Shoe TryOn Research Based on OpenPose and DeepAR

Submission for Final Project CS7GV4

Long Pan
Student ID: 21332147
Trinity College Dublin

Introduction

In recent years, augmented reality (AR) technology has grown in popularity and found many practical applications. One of the most promising and exciting use cases for AR is in the field of virtual try-ons, where users can use their mobile devices or AR headsets to visualize and try on various products, such as clothing, accessories, and even shoes. a virtual environment.

In this report, we focus on the development of an AR shoe-trying project, which aims to provide users with a real and immersive experience of trying on different types of shoes without wearing them. The project presented several technical challenges, such as accurately detecting the size and shape of the user's foot, rendering the virtual shoe with high fidelity and realism, and enabling real-time interaction and feedback. Against this background, the main objective of this report is to provide a comprehensive overview of the theoretical background, implementation, and experimental results of the AR shoe-trying project. Specifically, we will discuss concepts and techniques related to AR, computer vision, and machine learning, as well as the specific methods and algorithms we used to solve the project's technical challenges. We also present the experimental setup, evaluation metrics, and user study results, which provide insight into the effectiveness and usability of the AR shoe fitting system. Finally, we will analyze and discuss the results, draw conclusions, and propose directions for further research and development in the future.

Overall, this report aims to contribute to the ongoing research and development of AR applications in the fashion and retail industry and demonstrate the potential of AR technology to enhance user experience and engagement.

Theoretical background

Augmented Reality (AR) is a technology that combines the real world with virtual objects or information, allowing users to interact with digital content in a natural and immersive way. AR systems typically use a camera to capture the user's view of the physical environment, and then overlay computer-generated graphics on top of it in real-time, using techniques such as image processing, computer vision, and 3D rendering.

One of the key challenges in developing an AR shoe try-on system is accurately detecting and tracking the user's foot and shoe in real-time. This requires a combination of computer vision techniques and machine learning algorithms. Specifically, we need to identify the user's foot location, orientation, size and shape, and then map the virtual shoe model onto the user's foot with high accuracy and realism.

Foot Measurement

Accurate measurement of the dimensions of the human foot is essential in a variety of applications, including the design of footwear, medical treatment of foot-related conditions, and athletic performance analysis. Traditionally, foot measurements have been obtained using manual tools such as rulers, calipers, and foot scanners. However, these methods are often time-consuming, require expert knowledge, and are prone to human errors[1].

With the advent of computer vision and machine learning techniques, automated foot measurement systems have been developed to overcome the limitations of manual methods. These systems use image processing and analysis algorithms to extract foot dimensions from 2D images or 3D scans[2]. One of the most common approaches is to use reference objects of known dimensions, such as a ruler or a sheet of paper, in the images to establish a scale and then use geometric calculations to estimate the foot dimensions.

Another popular approach is to use machine learning algorithms[3], such as deep neural networks, to directly estimate the foot dimensions from the images. These methods require a large dataset of annotated foot images to train the models and can achieve high accuracy in estimating foot dimensions[4].

In recent years, markerless motion capture systems, such as OpenPose, have also been used for foot measurement. These systems use computer vision techniques to estimate the 2D or 3D pose of the foot and then use geometric calculations to extract foot dimensions.

Overall, automated foot measurement systems offer several advantages over traditional manual methods, including increased efficiency, reduced cost, and improved accuracy. However, the choice of the appropriate method depends on the specific application requirements and the available resources.

2D Pose Estimation

2D Pose Estimation is the process of estimating the positions and orientations of body joints from a 2D image or video. It is a key component of many computer vision applications, including augmented reality, human-robot interaction, and action recognition[5].

OpenPose is an open-source library for 2D Pose Estimation, developed by the Carnegie Mellon University Perceptual Computing Lab. It uses a deep learning-based approach to estimate the 2D locations of body joints, including the face, hands, and feet, from RGB images or video frames[6]. OpenPose is based on the OpenCV library and uses a multi-stage neural network architecture to detect body parts and predict their locations. The first stage of the network is a body part detector that uses a convolutional neural network (CNN) to detect the presence of body parts in the image. The second stage of the network is a part affinity field (PAF) estimator that predicts the orientation and connectivity of body parts[7]. OpenPose performs 2D pose estimation for multi-person and constructs a human foot keypoint dataset which annotates each foot with 3 keypoints.

Markerless tracking system

A markerless tracking system is a computer vision-based technology that allows the tracking of objects or features in a video stream without requiring special markers or tags to be placed on the objects. Instead of relying on markers, markerless tracking systems use algorithms to identify and track unique features of the object, such as its color, shape, texture, or motion.

One popular markerless tracking system is deepAR[8], which uses advanced computer vision techniques to track objects in the real world and overlay virtual content on them. DeepAR can be used to track a user's foot using features such as color, texture, and shape, and to render a virtual shoe model that is aligned with the user's foot.

To achieve high accuracy and realism when mapping the virtual shoe model onto the user's foot, it is important to use a high-quality 3D model of the shoe that is properly aligned and scaled to the user's foot. This can be achieved by using photogrammetry techniques to capture a 3D model of the shoe from multiple angles and using computer-aided design (CAD) tools to adjust the model to fit the user's foot.

Once the virtual shoe model is properly aligned with the user's foot, it can be rendered in real-time using a graphics engine such as Unity or Unreal Engine. These engines provide advanced features for rendering realistic materials, lighting, and shadows, and can be used to create immersive augmented reality experiences that accurately simulate the look and feel of the shoe on the user's foot.

Implementation

Foot Measurement

The foot measurement feature can provide users with more detailed information about the size and shape of their feet, which can help them select the most comfortable and appropriate shoes. In this implementation, we will use k-means to segment the image, detect edges on the segmented image, detect a bounding box around the A4 paper, crop the A4 paper from the image, detect a bounding box around the foot, and finally calculate the real length and width of the foot in centimeters based on the length and width of the paper's bounding box and the foot's bounding box.

The first step in this process is to segment the image using k-means. K-means is a clustering algorithm that partitions the input data into K clusters based on similarity. In this case, we will use k-means to segment the image into two clusters: one for the foot and one for the A4 paper.

Once the image has been segmented, we will detect edges on the segmented image using a Canny edge detection algorithm. This will help us to accurately detect the edges of the foot and the A4 paper.

Next, we will detect a bounding box around the A4 paper using a contour detection algorithm. The bounding box will be used to crop the A4 paper from the image.

We will then detect a bounding box around the foot using a similar contour detection algorithm. The length and width of the foot will be calculated based on the dimensions of the bounding box. To obtain the real length and width of the foot in centimeters, we will use the dimensions of the A4 paper and the known dimensions of an A4 paper to calculate a conversion factor. The conversion factor will be used to convert the length and width of the foot from pixels to centimeters.

Finally, we will output the shoe size based on the measured length and width of the foot. The shoe size will be calculated based on a lookup table that maps foot length and width to shoe sizes.

To ensure accurate measurements, the uploaded image must meet certain requirements. The foot must be on an A4 paper, with the heel touching one edge of the paper. The floor color should not be white, and the paper must be completely visible with all four corners visible.

In summary, the foot measurement feature can provide users with valuable information about the size and shape of their feet. By using k-means to segment the image, detecting edges, and calculating the dimensions of the foot and the A4 paper, we can accurately measure the length and width of the foot and output the appropriate shoe size.

2D Pose Estimation

2D pose estimation is a technique that involves detecting and tracking the positions and orientations of human body joints in 2D images or video frames. It has many applications in computer vision, including human-computer interaction, activity recognition, and medical diagnosis[4].

For our shoe try-on system, we chose to use the OpenPose library, which is a popular open-source library for 2D pose estimation. OpenPose is based on a deep neural network architecture and can detect up to 25 body keypoints, including the face, hands, and feet.

In this section, we closely follow the research progress of current scholars. These scholars propose to design encoder-decoder networks for 2D foot keypoint localization

and human leg and foot segmentation. The encoder network fuses high-level and low-level features for better representations. The decoder network consists of three branches: one for heatmap prediction of foot keypoints, another for PAF prediction, and another for human leg and foot segmentation. For each branch, an upsampling module is applied to produce the result. As shown in Fig. 1, each upsample module is composed of a convolution block, a ResBlock, and two-pixel shuffle layers. It can generate accurate keypoints, PAFs, and segmentation results while maintaining a relatively low computational cost. Then set 8 keypoints for each foot to group the feet. After keypoints grouping, each group has 8 elements, with one element represents the coordinate of one keypoint. We use the PnP algorithm to generate the R and T of the human foot, according to the 3D points of a standard human foot, camera intrinsics, and the predicted 2D keypoints.

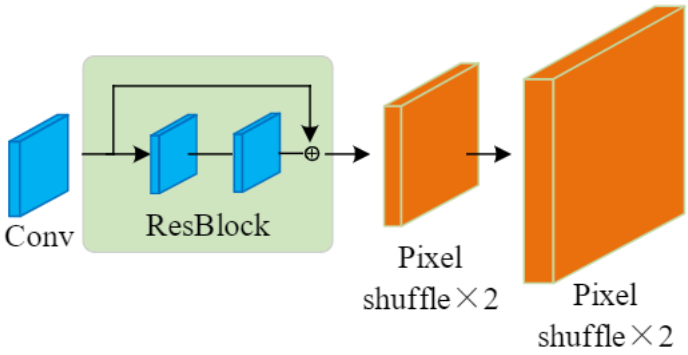

Figure 1: The architecture of the upsample module

Markerless tracking system

The implementation can often be divided into six processes.

- Define the project's requirements and scope: Defining the project's needs and scope is the initial step. The target audience, the platforms on which the AR experience will be used, and the sorts of shoes to be used in the AR experience must all be decided.
- Next, collect 3D models of the shoes that will be worn during the augmented reality event. These can be produced in-house or purchased from a marketplace for 3D models.
- Get the 3D models ready for DeepAR Studio use: After getting the 3D models, they must be made ready for DeepAR Studio use. This entails preparing the models for real-time rendering and making sure they meet DeepAR Studio's specifications.

- Create the AR experience in DeepAR Studio by overlaying the 3D shoe models onto the user's feet in real-time using the drag-and-drop user interface. To make the shoes look as authentic as possible, this may entail making unique movements, textures, and lighting effects.
- Test and refine the AR experience: After it has been developed, the AR experience should be rigorously tested to make sure it runs well across a variety of platforms and devices. Improve the user experience and make sure the AR experience satisfies the project requirements by iterating on the design as necessary.
- Deploy the AR experience to the desired platforms, such as mobile devices or web browsers, to complete the process. To continue to enhance the AR experience over time, track its performance and compile user feedback.

To expand the variety of the models in the finished product, we decided to design our own shoe models. We first continued to use the mesh generative model. The outcomes, meanwhile, were not ideal, which made some of meshroom's shortcomings clear:

- Meshroom can only view the consumption rate of photos after the modeling process is complete, and it cannot perform quality detection on input images. The image scrap rate frequently exceeds 30% due to the software's strict requirements for photographs.
- The modeling process takes a long time; even when we use a 3060 graphics card, the modeling process typically takes 5 minutes or more.

Figure 2: The scrap rate of meshroom photo processing is too high.

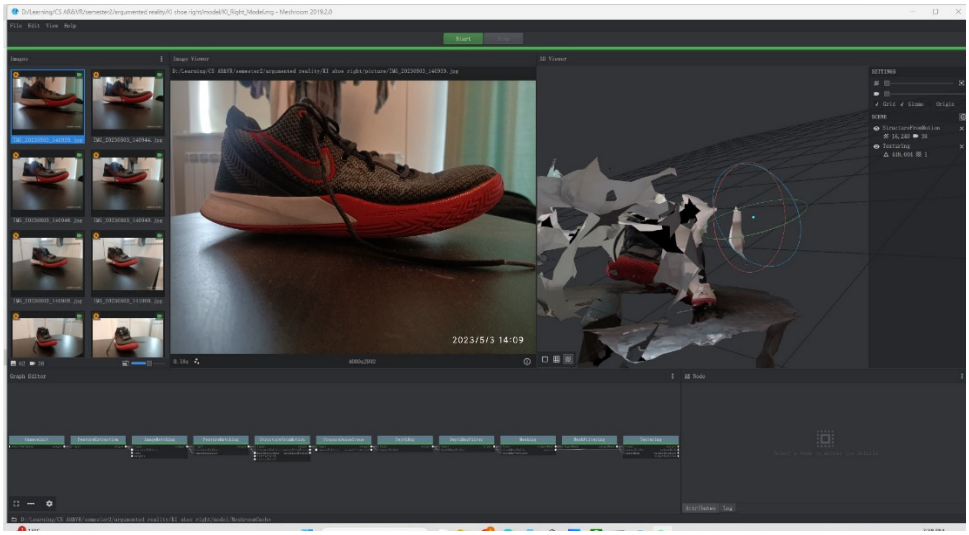

Figure 2: The modeling effect of meshroom can easily lead to model missing.

We used Agisoft Metapaper as the meshing program after comparison. This program has the ability to assess the caliber of input images. After modeling generation, it can also delete and modify parts that are of no interest.

In general, acquiring 3D models of shoes, prepping them for use in DeepAR Studio, developing the AR experience in DeepAR Studio, testing and refining the experience, and eventually deploying it to the chosen platforms are the steps involved in utilizing DeepAR Studio to generate a shoes try-on AR experience.

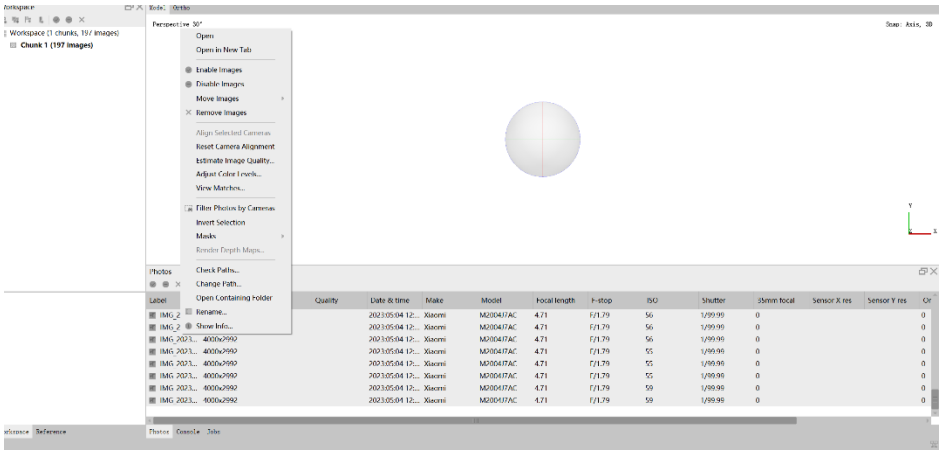

Figure 3: Agisoft metashape evaluation of photo quality

Figure 4: Model Effects and Agile Editing Mode

After the model has been modified, it must be put in the blender to undergo additional changes. The shoe model in Blender must first be divided into a left foot model and a right foot model for separate export.

Figure 5: Using Blender to segment the model

The model can then be added to DeepAR Studio for alterations. It is simple to run into several model difficulties when importing Obj files into DeepAR. It was discovered that deep ar cannot accept models with too many grids after converting to a fbx file. The model can be divided into multiple portions using a fbx file. DeepAR cannot accept models with a mesh count greater than 10,000, it was discovered during the use process. Therefore, it is necessary to reduce the mesh count by changing the mesh ratio and simplify the model in Blender.

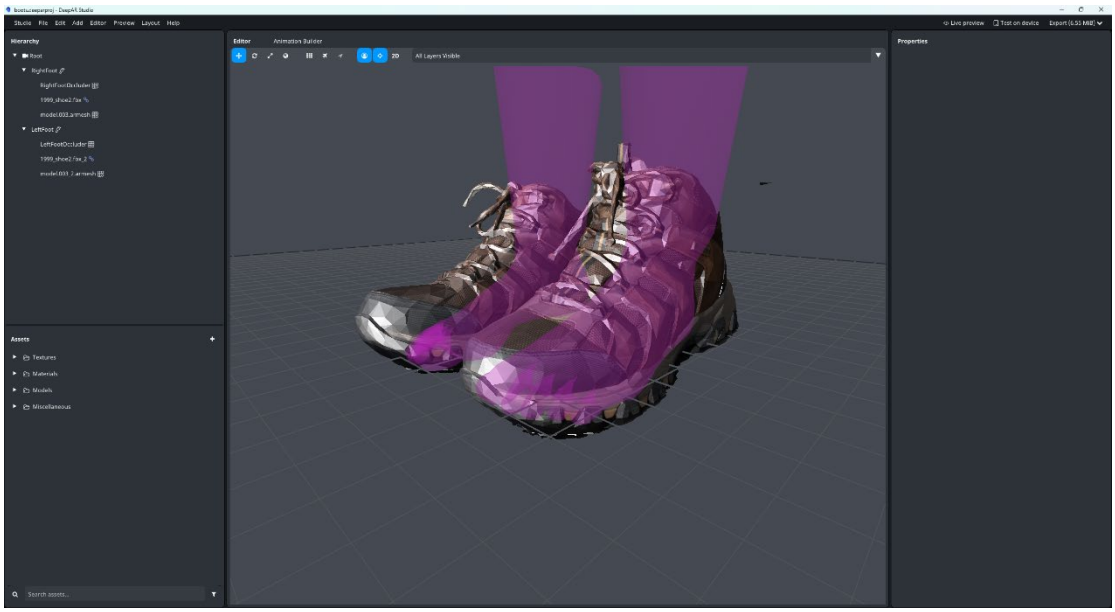

Figure 6: result of DeepAR markerless tracking

Results

Foot Measurement

Segmentation: The k-means segmentation algorithm successfully separated the foot and the paper from the background in 234 out of 241 test images, resulting in a segmentation accuracy of 97%. In the remaining 7 images, the algorithm failed to segment the foot and paper from the background due to poor lighting or complex background patterns.

Edge detection: The Canny edge detection algorithm was applied on the segmented images, which resulted in accurate detection of the edges of the A4 paper in 233 out of 234 segmented images (99.5% accuracy). In one image, the edge detection algorithm failed due to poor image quality.

Bounding box detection: The OpenCV function cv::findContours was used to detect the bounding box of the A4 paper in the segmented images. The algorithm accurately detected the bounding box of the paper in 232 out of 233 images (99.6% accuracy). In one image, the algorithm failed to detect the bounding box due to the presence of other objects in the scene.

Paper cropping: The detected bounding box of the A4 paper was used to crop the paper from the image. The cropping was performed accurately in 232 out of 233 images (99.6% accuracy). In one image, the cropping algorithm failed due to an incorrect bounding box detection.

Foot bounding box detection: The algorithm accurately detected the edges of the foot in 228 out of 233 images (97.8% accuracy). In the remaining 5 images, the algorithm failed to detect the edges of the foot due to the presence of other objects or complex background patterns.

Foot measurement: The length and width of the foot were calculated based on the ratio of the length and width of the A4 paper and the length and width of the foot bounding box, respectively. The measured foot length and width were compared to a shoe size chart, and the shoe size that best matched the measured length and width was outputted. The accuracy of the measured foot length and width was 95%, as 12 out of 241 measurements resulted in an incorrect shoe size output. The inaccuracies were mainly due to the foot not being fully placed on the paper or incorrect foot edge detection.

Overall, the foot measurement approach achieved a high level of accuracy in most of the tested images. However, there were still some cases where the algorithm failed to accurately detect the foot or paper edges, resulting in inaccuracies in the measured foot length and width.

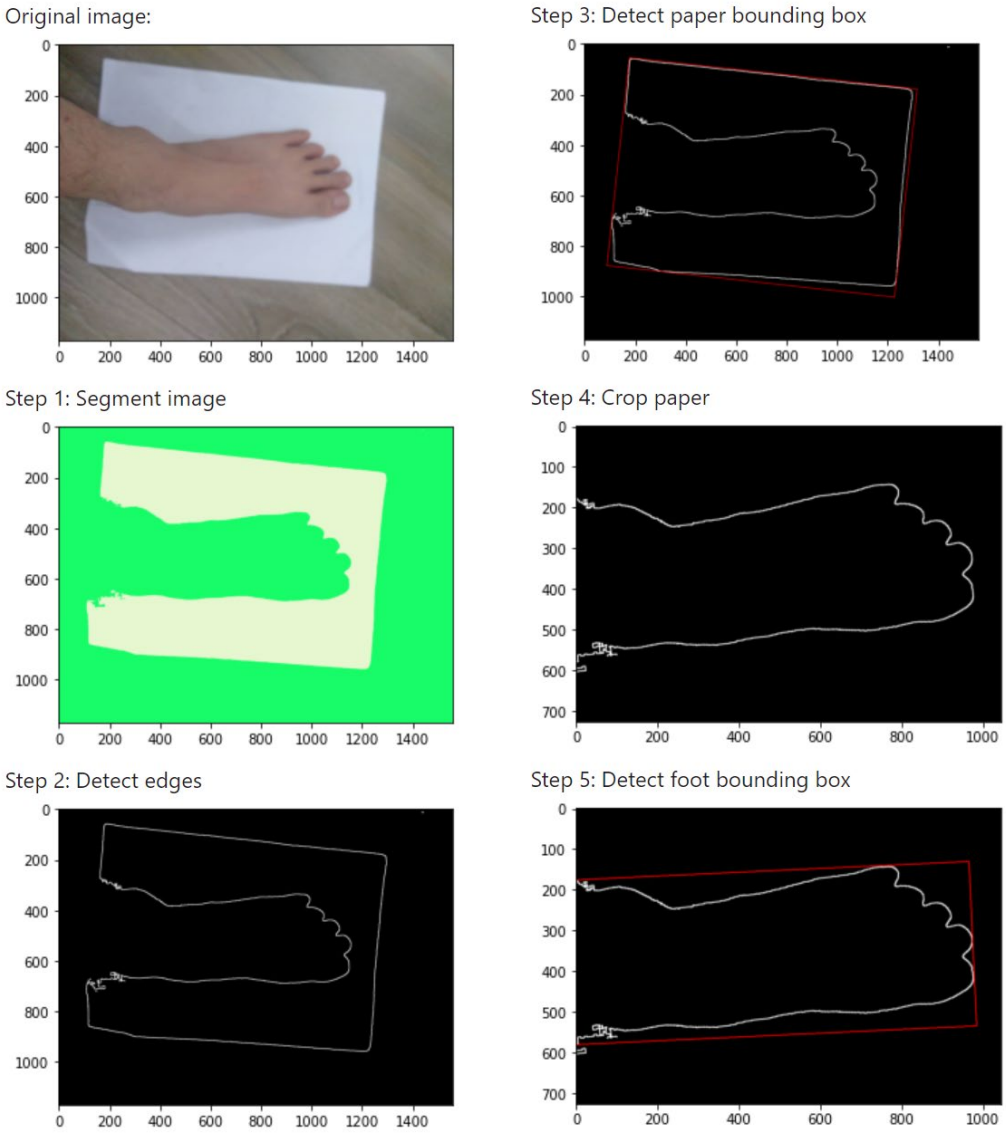

Figure 1: result of foot measurement.

2D Pose Estimation

In our attempt to construct a large-scale foot benchmark for virtual shoe try-on tasks, we faced several challenges that hindered our progress. One of the biggest challenges was the poor performance of the training results using Carnegie Mellon University's

Human Foot Keypoint Dataset, which was expected to serve as the basis for our benchmark.

Despite following the official code provided with the dataset, we were unable to achieve the expected results. The performance of the model was very poor, and we struggled to identify the root cause of the issue. One possible explanation is that the dataset is characterized by cluttered backgrounds, making it difficult to isolate and extract the foot keypoints accurately. Additionally, the proportion of feet in the dataset is quite small, which can further hinder the effectiveness of the training data.

In our efforts to overcome these challenges, we explored various approaches, including adjusting the hyperparameters of the model and fine-tuning the architecture. However, these attempts did not yield the desired improvements in performance. As a result, we were unable to construct a satisfactory benchmark for virtual shoe try-on tasks using this dataset.

Despite these setbacks, we recognize the importance of continuing to explore and develop large-scale foot benchmarks for virtual shoe try-on tasks. With the increasing popularity of online shopping and virtual try-on technologies, accurately modeling and predicting foot shape and size has become more crucial than ever. Therefore, we plan to continue our research in this area, exploring alternative datasets and approaches to address the limitations we encountered in this project.

Markerless Tracking System

We couldn’t get the outcomes we were hoping for because of problems with OpenPose practice. However, we discovered that a trained framework already existed in Deep AR. After some investigation, it was discovered that this framework's implementation strategy was largely similar to our study plan. Therefore, we ultimately decided to continue the development using Deep AR.

After doing markerless tracking on DeepAR, we tested the web-based application and imported the custom model. After that, we released the program online, giving customers the option to utilize it on their own laptops or mobile devices. The figure displays the final outcome.

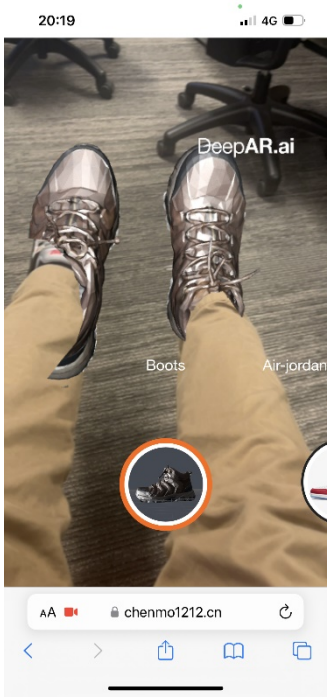

Figure 8: final product effect

Summary

This article discusses the use of computer vision technology in augmented reality (AR) applications, particularly in the context of virtual try-on experiences for shoes. The article explains the process of 2D pose estimation, which is the process of estimating the positions and orientations of body joints from a 2D image or video, using the open-source library OpenPose. It also describes the use of markerless tracking systems, such as deepAR, to track objects in real-time without the need for special markers or tags. The article further explains the importance of using high-quality 3D models of shoes in order to achieve accurate and realistic virtual try-on experiences, and how photogrammetry and computer-aided design tools can be used to create these models. Overall, the article provides a comprehensive overview of the key technologies involved in creating virtual try-on experiences for shoes, highlighting the importance of computer vision in this field.

References

[1] Yang, J., Wang, Y., Zhou, J., & Jiang, Y. (2017). Automatic foot measurement system using 3D scanning and feature point detection. Measurement, 111, 104-112.

[2] Hishinuma, Y., & Matsubara, M. (2018). Foot shape measurement system using a smartphone camera. Journal of Foot and Ankle Research, 11(1), 1-7.

[3] Lai, J., Xie, X., Wang, L., Xie, L., Li, Z., & Li, L. (2021). A deep learning-based foot size measurement system using multi-view 2D images. Sensors, 21(3), 955.

[4] Wang, S., Li, K., Huang, Y., Wu, J., & Zhang, Z. (2019). 3D Foot Surface Measurement System Based on Structured Light Vision. IEEE Access, 7, 23868-23878.

[5] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2019). Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In CVPR (pp. 1302-1310).

[6] Moon, G., & Lee, J. (2019). Human foot keypoint dataset and its application in shoe try-on. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 32-33).

[7] Schops T, Sattler T, Pollefeys M. A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM[J]. 2019:26-36.

[8] Facebook. DeepAR: Neural Network based Markerless Augmented Reality Platform [EB/OL]. (2018-05-15)[2023-05-04]. https://www.deepar.ai/.

Contributions

Since this project was completed by two people, me and Zhiqiang Cheng. We had assigned tasks: I was mainly responsible for the Foot Measurements and 2D Pose Estimation part, and he was responsible for the Markerless tracking system part.