Virtual Glasses Try-on System

Siyu Quan

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Computer Science

Concordia University

Montreal, Quebec, Canada

January 2016

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Siyu Quan

Entitled: Virtual Glasses Try-on System

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. H. Harutyunyan

_____ Examiner
Dr. T. Fevens

_____ Examiner
Dr. D. Goswami

_____ Supervisor
Dr. S.P. Mudur

_____ Supervisor
Dr. T. Popa

Approved by _____
Chair of Department or Graduate Program Director

_____ 2016 _____
Date Dean of Faculty

Abstract

Virtual Glasses Try-on System

Siyu Quan

Recent advances in data-driven modeling have enabled the simulation of wearing the glasses virtually based on 2D images (web-camera). For real-life glasses wearing, it needs not only suitable for appearance, but also comfort. Although the simulations based on 2D images can bring out certain conveniences for customers who want to try on glasses online first, there are still many challenging problems ahead because of the high complexity of simulations for wearing glasses. Obviously, it can hardly tell if the glasses are comfortable or can seat on the customer's nose correctly. Furthermore, customers may want to take a look from different angle to make sure that the glasses selected are perfect. Such requirements cannot be met by using the simulations based on 2D images so we present an interactive real-time system with simulations for wearing glasses, providing users with a high degree of simulation quality including physics application. With our system the user uses the Kinect sensor or the common web camera as input device to acquire the result of wearing the preferred glasses virtually, which is real-time. Input device captures the user's face and generate a geometry face-mesh which is expected to be aligned to the face-mesh template we pre-set manually. The 3D data captured from Kinect dramatically improve user's experience by constructing geometry face mesh.

Acknowledgements

I would like to express heartfelt thanks to my supervisors, Dr. Sudhir P. Mudur, Dr. Tiberiu Popa for their insightful instructions during the period of my graduate study in Concordia University. They always guide me to the right direction not only in the academic field, but also in research attitude. Their enthusiasm in the work place set impressive examples for me. I would also like to thank Alex Consel for his participation throughout the whole progress of this academic exploration with remarkable efforts and help.

My sincere appreciation also goes to all my colleagues in the 3D graphics lab who give me continuous help and support and let me learn different and new information in their fields.

I would like to thank my parents to who gave me the opportunity to come to Canada to further my graduate studies.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

Virtual glasses versus real life glasses: what are the advantages, the disadvantages and the differences? In this thesis, I will present a new method of people to be able to shop online for glasses without needing to go physically to a store to try these glasses, with the use of virtual try-on system.

Nowadays, virtual try-on of glasses has certain advantages over physical try-on in some cases. Usually, it is time consuming to select a model that looks good on a customer's face. The virtual try-on system encourages online shopping which enjoys greater popularity. In this system, customers are allowed to upload their photos on the website's system which will generate new 2D images with a chosen pair of glasses on customer's face with the proper position. The virtual try-on system can be used to narrow down the selection to a few designs and sizes efficiently and interactively with possible recommendations from an AI-based expert system. Another problem encountered with customers who suffer from weak eye-sight is that they try on glasses with only the frames and without ophthalmic lens. They need to take off their original glasses in order to try on new frames physically, which make the view less than perfect since with near eye-sight people, they cannot see without their prescription. While using virtual try-on system, the customers are not actually wearing the new spectacles, and they also do not need to remove their actual vision-aid glasses, if they are wearing any at the moment, they can see a clear view of face with the chosen new spectacles.

Due to the advantages of virtual try-on as mentioned above, the virtual try-on has a reasonable good commercial potential in the eyewear market. However, there are still certain limitations that need to be addressed. Currently, there are several categories of eyeglasses virtual try-on techniques. The first one is based on 2D images, adding 2D glasses image onto the user's facial image. Technically, this kind of method is relatively simple and the results are acceptable. With this way, it has been applied widely in commercial use. Superficially, while this kind of simulation can make a lot convenience for customers who want to try on the virtual glasses in front of the computer at first, many challenging problems remain because of the high complexity of simulations of wearing virtual glasses. First of all, customers may want to take a look from different angles to make sure that the selected pair of glasses fits perfectly on the face. However, this kind of systems can only deal with a frontal view, and cannot provide dynamic feedback for the user's action.

Another kind of methods combines 3D glasses model with 2D facial images. With 3D glasses models instead of 2D glasses images, those systems can handle multiple-angle views. To be more precise, users are able to see his or her looks from different angles. This kind of methods usually takes videos as input, and good tracking algorithm is needed. Furthermore, customers may also want to experience physical feedback. For example, if the bridge of certain frame is too wide for customers' nose, a feedback would be necessary for the customer when choosing a pair of glasses. This kind of physical feedback should be accurate and precise. Obviously, those systems based on 2D images are not capable of meeting such requirements. They can hardly tell if selected pair of glasses is comfortable or what it looks like from different angles or if they can even be seated on the customer's nose correctly.

Our solution is to make the customer's experience into a reality from the customer's own home without the need to go out into the shops to try on each pair of glasses. Our goal is to make a true virtual online shopping experience in the glasses department. In order to meet all those, we have developed three separate systems which require three different kinds of cameras respectively: Kinect V1, Kinect V2 and common web-camera. Kinect V1/V2[1] builds on software technology developed internally by Rare, a subsidiary of Microsoft Game Studios owned by Microsoft, and on range camera technology by Israeli developer PrimeSense, which developed a system that can interpret specific faces, gestures, making completely hands-free control of electronic devices possible by using an infrared projector and camera and a special microchip to track the movement of objects and individuals in three dimensions. This 3D scanner system called Light Coding employs a variant of image-based 3D reconstruction. By using Kinect V1/V2, we are able to acquire precise 3D data (including depth data) to construct geometry face mesh which is going to be aligned to the face-mesh template that we pre-set manually. In case of users' lack of access to Kinect V1/V2, we develop the system for web-camera as well.

1.2 Main Challenges

Many problems were raised with this work. Here are the following challenges that we encountered during test drives:

During testing, we were concerned with reconstructing and aligning the user's face mesh to the face-mesh template that we pre-set manually. The problem encountered is that given two similar

objects' meshes in their own coordinate space, we have to find the transformations needed to bring these two meshes and virtual glasses into one common coordinate system - Bullet physics world. Given that the systems are real-time, we have to keep conducting this alignment every frame according to movement of the user's face. We assume that the required transformations are rigid body transformations, which means we are going to ignore the user's different expressions.

The available solutions to apply physics simulation can be divided by the different level of details in its specification. In order to do so, we need to take a closer look at the real-time physics simulation which is the suitable simulation in our case. Real-time physics engines, such as Bullet, are capable of a faster operation than the high precision physics engines, although they may have relatively low operational precision. Furthermore, the other factor that affects operational efficiency is the volume of points (triangle faces) which involved the physics simulation. Given that the user's face mesh we reconstruct and the face-mesh template we pre-set are both high resolution mesh, this means the volume of points in each mesh is numerous (See table 1). The question of how to decrease the physics operation time between each frame is another focus in this thesis.

Here are point's volumes and faces volumes of each part of mesh used:

meshes	Point's volumes	Face's volumes
Generic Face	1617	3176
Glass middle frame	7278	14560
Glass left supporter	2530	5056
Glass right supporter	2530	5056

Table 1: mesh volumes

1.3 Significant Contributions

- We propose a real-time 3D virtual glass try-on system which applies generic template geometry to simulate the user's face and creates accurate human-computer interaction.

- It is physically based on a face mesh, which means virtual glasses have accurate contact and physics interaction with template geometry (face mesh).

- It is applicable to a range of cameras, such as Kinect V1, Kinect V2 and web-cams.

- It is customizable which can scan the user's face geometry as face mesh input to improve the try-on results.

1.4 Outline of Thesis

The rest of the thesis is organized as follows:

In Chapter 2, we provide a review of related works in virtual glasses try-on. In Chapter 3, we describe two alignment approaches along with real-time physics solution and what exactly we are trying to achieve by conquering main challenges. In Chapter 4, we present the implementation in details. In Chapter 5, we compare our three system results. In Chapter 6, we

conclude our work by summarizing the advantages and the weakness of our solution, and indicate potential for future work as well.

Chapter 2 Related Works

Since virtual glasses trying on systems have so much commercial potential, there are a number of related approaches that have been deployed already.

2.1 2D Approaches

Framesdirect[2] has been providing a virtual try-on system based on 2D images, which allows customers to upload their personal pictures and choose the frames they like. Then the system will add the wanted frame onto their pictures if a human being face is detected in that picture. (See figure 1) This kind of virtual try-on systems has been applied widely in that it is relatively simple to develop and operate and the results are fast and acceptable.

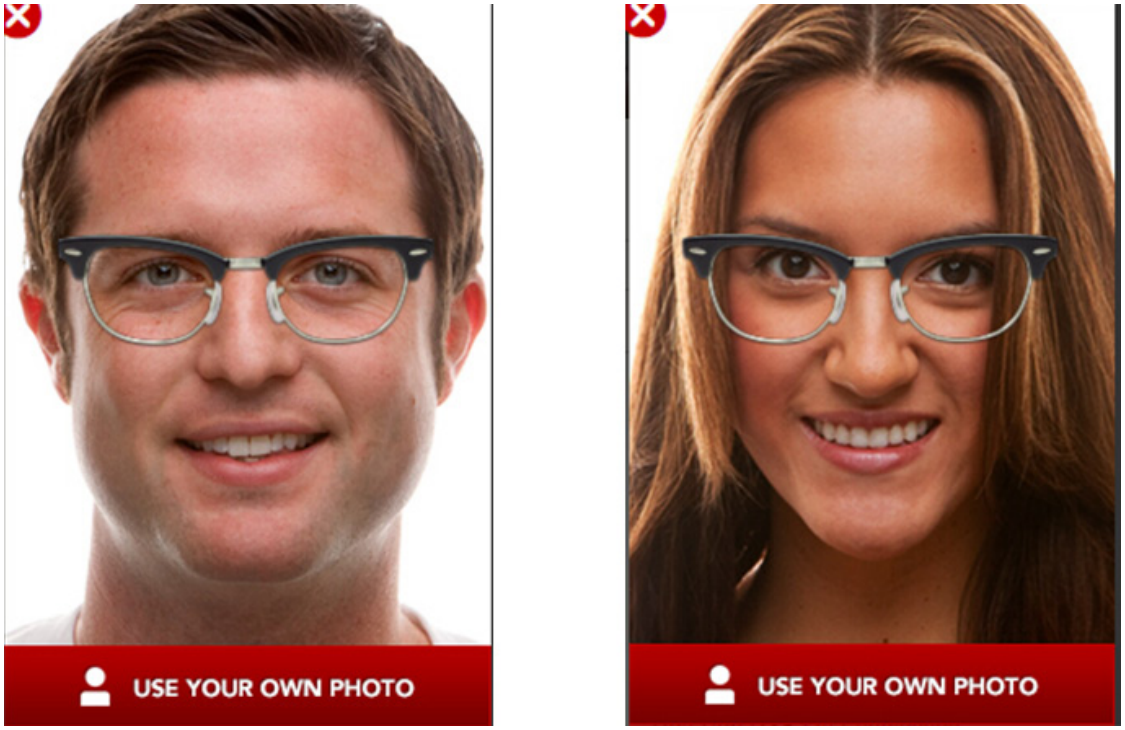

Figure 1: Framesdirect example

However, the drawbacks are obvious. It can only deal with front view; it does not scale the glasses image properly and it cannot provide dynamic feedback for the user's action such as rotation of the user's head.

In [3], they adopt a cascaded AdaBoost classifier to detect the eye corners from face image sequence. And then, based on the face pose and eye location, fit the glasses image to the eye area using affine transform[4]. (See figure 2)

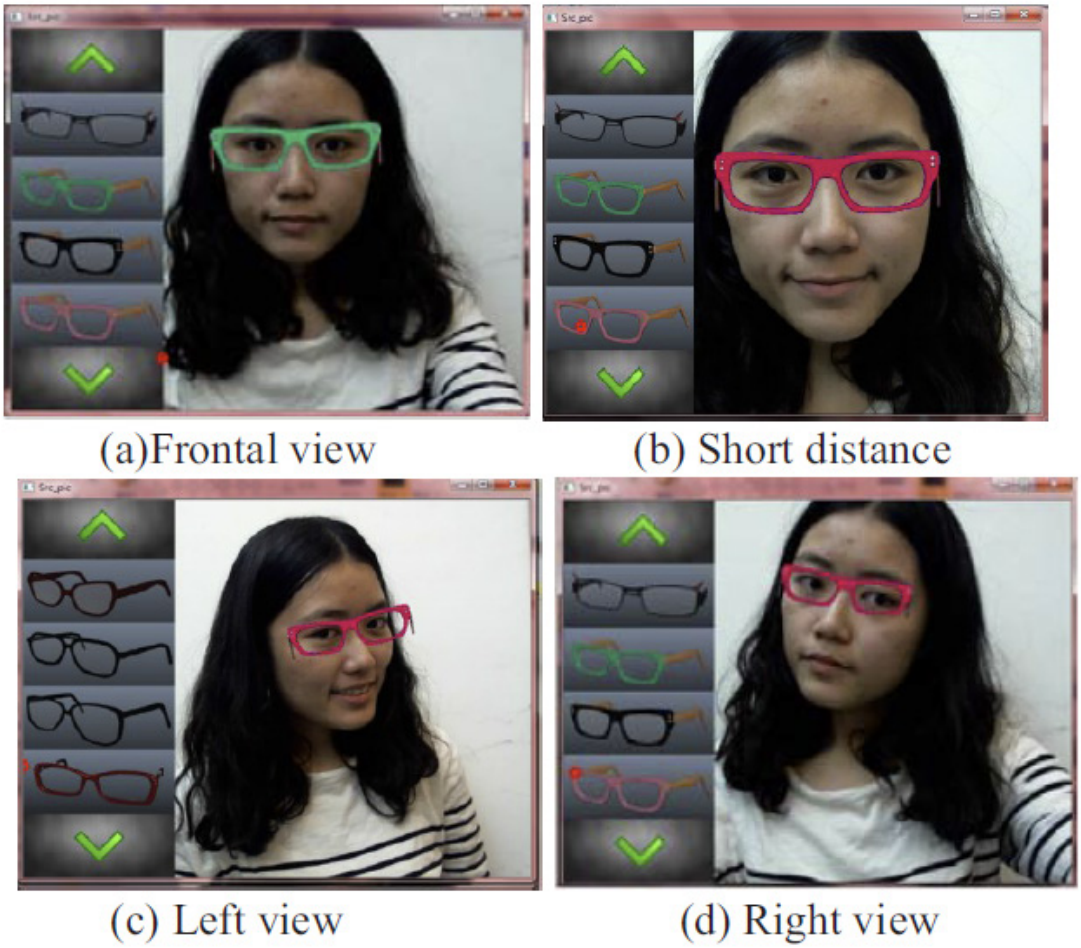

Figure 2: vision-based virtual eyeglasses fitting system results

The glasses images fit the face well, even the user moves close to or far from the camera. On the other hand, this system is not able to handle the rotation of the user's head because the glasses images are created form the frontal view.

2.2 3D Glasses Model with 2D Face Image Sequence Approaches

Vacchetti et al.[5] develop a fast and reliable 3D tracking system, and in their implementation, they showed that virtual glasses can be superimposed onto a user's face image sequence properly and stably. This method also requires manual initialization in order to obtain several pairs of 2D-3D correspondences between the reference image and a corresponding 3D model. Oscar et al.[6] develop a live video eyeglasses selection system using computer vision tracking technique. However, as mentioned in their paper, the system is unable to meet real-time performance.

The Ditto system in[7] provides virtual try-on for any user on the web. (See figure 3) It first captures and records the image sequence of the user while he/she moves head. Then the system provides the dynamic virtual try-on result for the recoded sequence.

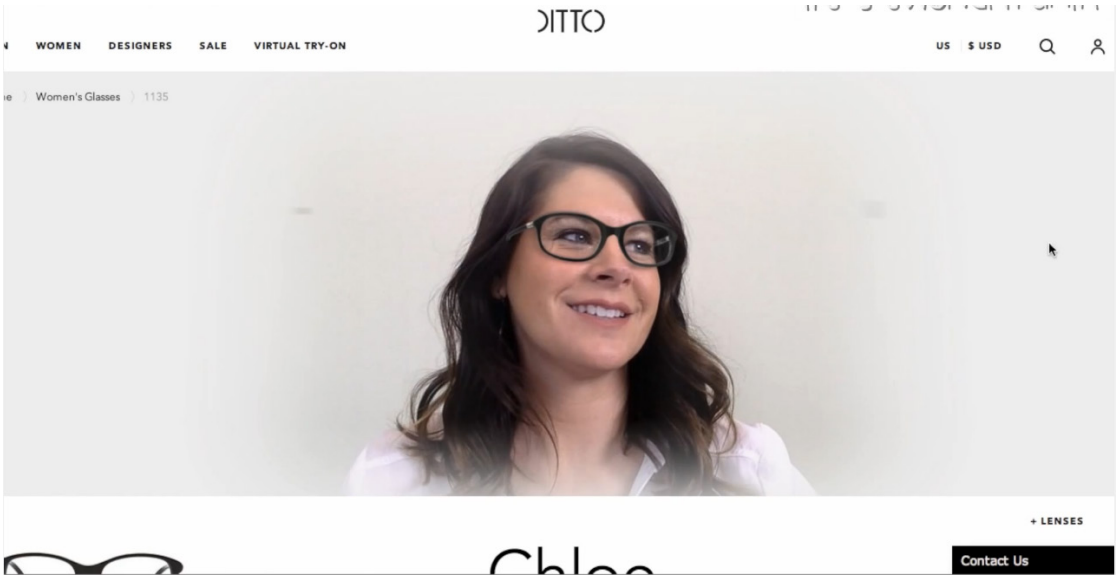

Figure 3: Ditto system

However, it is not a real-time processing system, and the head rotation is limited in only the yaw direction.

Mixed reality system in [8] adopts 3D glasses models try-on which enables the user to view different virtual models fitted on his/her face properly in the current pose. Their system does not require any manual initialization and works in real-time. On the other hand, this system requires a special mirror system for display (See figure 4), which makes the method inconvenient.

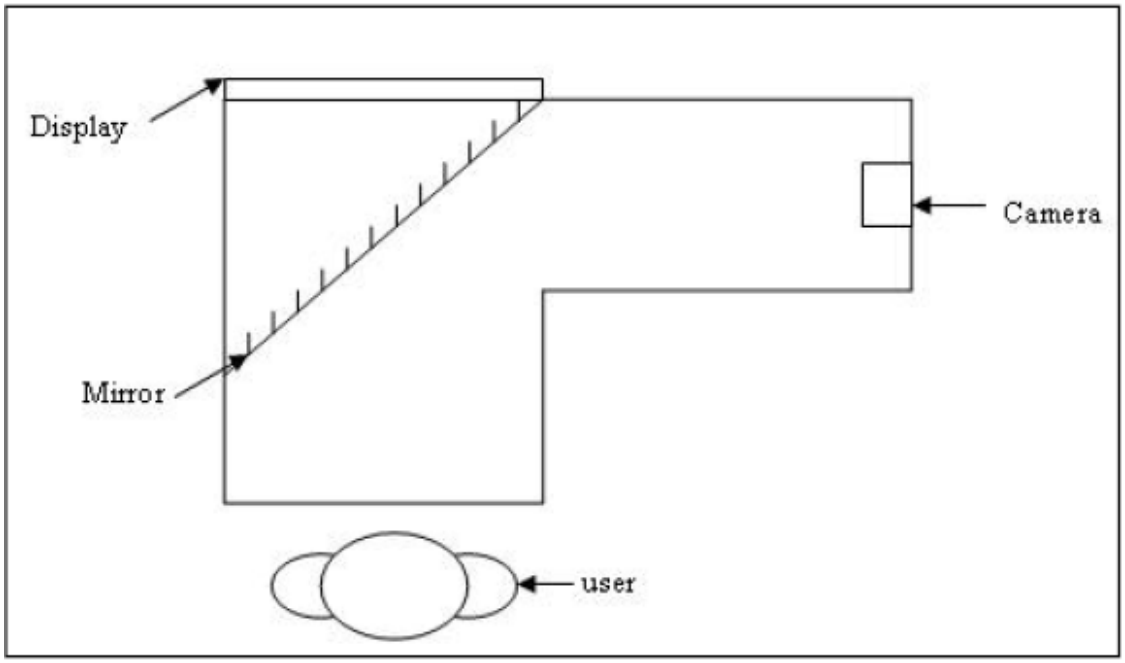

Figure 4: Mixed Reality System

This system can scale and orient to face size and orientation as far as face is within the camera view which shows good commercial potential in the eyewear market.

2.3 3D Glasses Models with Pre-reconstruct 3D Head Model

In[9] , the system relies on a 3D user head reconstruction, and then fits 3D glasses model to 3D head model and renders the final results. This system's architecture is an enlightenment of considering 3D user head model as part of try-on system.

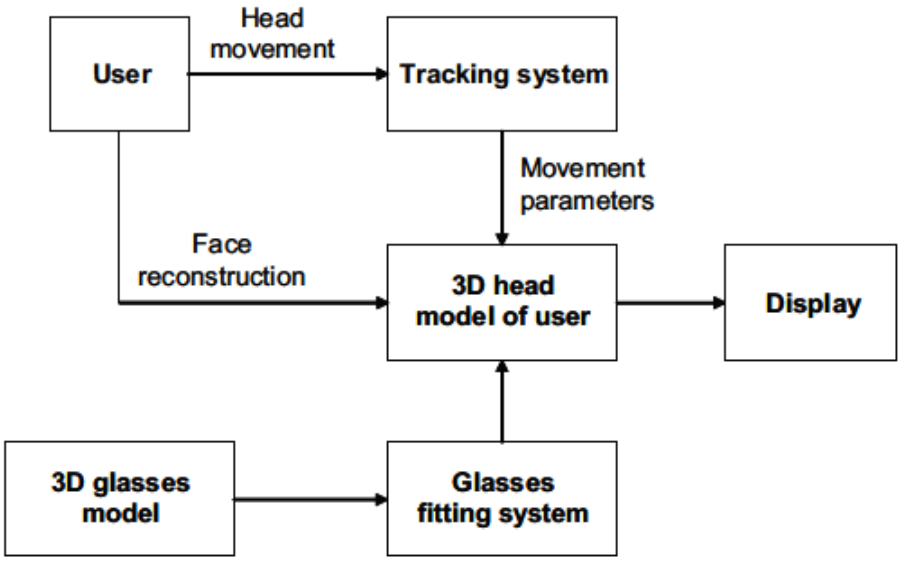

Figure 5: virtual try-on of eyeglasses using 3D model of the head pipeline

The main components of the system are the glasses fitting system and the tracking system. After the 3D head model has been reconstructed from an image of the user, the 3D glasses are fitted on this 3D head model. The fitting is performed by computing the affine transformation3 parameter that "put" the 3D glasses on the head model. The user's head movement is then tracked by the tracking system, which produces the head movement parameters to update the 3D head model.

As Kinect devices spread fast, applications of Kinect in virtual try-on domain become increasingly popular. In[10], they propose a virtual 3D Eyeglasses try-on (3DET) system, which captures the user's performance with the help of both depth stream and color stream from Kinect. Then the system renders the glasses properly on the video stream immediately after the user's selection. The virtual eye-glasses follow with the movement/rotation of the user's head simultaneously.

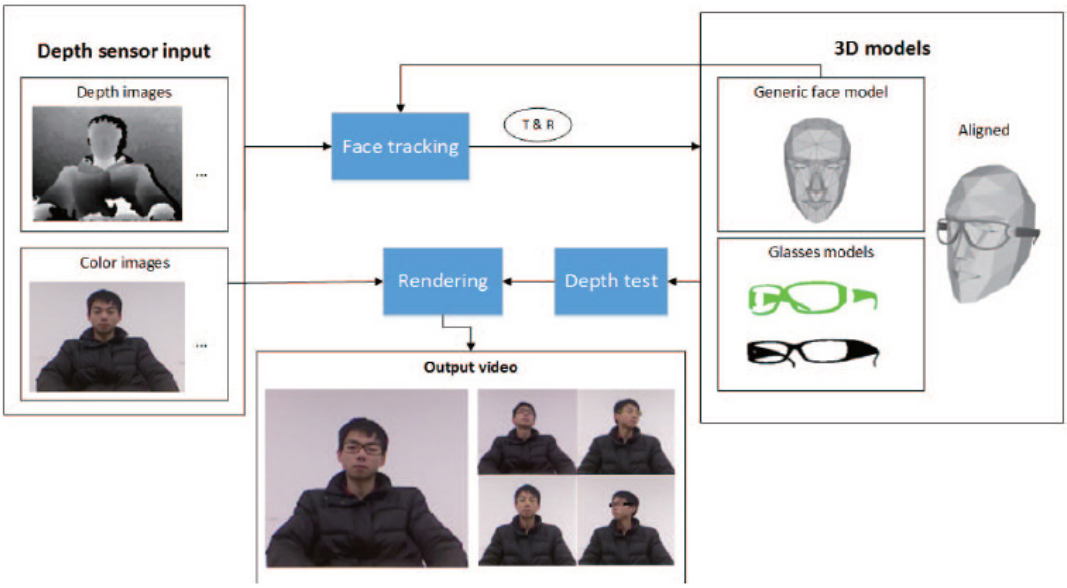

Figure 6: the pipeline of 3DET

The pipeline of 3DET is shown above. (See figure 6) The Drawback of this system is that each pair glasses is pre-aligned with generic face model, which means they are assuming that each pair of glasses can be fitted perfected on the face regardless of size of each pair. In practice, the size of glasses frames can vary widely based on different users' face shapes. In other words, the assumption regardless of frame size is impractical.

In virtual try-on domain, not only the appearance but also physical feedbacks are taken into account. Therefore, the lack of physical feedback becomes most critical weakness of those virtual try-on systems.

Chapter 3 Main Challenges

3.1 System Overview

The pipeline of our virtual glasses try-on system is illustrated here (See figure 7)

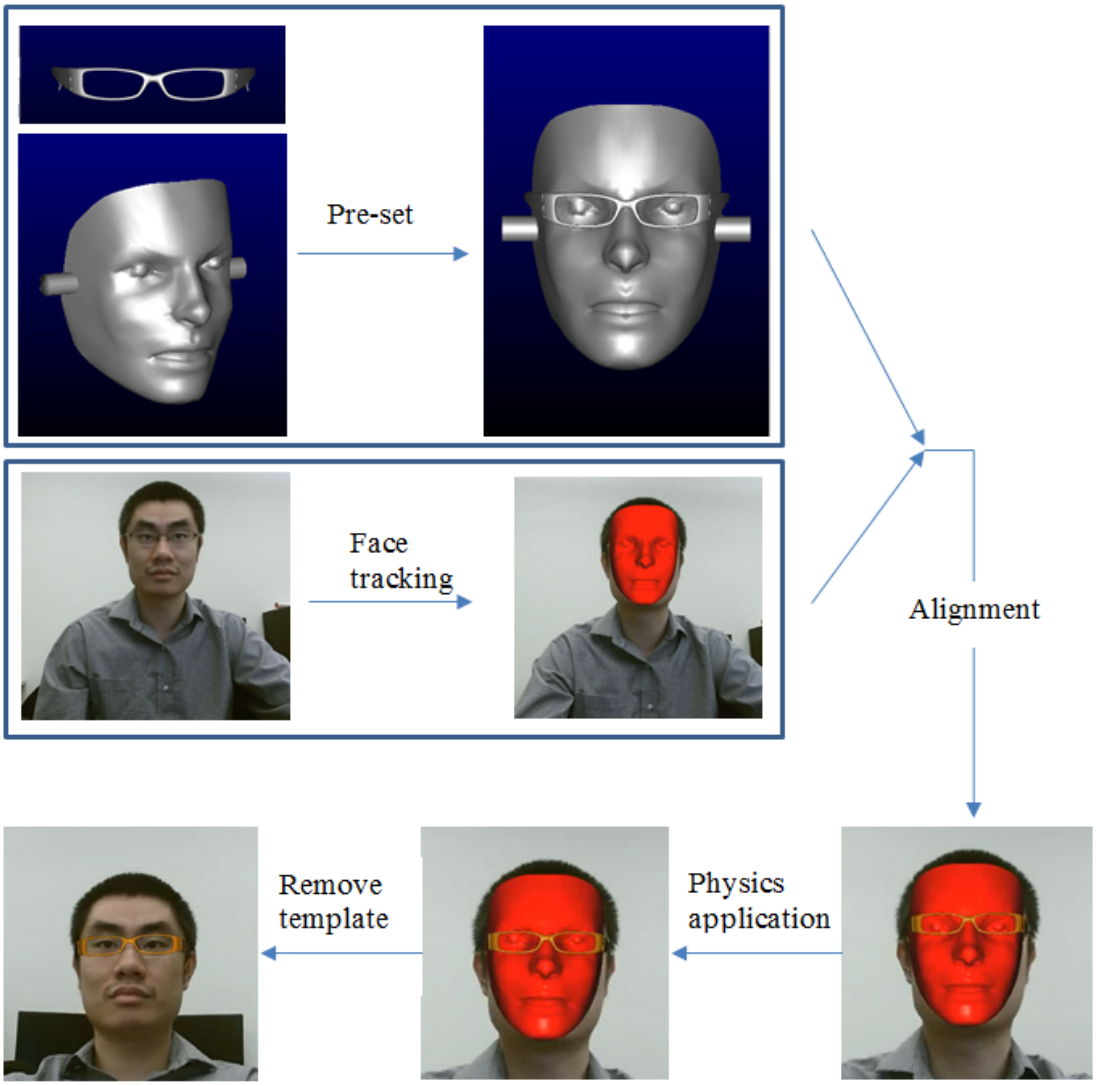

Figure 7: system pipeline

In preparation, generic face model is aligned with glasses models while pre-setting. We do this alignment manually to make sure the size of glasses fit the generic face model perfectly. When our system stars to work, the color stream capture together with face tracking is done by Kinect Sensor in real-time. Based on the input of images' sequence, the area of the user's face is detected and tracked. Then, HD face mesh of the user is reconstructed. Therefore, the movement of the user face mesh, which can be represented by transformation matrix, is going to be calculated each frame. We then find out the transformations needed to make two meshes (HD face mesh and pre-setting generic face mesh) coincide, and bring meshes and virtual glasses into one common coordinate system - Bullet physics world, to finish the alignment. At last, we apply the physics simulation and get the result.

In this work, we are concerned with reconstructing and aligning the user's face mesh to the face-mesh template that we pre-set manually in real-time. Since we develop two separate systems for both Kinect V1/V2 and web camera, we are going to introduce two different approaches to conduct the alignment between objects' coordinate spaces.

3.2 Alignment

3.2.1 Offline Stage

To represent the movement of the user's face in real world, we take advantage of HD face mesh captured and reconstructed by Kinect Sensor each frame. So we need to find out the transformations to make two meshes (HD face mesh and pre-set generic face mesh) coincide. We

first specify 80 points on both face meshes. We call them anchor points here. This is the face mesh that reconstructed by Kinect Sensor.

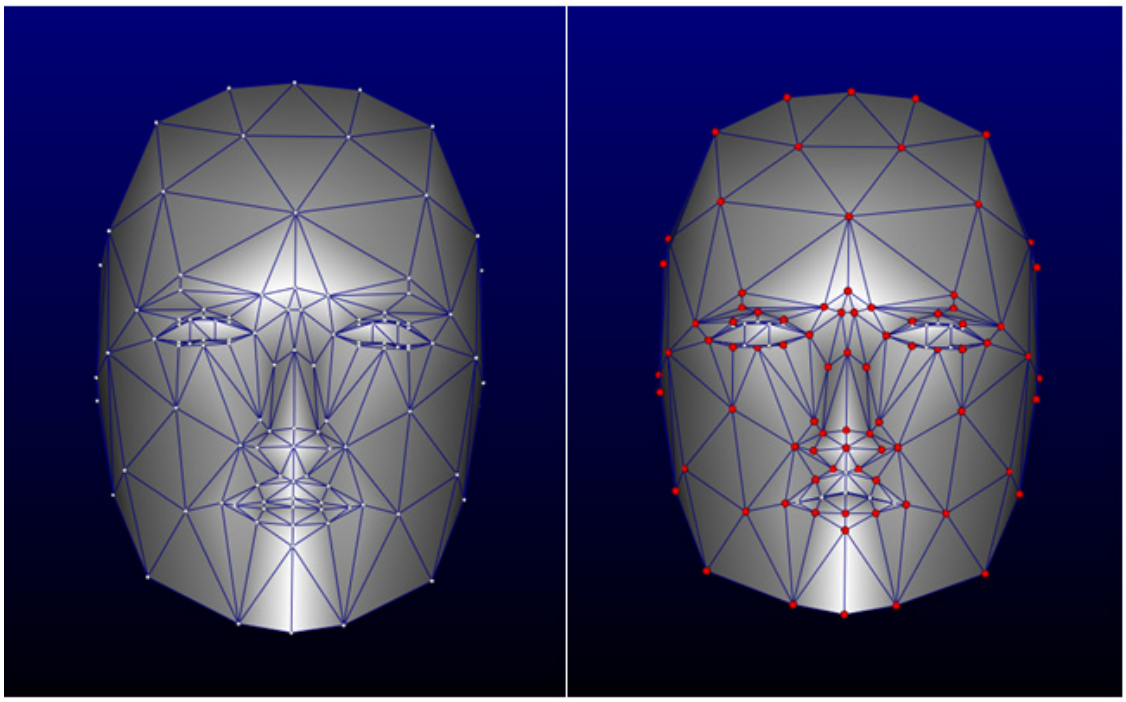

Figure 8: face mesh and picked anchor points

To be clear, this step can be done absolutely before the system starts because the indices of face mesh points maintain the same each time Kinect Sensor reconstructs the HD face mesh.

And this is the generic face mesh we use to represent the user's face.

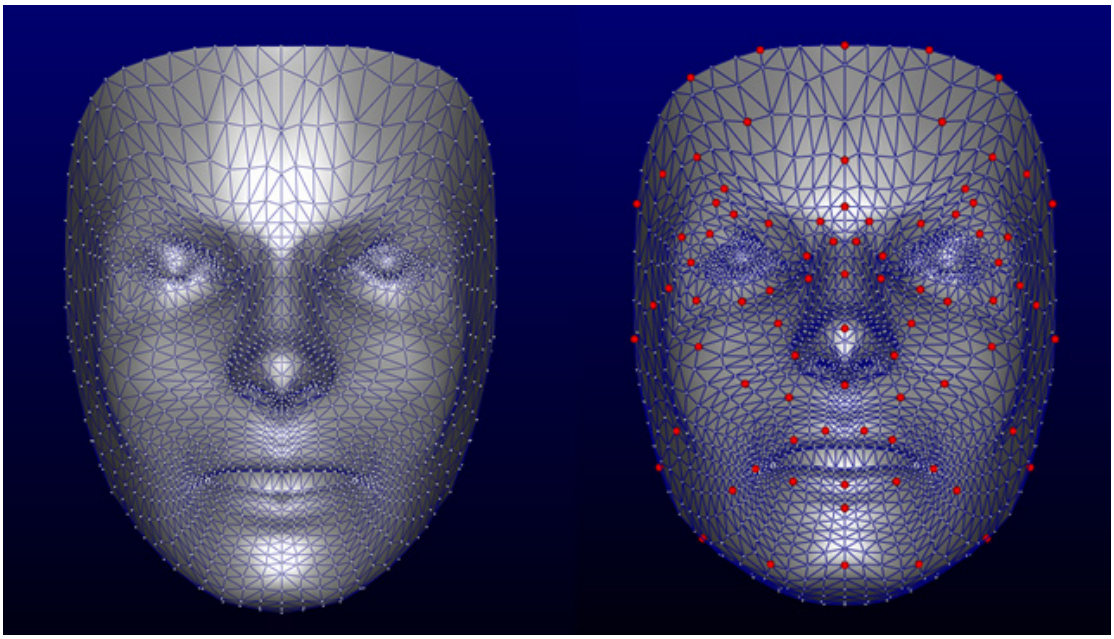

Figure 9: generic face mesh and picked anchor points

Those anchor points will be used to solve the alignment problem between HD face mesh captured and reconstructed by Kinect Sensor and pre-set generic face mesh during the online stage, which will be discussed later in this section.

3.2.2 Online Stage

3.2.2.1 Kinect Version

After specifying those anchor points on face meshes, we go to the online stage of alignment. In Kinect Version, we do not take the deformation of the user's face mesh into account. We assume that the required transformations are rigid body transformations, which means that we are going to ignore the user's different expression.

To align the HD face mesh to pre-set face mesh, we need to calculate transformation of HD face mesh (denoted as T_{HD}).

Since the HD face mesh captured and reconstructed by Kinect Sensor is measured in world space unit (meter/M), we do not need to consider of scaling it when we build the Bullet physics world, which is built in world space unit (meter/M) as well. In this case, the transformation matrix of the user's movement (denoted as T_{HD}) can be calculated by following equation:

$$T_{HD} = R * T \tag{1}$$

where R represents the rotation of HD face mesh in world space and T is the translation of HD face mesh in world space. Then the transformation of glasses in world space (denoted as T_G) is:

$$T_G = T_B * T_{HD} \tag{2}$$

where T_B is the transformation of glasses in Bullet physics world.

3.2.2.2 Webcam Version

While we ignore the user's different expression in Kinect version, we have to deal with the user's expression in Webcam version since the face tracker we use for Webcam version is able to reconstruct deformable face mesh. That is, there will be deformation of the user's face mesh when the user's face motion occurs. Thus, we apply SVD algorism to extract rigid

transformation to represent the movement of the user's face in real world. This approximate rigid transformation matrix (demoted as T_{SVD}) can be calculated by following equation:

$$P_t = T_{SVD} * P_0 \quad (3)$$

where P_t is set of anchor points on face mesh captured by face tracker at time t, P_0 is set of anchor points on generic face mesh. Again, the transformation of glasses in world space (denoted as T_G) is:

$$T_G = T_B * T_{SVD} \quad (4)$$

where T_B is the transformation of glasses in Bullet physics world.

3.3 Physics Applications among large scale meshes

To retrieve proper transformation matrix of glasses in world space (denoted as T_G) as mentioned before, we need to establish an accurate Bullet physics world. Collision detection is essential for realistic physical interactions. In order to ensure real-time interactivity with the user, 3D modeling software developers usually approximate the 3D models composing the scene (e.g. animated characters, static objects...) by a set of simple convex shapes such as ellipsoids, capsules or convex-hulls. In practice, these simple shapes provide poor approximations for concave surfaces and generate false collision detections (See figure 11). In our case, this kind of approximations is not able to meet the requirement. Both glasses mesh and face mesh have to be precise to ensure that physical simulation is accurate and proper. If we do not approximate objects in Bullet physics world as mentioned above, collision detection is not supported when the collision occurs between two static meshes due to the Bullet physics limitation. Thus, we have to

decompose large scale static mesh, which means the glasses mesh here, into a set of convex polygons before we create Bullet physics world.

3.3.1 HACD introduction

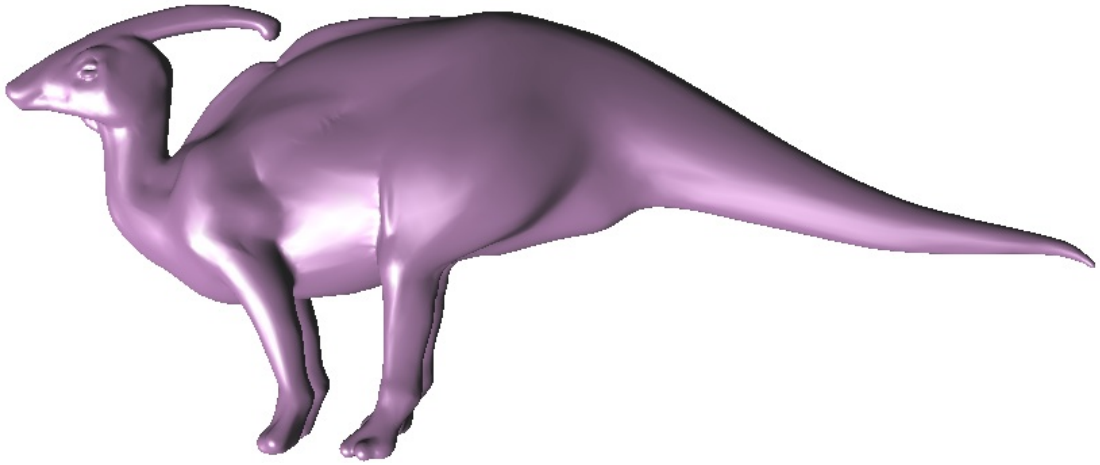

Figure 10: original mesh

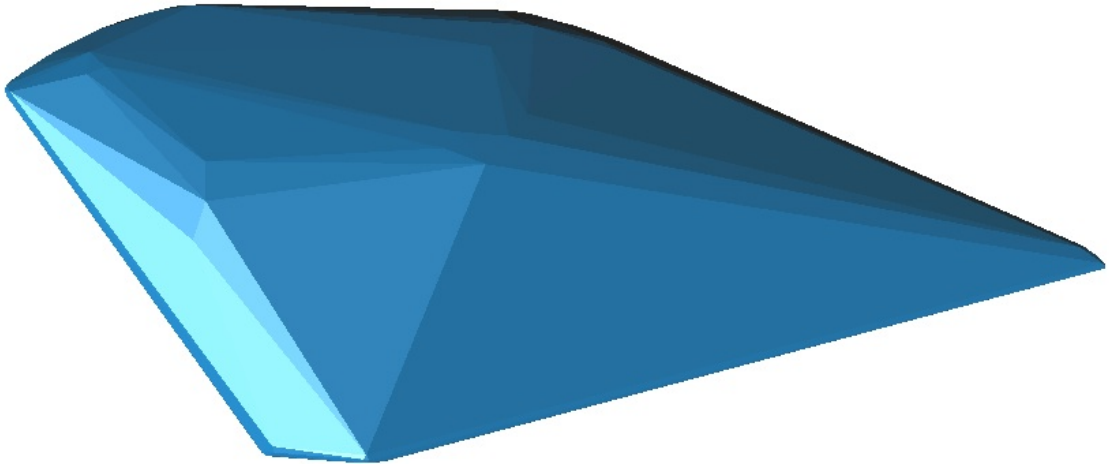

Figure 11: convex-hull

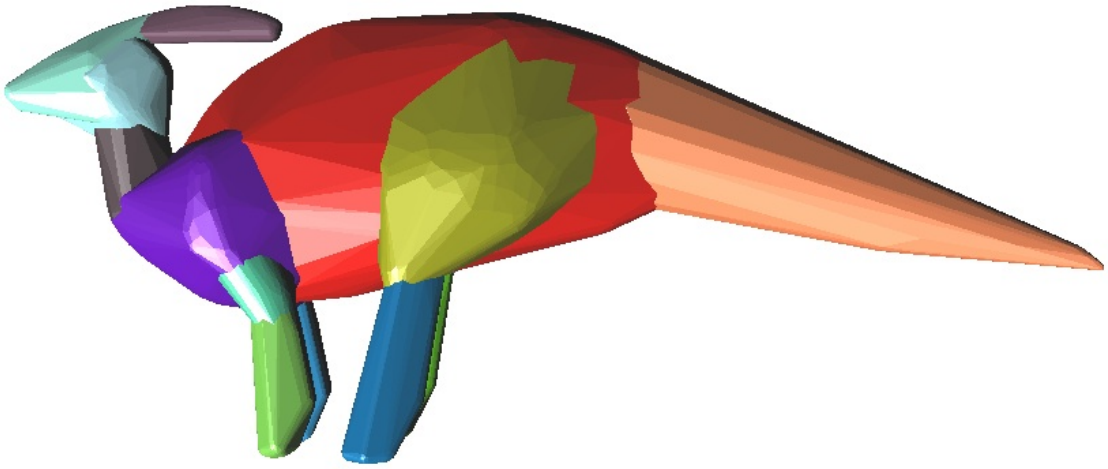

Figure 12: Approximate Convex Decomposition example

A second approach consists in computing an exact convex decomposition of a surface S[11], which consists in partitioning it into a minimal set of convex sub-surfaces. Exact convex decomposition algorithms[12] are NP-hard and non-practical since they produce a high number of clusters. These limitations lead to failure of real-time system requirement. To overcome these limitations, the exact convexity constraint is relaxed and an approximate convex decomposition[13] of S is instead computed. (See figure 12) An approximate convex decomposition of this original dinosaur mesh is computed and each part with specific color represents a convex hull, which makes real-time collision detection between certain object and this dinosaur mesh affordable and accurate.

3.3.2 Experimental Results and Conclusion

In our work, we apply HACD (Hierarchical Approximate Convex Decomposition) to our glasses meshes before the physics simulation in order to meet our real-time requirement.

These are high resolution meshes that we use to display glasses in our system. (See figure 13, figure 14)

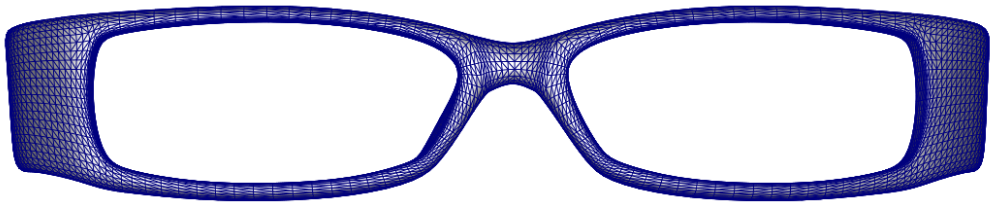

Figure 13: middle part of glasses high-resolution mesh

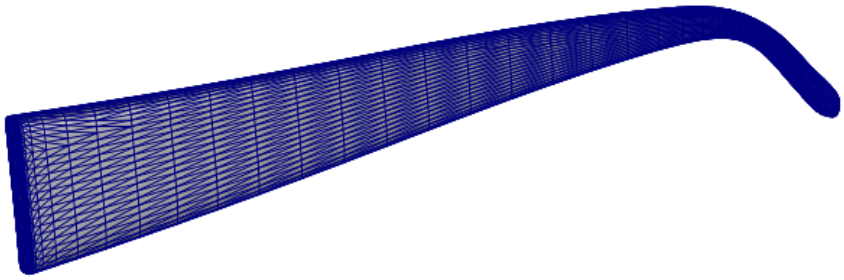

Figure 14: supporter of glasses high-resolution mesh

These are low resolution meshes we use to simulate physical collision in Bullet world. In the mean time, they are the inputs of HACD algorithm. (See figure 15, figure 16)

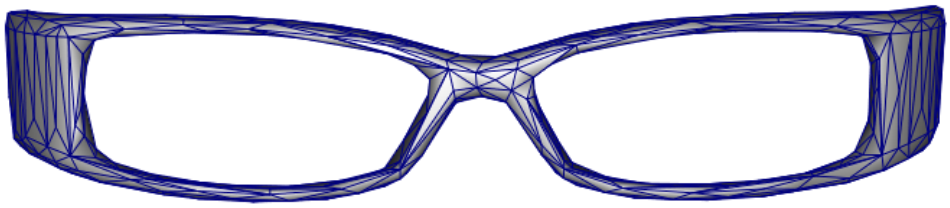

Figure 15: middle part of glasses low-resolution mesh

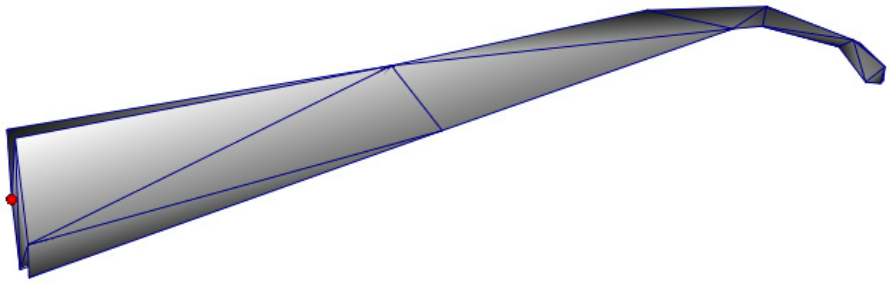

Figure 16: supporter of glasses low-resolution mesh

We have experimented with different thresholds used in HACD algorithm. We have found that this method is not alwayss successful in decompositing complex concave objects into simple convex objects set. We shall first demonstreate a result where the method is going sucessfully.

Successful results:

These are results after we apply HACD algorithm with proper threshold to glasses meshes. (see figure 16, figure 17)

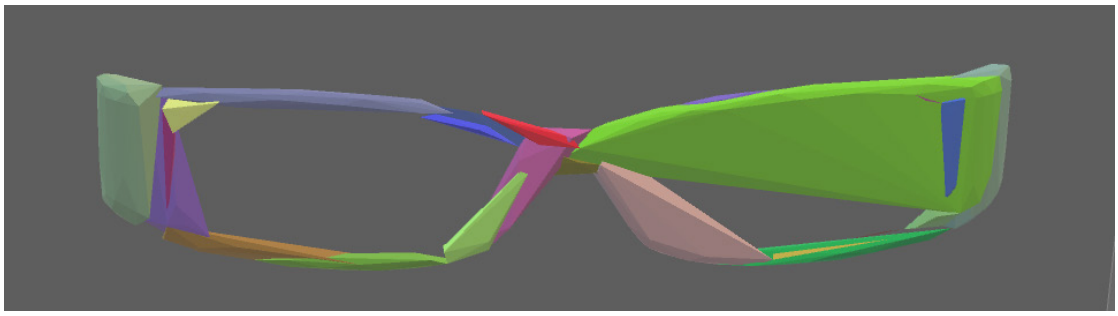

Figure 17: HACD output of middle part of glasses

This is the result after we apply HACD algorithm to the middle part of glasses frame. (See figure 17) This set objects (output) consists of 26 convex polygons, which provides an accurate silhouette for physical collision detection.

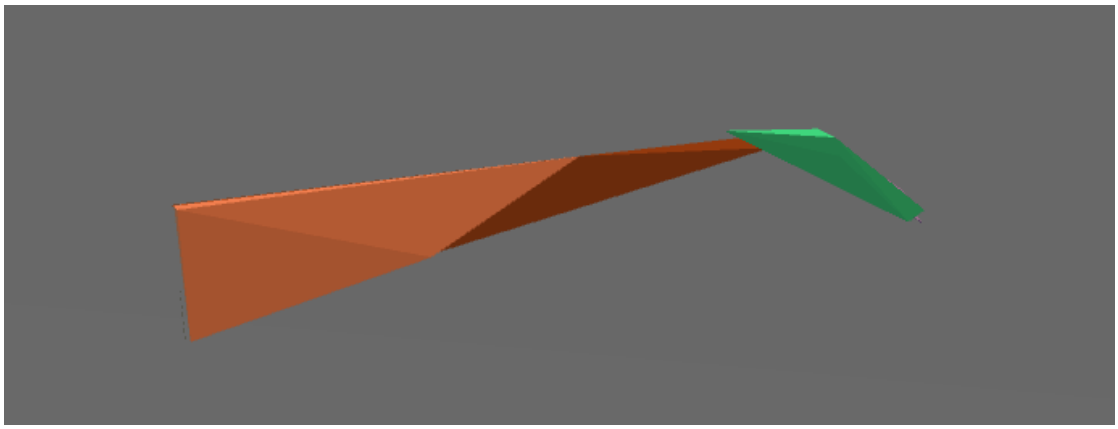

Figure 18: HACD output of supporter of glasses

This is the result after we apply HACD algorithm to one of supporter of glasses frame. (See figure18) This set of objects (output) consists of 5 convex polygons, which provides an accurate silhouette for physical collision detection as well.

And here is a failed case:

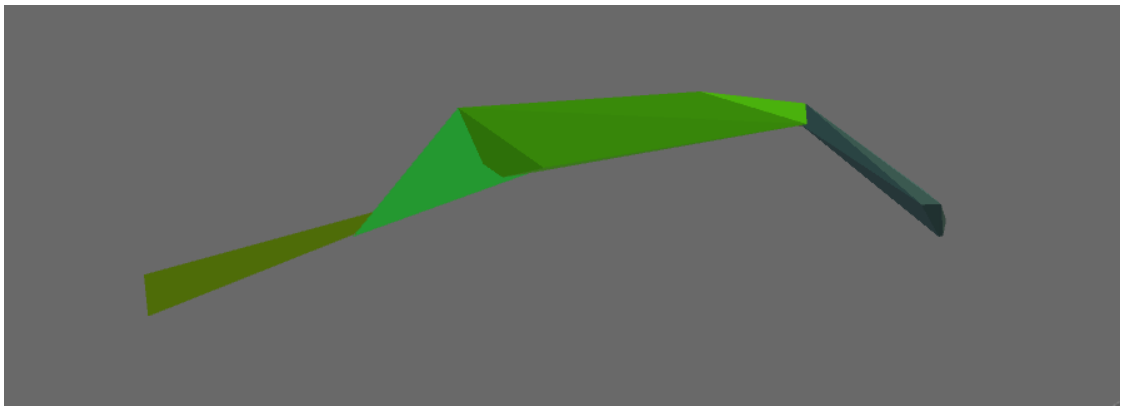

Figure 19: HACD output of supporter of glasses with improper threshold

This is a result after we apply HACD algorithm with improper threshold, which drives the final result as a set of 4 convex objects. (See figure 19) Obviously, the left end of this supporter loses too many details, which would lead to incorrect result of collision detection. Therefore, as shown above, to improve the HACD algorithm results to meet system requirements, proper threshold has to be set.

Chapter 4 Implementation

In this chapter, we shall present the implementation in details based on following steps.

4.1 Tracking User's Face tracker

4.1.1 Kinect Version

In Kinect version system, we take advantage of the Kinect for Windows SDK API[14]. For each frame, we get real-time 3D information of the user's face positions that relate to the user's facial movements. Since this information can be applied either to a generic face mesh that represents the user's face, or to a customized user face mesh that does look like the user, we are able to track the user's face movement during the operation of the system.

During the initiation step, we scan the user's face from 4 different angles: left angle, right angle, eye level and bottom angle. (See figure 20-23)

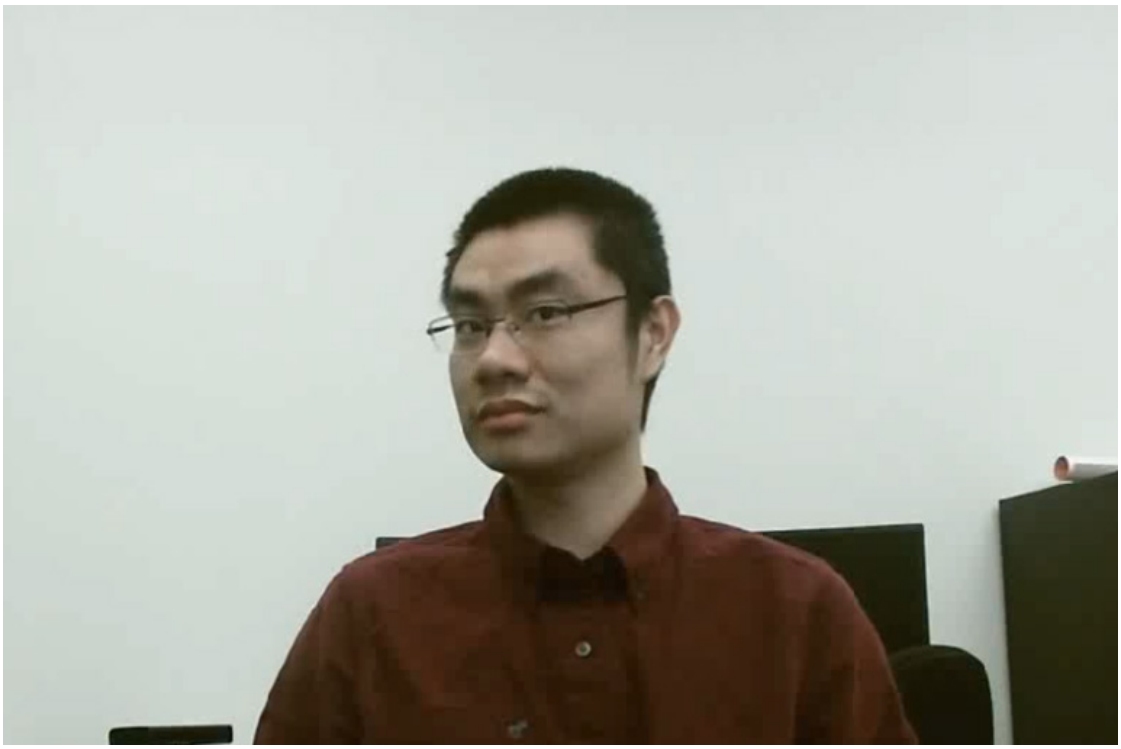

Figure 20: scanning from left angle

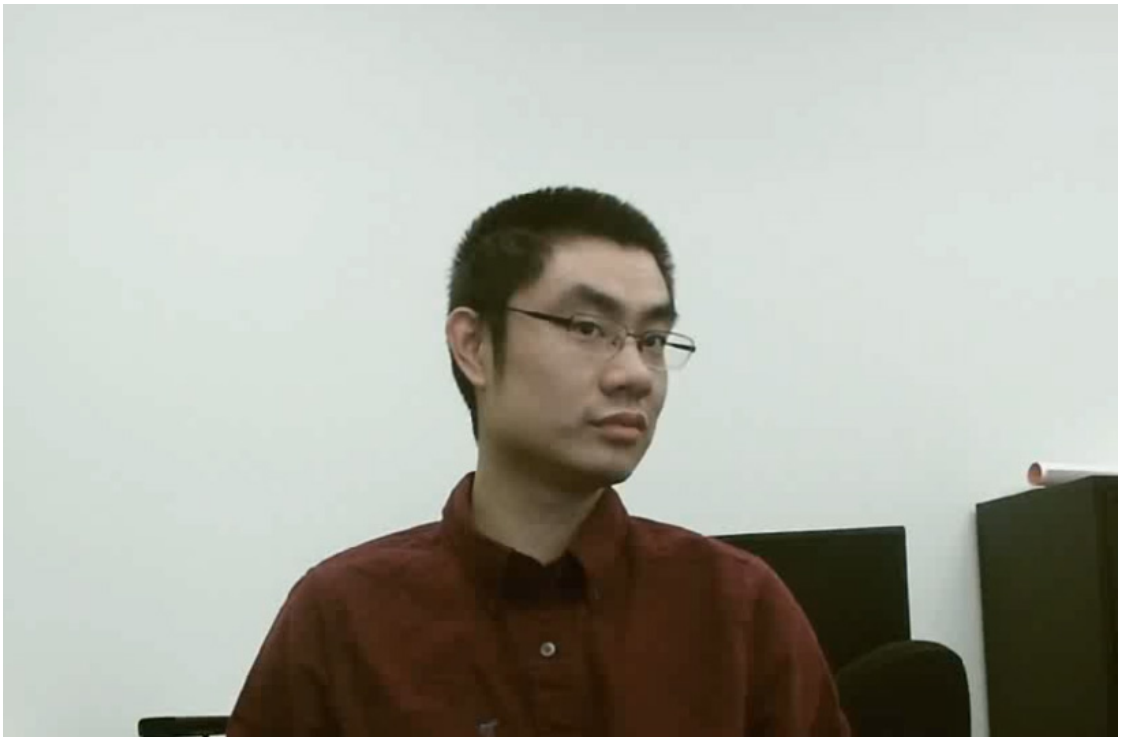

Figure 21: scanning from right angle

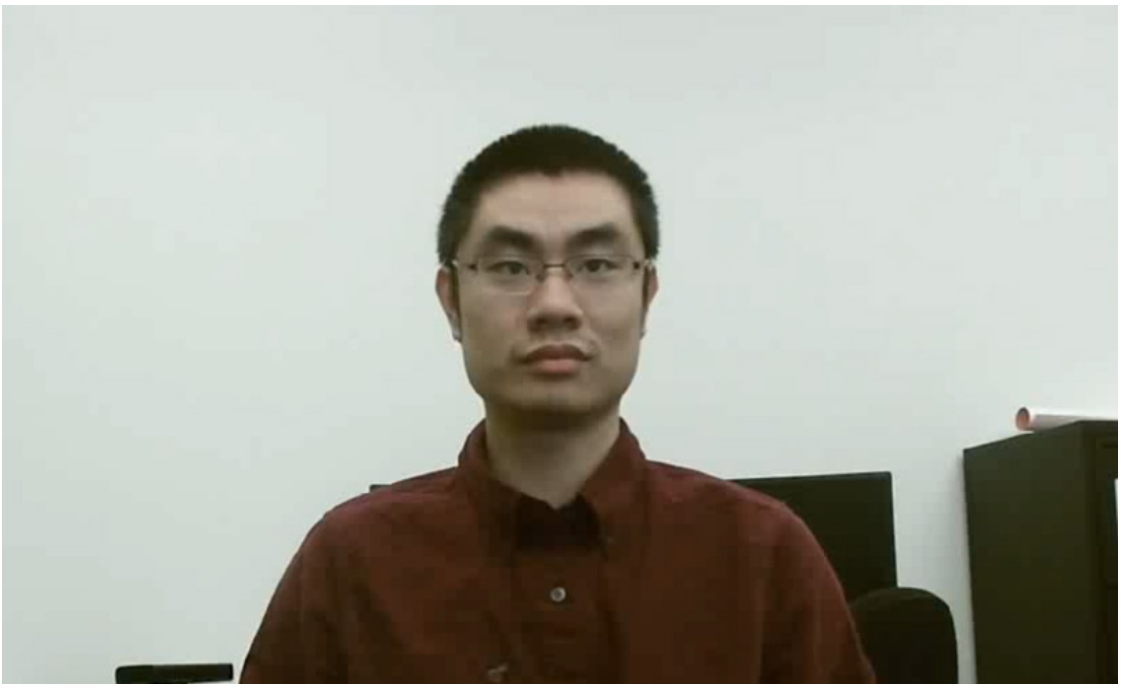

Figure 22: scanning from eye level

Figure 23: scanning from bottom angle

When it is finished, the system starts to capture the user's face and reconstruct 3D HD face mesh at every frame. While 3D HD face mesh is computed and reconstructed by each frame, we use it as input for alignment step mentioned in section 3.2.1. The API will only track those frames for which the NUI skeleton has already identified the head and neck joints, which means our system works within certain distance range.

4.1.2 Web-cam Version

In web-cam version system, as we lack the depth information retrieved from camera, we apply CLM tracker, which is a library for fitting facial models to faces in images, to find out if a valid user's face occurs in image of each frame. CLM tracker is an implementation of Saragih et al.[15], which tracks a face and outputs the coordinate positions of the face model as an array, following the numbering of the model below: (See figure 24)

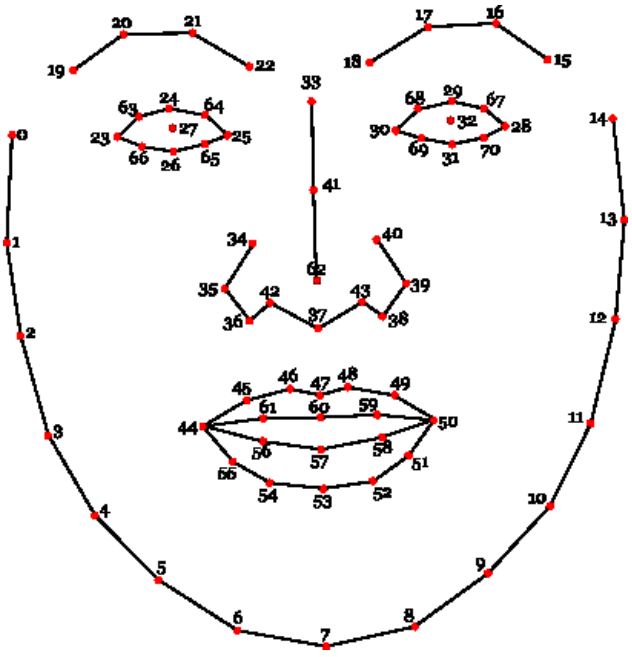

Figure 24: numbering of tracker points

As the output of CLM tracker, these points with depth information correspondence the anchor points we pick on generic face template during the pre-set progress. We use them as input for alignment step for web-cam version system.

4.2 Building Bullet World

In order to create accurate human-computer interaction, building a physics-based virtual world is necessary. Bullet physics engine provides the simulations of collision detection, rigid body dynamics. We select Bullet physics engine since it is of no charge and open-source and it fully matches our technical requirement.

We create our Bullet physics world where gravity is the only force field. In this space, both generic template face mesh and glasses mesh are hypothesized to be rigid, and the generic template face mesh is static. In Equations (2) and (4), in order to calculate the transformation of glasses in world space (denoted as T_G) , we need to acquire T_B which refers to the transformation of glasses in Bullet physics world.

We register the generic template face mesh as a static rigid body in Bullet physics world and register each part of glasses as dynamic rigid body. The middle part of glasses frame connects with left and right supporter via two pivot points, which restrict the two supporters to move along the axes. (See figure 23)

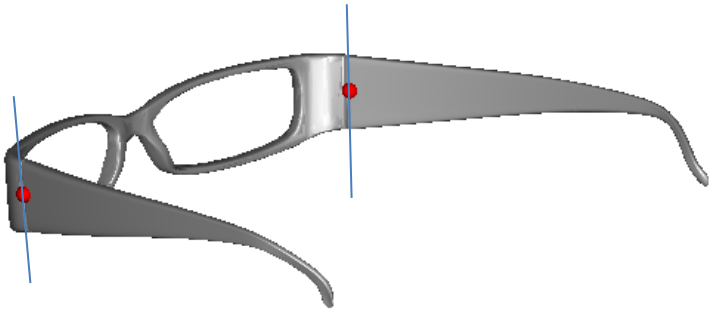

Figure 25: glasses frame with pivot points

When the registration is finished, we enable the gravity and friction in our Bullet physics world to retrieve T_B. Here is the result: (See figure 24)

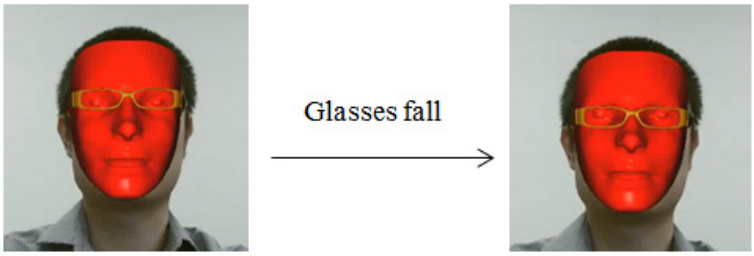

Figure 26: result after gravity and friction activated

After gravity and friction activated, the Glasses fall on to nose as we expect. That is to say our hypothesis in Bullet physics world is proper and correct.

4.3 Scanning Customized Face

Generic template used in the system makes this try-on system easy to develop and deploy because we don't take specific user's face geometry into account. On the other hand, it will lead to inaccurate physical feedbacks because the geometry of face may vary significantly among

users. Thus, we have used another application to acquire specific user's face mesh to improve our accuracy of physical simulation.

KinectFusion provides 3D object scanning and model creation using a Kinect for Windows sensor. With KinectFusion, we can create a scene with the Kinect camera and simultaneously see, and interact with, a detailed 3D model of the scene[16]. In our work, we use KinectFusion to scan the user's face from multiple view angles. After we retrieve the surface which contains the user's face geometry, we cut the face part off the scene as customized face mesh. (See Figure 27)

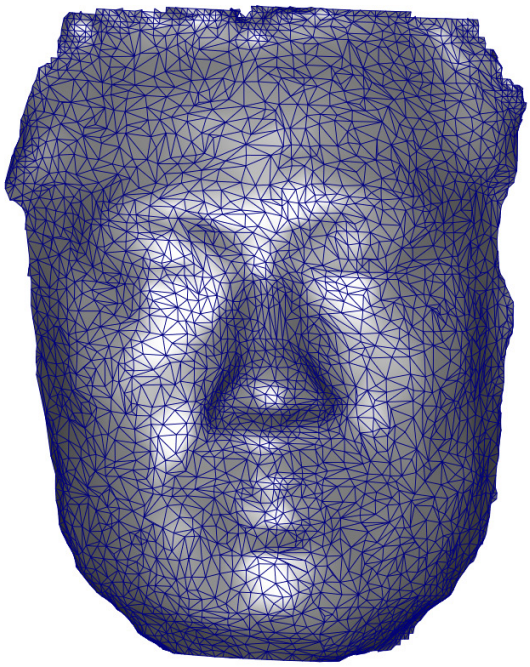

Figure 27: customized face mesh

Here is the result after we replace the generic template face mesh with customized face mesh.

Figure 28: result of customized face mesh used

Although scanning the user's face with KinectFusion provides a new method to increase the physical feedback's accuracy, it is an offline method rather than a real-time method. Thus, it is a tradeoff between the performance and operating efficiency.

Chapter 5 Results and Comparison

After removal of the face mesh, we can finally take a look at the results with different cameras at same time.

Here is Kinect version 2 system result: (See figure 29)

Figure 29: Kinect V2 result

Here is Kinect version 1 system result: (See figure 30)

Figure 30: Kinect V1 result

Here is web-cam system result: (See figure 31)

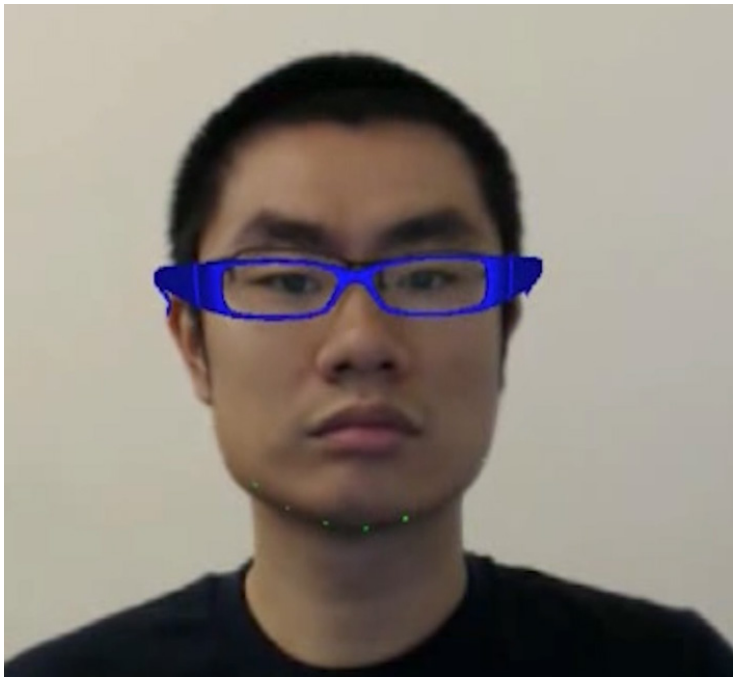

Figure 31: web-cam result

Noted that both Kinect V2 and web-cam have 1080P color stream resolution while Kinect V1 only has 480P color stream, it has been the huge disadvantage for Kinect V1 system compared to Kinect V2 and web-cam system which have much higher resolution.

On the other hand, due to technical limit, 2D face tracker used in web-cam system would fail to track the user's face when 2D input image distorts. That is, there is a limitation for web-cam users that they only have few different view angles to take a look at results. In other words, if they turn their head too much, the system will lose face tracking progress.

Compared to Kinect Verion1 and web-cam system, Kinect V2 system has several advantages:

<1> Larger view angles range, which offers users more different view angles to see if they like the glasses frame or not.

<2> less lightning condition sensitive, which makes Kinect V2 system more robust. Unlike Kinect Verion1 and web-cam system have to operate indoors with good light condition, Kinect V2 system is able to work in hall or even outdoors.

As for the result of using customized face mesh, it is shown below: (See figure 32)

Figure 32: customized face mesh used

And here is the result of using generic template face mesh: (See figure 33)

Figure 33: generic template face mesh used

It is obvious that system using customized face mesh gets better result. The virtual glasses sit tight on the user's nose and coincide with the user's real glasses. (See figure 32)

Chapter 6 Conclusion and Future Work

Rapid and accurate 3D virtual-reality human-computer interaction is highly needed for many different applications today in the entertainment world, commercial enterprises and engineering sectors. The research reported in this thesis lies primarily in this domain. High accuracy, low costs and simplicity with applying 3D sensors are becoming available gradually with the rapid evolution of technology. These sensors are to provide a convenient and fast mechanism to acquire the 3D geometry of existing surfaces/objects. In order to acquire a better 3D virtual-reality human-computer interaction, the combination of scanning the 3D geometry of an object surface and applying the physics simulation becomes essential in this modern world. The most important problems to be addressed in this research are the alignment between a user's movement and pre-set 3D face mesh and precise physics simulation among large scale meshes which have been addressed in this thesis.

Our goal was to build a new system for 3D virtual glasses try-on and we succeeded. Our system is based on Kinect sensor and Bullet physics engine. Compared to existing methods, our system is real-time and more accurate along with a precise physics simulation. Since a Kinect sensor is all the user needs to achieve our new system, it makes glasses try-on considerably more convenient. Not only that, it makes shopping online for glasses more practical and much more reliable. The advantages and disadvantages of our system are listed below.

6.1 Advantages

<1> We built this real-time virtual glasses try-on system for both Kinect sensor and common web camera users. This highly increases the system's accessibility.

<2> Physics simulation is fast and accurate. Experience of physical feedback is useful when it comes to scenario of glasses try-on.

6.2 Disadvantages

<1> However, some parts of those test glasses are occluded by the user's head and will not be seen when the user turns his or her head around.

<2> Meshes are pre-set and are not automatic. We have to pre-set each pair of glasses manually.

6.3 Future Work

There are still a number of ways to explore this kind of work even further than we did. A few of these are listed below.

- The method of handling collision, which occurs during the physical simulation, is set by default by Bullet engine. In future works, it will be even better if we can find a non-penetration collision handling method.

- During our research, we tried ICP algorithm to capture the user's face movement in order to replace the Kinect face tracker. However, in some extreme cases, this algorithm fails to recognize the exact part of the user's face in real-time video consequences. It is necessary to work out a more robust technique to track a user's face and its movement.

References

[1] https://dev.windows.com/en-us/kinect

[2] http://www.framesdirect.com

[3] Wan-Yu Huang, Chaur-Heh Hsieh, and Jeng-Sheng Yeh, "Vision-based virtual eyeglasses fitting system," IEEE International Symposium on Consumer Electronics (ISCE), pp. 45-46, 2013

[4] Hazewinkel, Michiel, ed. (2001), "Affine transformation," Encyclopedia of Mathematics, Springer.

[5] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3D tracking using online and offline Information," IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(10), 1385-1391, 2004

[6] D. Oscar, C. Modesto, L. Javier and A Lius, "Computer vision based eyewear selector," J Zhejiang University Science, 11(2), pp. 79-91, 2010

[7] http://www.ditto.com/create

[8] Miaolong Yuan, Ishtiaq Rasool Khan, farzam Farbiz, Arthur Niswar, and Zhiyong Huang, "A mixed reality system for virtual glasses try-on," Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, pp. 363-366, 2011

[9] Arthur Niswar, Ishtiaq Rasool Khan, and Farzam Farbiz, "Virtual try-on of eyeglasses using 3D model of the head," Proceedings of the 10th International

Conference on Virtual Reality Continuum and Its Applications in Industry, pp. 435-438, 2011

[10] Difei Tang, Juyong Zhang, Ketan Tang, Lingfeng Xu, Lu Fang, "Making 3D eyeglasses try-on practical," IEEE Conference on Multimedia and Expo Workshops (ICMEW), pp. 1-6, 2014

[11] J. Mark Keil, "Decomposing a polygon into simpler components," SIAM Journal on Computing, Volume 14, No. 4, pp. 799-817, 1985

[12] Peter Hachenberger, "Exact Minkowksi Sums of Polyhedra and Exact and Efficient Decomposition of Polyhedra in Convex Pieces," Algorithms – ESA 2007, pp. 669-680, 2007

[13] Jyh-Ming Lien, Nancy M. Amato, "Approximate Convex Decomposition," In Proc. ACM Symposium on Computational Geometry, pp. 457-458, 2004

[14] https://msdn.microsoft.com/en-us/library/dn785525.aspx

[15] Jason M. Saragih, Simon Lucey, Jeffrey F. Cohn, "Deformable model fitting by regularized landmark mean-shift," International Journal of Computer Vision, Volume 91 Issue 2, pp. 200-215, 2011

[16] https://msdn.microsoft.com/en-us/library/dn188670.aspx