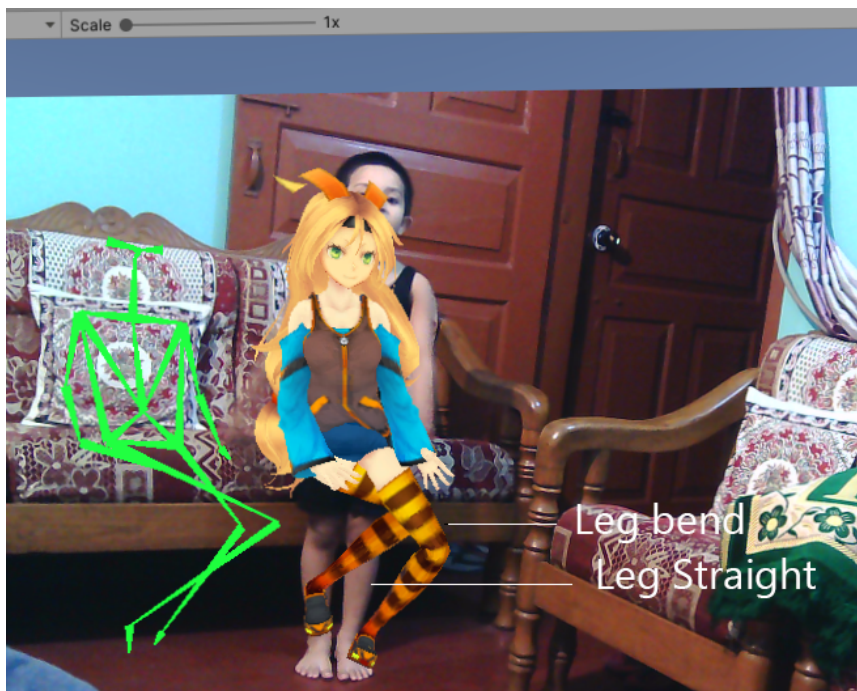


Figure 5.1: Missing Points

In the figure 9, the human is standing still yet the skeleton shows both the legs and hands are bend.

Moreover, if the human is moving very fast the system becomes very slow and the 3D character cannot move accurately. The accuracy also depends on the dress of the human being. If the dress is too baggy or complex, sometimes the algorithm



cannot detect the pose properly.

The ambient lighting condition also plays vital role in the accuracy. Apart from these issues the accuracy of the system is very well and up to the mark. As the 3D dress does not have hair simulations it perform a bit better than the 3D character movement.

## 5.2 Performance

The performance of the system depends on various things. The most common one being the computational power of the system. If we use the high-quality trained model, we get around 30 FPS in RTX2060 SUPER GPU and 20 FPS in GTX1070 GPU. For a low-quality trained model, we get around 60 FPS in RTX2060 SUPER GPU. The CPU usage keeps around 20 percent whereas the GPU usage fluctuates around 90 to 99 percent. The performance also depends on how fast the human moving. Fast moving human moves the 3D character fast which in terms slows the system. If the 3D object moves fast different simulations like clothing simulations, hair simulation becomes very complex as the physics of fast-moving object itself is very complicated.

GPU	Low Quality Trained NN	High Quality Trained NN
RTX2060 Super	60 fps	30 fps
GTX1070	40fps	20fps
Vega11	15 fps	null

Table 5.1: Performance Table in FPS

Resource	CPU	GPU
Low trained model	20 percent	90 percent
High trained model	30 percent	95 percent

Table 5.2: Performance Table in Resource Usage

# Chapter 6

## Future Work

### 6.1 Facial recognition

Human face detection plays an important role in applications such as video surveillance, human computer interface, face recognition, and face image database management.[2] Face recognition is one of the most researched computer vision problems over last two decades. In the literature [3] [8] [6] there are many algorithms used to solve face recognition problem. For example – Eigenface Algorithm (1991), Local Binary Patterns Histograms (LBPH) (1996)

There are at least two reasons for the importance behind the research of face recognition which has recently received significant attention, especially during the past several years. The first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research.[11] Facial recognition can be used to detect different facial features like lips placement, eye placement etc. which can be used to trial virtual lipstick or sunglasses. Facial expression detection is also very important for giving facial express to a 3D character. For example, if the human is crying or laughing the 3D character could also follow the same expression. MTCNN is a very popular tool for face recognition. It uses OpenCV library to read the image or video data. Then it creates embedding points for face using FaceNet.

### 6.2 Real Time Hair Simulation

Simulating and rendering realistic hair with tens of thousands of strands is something that until recently was not possible in real time. Modern GPU has become powerful enough to somewhat realistically simulate hair. Hair simulation is different for different types of hairs. A simulation that works correctly on straight hair will not work for curly hairs. For short hairs the simulation is completely different. Particle constrain method is a very popular method for hair simulation which is extremely parallelizable that is suitable for modern GPU.

### 6.3 Real Time Cloth Simulations

Cloth simulation is also a very popular research area. Cloth simulations is relatively computationally less expensive. The bottle-neck in most cloth simulation systems

is that time steps must be small to avoid numerical instability. [1] In this research the Unity default cloth simulation was good enough. But advanced cloth simulation will make it more visually pleasing.

# Bibliography

- [1] D. Baraff and A. Witkin, “Large steps in cloth simulation”, in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’98, New York, NY, USA: Association for Computing Machinery, 1998, pp. 43–54, ISBN: 0897919998. DOI: 10.1145/280814.280821. [Online]. Available: <https://doi.org/10.1145/280814.280821>.
- [2] Rein-Lien Hsu, M. Abdel-Mottaleb, and A. K. Jain, “Face detection in color images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, 2002.
- [3] A. Senior, R.-L. Hsu, M. A. Mottaleb, and A. K. Jain, “Face detection in color images”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 696–706, May 2002, ISSN: 0162-8828. DOI: 10.1109/34.1000242. [Online]. Available: <https://doi.org/10.1109/34.1000242>.
- [4] J. Smisek, M. Jancosek, and T. Pajdla, “3d with kinect”, in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1154–1160.
- [5] D. Takahashi. (2013). How pixar made monsters university, its latest technological marvel, [Online]. Available: <https://venturebeat.com/2013/04/24/the-making-of-pixars-latest-technological-marvel-monsters-university> (visited on 09/30/2019).
- [6] P. Jaturawat and M. Phankokkruad, “An evaluation of face recognition algorithms and accuracy based on video in unconstrained factors”, in *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2016, pp. 240–244.
- [7] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields”, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1302–1310.
- [8] J. Dhamija, T. Choudhury, P. Kumar, and Y. S. Rathore, “An advancement towards efficient face recognition using live video feed: ”for the future””, in *2017 3rd International Conference on Computational Intelligence and Networks (CINE)*, 2017, pp. 53–56.
- [9] J. Gupta. (2018). How virtual mirror technology will change the way you shop, [Online]. Available: <https://www.quytech.com/blog/how-virtual-mirror-technology-will-change-the-way-you-shop>.
- [10] S. Nikolenko. (2018). Neuronuggets: Understanding human poses in real-time, [Online]. Available: <https://medium.com/neuromation-blog/neuronuggets-understanding-human-poses-in-real-time-b73cb74b3818> (visited on 09/30/2019).

- [11] A. Imran, B. Shams, and N. F. Islam, “Analysis on face recognition based on five different viewpoint of face images using mtcnn and facenet”, 2019.