SHOE AR RECONSTRUCTION

A Project Report

Present to
Dr. Chris Pollett

Department of Computer Science
San Jose State University

In Partial Fulfillment
Of the Requirements for the Class
CS297

By
Sneh Kothari
December 2022
INTRODUCTION

Augmented Reality is the integration of digital information with the user’s environment in real-time. AR users experience a real-world environment with generated perceptual information overlaid on top of it. AR applications are growing in popularity and are being adopted by developers in their products.

Another product that has been continuously gaining more and more attraction from people is sneakers. It has become a style statement for people. This project is aimed to cater to people of these two factions. Imagine you go to buy a pair of sneakers and you love them but want a second opinion. The normal way is to click a photo of the sneaker and send it to people. The issue with this method is that a 2D image does not capture the sneaker correctly. Hence the opinion you receive is not accurate. This problem escalates even further when you are shopping for someone. If you are buying sneakers for someone they would want multiple images and then try to figure out how the sneaker will look. It would still be difficult to visualize the complete sneaker. Hence to solve this problem we are building Shoe AR Reconstruction.

The basic idea of the Shoe AR Reconstruction app is that when you are in a store you can scan the shoe by clicking multiple images of the shoe. Following this, we build a 3D model of the shoe and provide the user with a 3D model of the shoe which the user can view in the app. The user can also share the 3D model with other users of the app. This can be done by sharing the deep link of the model. Any user that has access to this deep link can view the model in the
app. The model can be viewed by placing it in the real world. So the user can point and place the object on a plane and interact with it to get the proper feel of the shoe and make a better more informed decision when shopping.

Another use case for this app is in the eCommerce space. So this can be an offering from a company like Amazon to its sellers where the sellers can give a better feel to their customers. Sellers can build a 3D model of their shoes by scanning them and then allowing customers to see them in 3D and even try them on. This can be a real differentiator in generating more sales for the sellers.

Currently there are some companies actively working in this domain and trying out new things. Like there is a company called Wanna Fashion which is aimed at trying AR products. They have expanded the category and introduced watches as well. The key difference between our idea and their app is that they rely on 3D models provided by companies or preexisting models. We are currently working on making an app that allows you to make and share models.

Amazon also recently announced its entry into virtual shoe try on phase. In this scenario as well Amazon has mentioned only companies like Nike and Adidas will offer it. This indicates that the sellers need to provide the models for now. Amazon has claimed they will be expanding to more sellers which also means that they are working on the mesh generation technology and there should be some tool or research available soon.
Deliverable 1: AR App with teapots

The goal for this deliverable is to build a basic AR app. This deliverable is aimed at getting familiar with AR frameworks and also deciding on what technologies, frameworks, and tools to use in building the AR app. It is also aimed at understanding the different types of interactions that we can have with an AR object.

In the weeks leading up to deliverable 1, we read a paper that was a comparative study between ARKit and ARCore. This helped me in understanding the pros and cons of these two frameworks. Following this, we spent a week reading about ARKit and all the frameworks that make up the AR experience in an iPhone.

In the first week of the deliverable, we further explored the frameworks that make up the AR experience. We came across RealityKit which uses information provided by the ARKit framework to integrate virtual objects into the real world. This framework allows importing fully formed meshes into the real world which we will be using for our teapots in this deliverable. It also supports interactions (translation, rotation, etc.) on the object and can allow changes based on environmental changes. We also came across other frameworks which could be used in place of RealityKit but we finally settled for RealityKit because it is the latest framework and the most powerful and optimized one.

One more framework that showed promise was Metal but it has a steep learning curve hence we stuck to RealityKit.
The app uses a mixture of UIKit and SwiftUI to build the UI of the app. The app also has a 3D model of a teapot. Once you open the app and tap on New Room, it first asks permission for your camera and then you can view the room from you on the screen. Now when you tap on the floor you place the pot there. This is done by recognizing the tap gesture and if that is a horizontal plane then anchoring the teapot to the ground at that location. Every time you tap on the screen you place a pot on the screen. The app also allowed you to move virtual objects. This app shows some of the capabilities of the AR frameworks and will be used in the project when building AR Shoe Reconstruction.

Deliverable 1 demo screenshot
Deliverable 2: AR app to make 3D mesh from 2D image data

The goal of this deliverable is to explore a way to build a 3D mesh. It is also aimed at exploring the way apps can capture data from different sensors on the phone and how we can use that sensor data to build a mesh.

The initial idea for building the app was to use the Lidar sensor on the phone and get depth information using that. Finally, this sensor data can be used to build our 3D mesh. There were blogs about scanning entire rooms using the Lidar sensor. So the idea was to scan it like a room and crop out the object and use it in our app. The issue with this is that the sensor is only available on Pro models of the iPhone. Hence this idea was dropped due to the lack of access to a Pro phone and which would limit people from using it.

While looking for other ways to make a 3D mesh from 2D images we also found that Apple provides the Photometry API. This API accepts 2D images and converts those images to 3D models. This seemed like a reasonable idea for the deliverable but has a lot of shortfalls. So for deliverable 2, there was a lot of time spent on what works for our case. For deliverable 2 the code is majorly from Apple’s sample project on creating a 3D model from 2D images.

This deliverable has 2 apps, 1 app for the phone where the user clicks multiple photos of the object from the app. The images needed to be clicked using the app because the app utilizes the dual camera on the iPhone and provides a depth and a gravity file both of which are needed to build a 3D model. We click multiple images of the object using this app. In the sample example provided by
Apple, they needed to click more than 300 images to create a mesh of a pot. So for our case of the ball, we clicked more than 200 images for building the mesh. After clicking these images we shared all the images with their supporting files to the laptop. The Photometry API is only available on the MacBook hence we need a second app on the laptop. These images were then used as input for the MacBook app and the photometry app built a 3D mesh out of this. This worked for our case where we clicked more than 200 images for a basketball.

The issue with using this system was that the Photometry API can only work on a MacBook hence for our actual use case we will need to have a server with macOS and XCode so that we can make the 3D model. This did not seem very reliable. Another issue is that the system needs a lot of images to make a 3D model. For a simple ball, the system failed to work when we had around 100 images.

Hence we realized that we need to write our 2D images into 3D model building code. Initially, the idea was to have everything run on the phone so we thought of running the model building code on the phone. The issue with this is that we could not find new papers on implementing this on phones and the old ones built using phones older than iPhone X did a bad job of making 3D mesh and were very slow. Since the phones are more powerful now it might still work for us but it would have to be run on the phone’s GPU which would require code to be written in Metal (steep learning curve).
So we decided to try the other idea which is to run the 2D image to 3D model code on a server. This way we can do the heavy work on the server with more processing power. Since this is run on a server we can use Open Source frameworks and libraries built for this purpose. Also, we did not have to be restricted by Metal or any language.

So for the final project, we are getting image data from the phone and then uploading that data to the server where we will be doing image stitching. Image stitching has various open-source libraries we will be trying out OpenCV’s image stitching library for our purpose. This model will then be accessed by the app.

Deliverable 2: Ball 3D model
Deliverable 3: Exploring Sharing Options

The goal of this deliverable is to explore different ways to share the 3D model. Trying different ways to share the 3D model and since now we will be getting the model from a server are there any more ways to share the 3D object? The initial idea when starting this was to allow 3D object sharing via iMessage. The issue with this idea is that Apple provides a custom option to view it using its API which does not give a great personalized experience. Another issue is that this object is stored on the user’s iCloud and this might lead to an issue. One more issue that can happen is that iMessage limits the size of messages hence we did not face issues in our tests but it might happen. Hence since we already integrated with the server for a 2D image to 3D model building. It was intuitive to use it for model sharing. So we finally decided to upload the 3D model to an S3 bucket which can be accessed by other users using the object’s id.

The sharing problem remains in this case where we get the S3 link for the object. So we implemented deep linking which is we gave our app a unique identifier and when the object will be shared it will embed the S3 id in this link which will be used by anyone with this link to download the 3D model and view it in the app.

Since the goal of this deliverable was to explore sharing we implemented a basic version where based on the id passed to the app a unique URL was generated which led to a random image being downloaded. This was a way to
show that we can dep link apps and based on the id fetch data. S3 is the sharing option we will be using for our final project.
Deliverable 4: AR app to render shoe model on screen

The goal for this deliverable was to explore different user experiences while interacting with the shoe model. It was basically to decide the optimal way to show the shoe on the screen. There were two ideas for how to implement this deliverable.

The first idea is that when the app is opened and the shared shoe needs to be presented to the user we should fix the shoe at a particular distance from the camera so that however the user moves the camera the shoe is always at a fixed distance. This seemed like a great option but the issue here is that in Apple’s ARKit framework every 3D object needs to be anchored to one of the planes. So the shoe has to be anchored on a horizontal or vertical plane which cannot work if it always has to be at a fixed distance from the camera. Hence this idea was dropped.

The second idea was to allow the user to place the shoe on the horizontal plane in front of him by tapping on the plane. This anchors the shoe at that location. Now users can walk around that point to understand how the shoe looks. This also seemed a bit tedious which is why we built the rotate gesture where the user can hold the shoe with one hand and swipe with the other hand to rotate the shoe and look at it from all angles. The shoe can also be resized and moved for a better view and to look at specific parts of the shoe.
Conclusion

Shoe AR Reconstruction is an AR app that can essentially be useful for users. Augmented Reality is an upcoming and interesting field with a lot of scope in the future. This semester we worked on different aspects of the final project. We broke down the project into different smaller components and tried building the most basic parts of it this semester to work out all the clinks. We worked on 2D image to 3D model building followed by sharing the model with different users and then explored ways to interact with this 3D model. All these parts done on simpler objects are the building blocks for our project next semester.

In the second semester of the project, we will focus on building an app combining skills gathered from these building blocks. We will also be working on optimizations in instances like when a user gets a 3D model link then downloading the object and displaying might be time-consuming. We will be needing to deploy the 2D image to the 3D model code on the server which will be a challenge.