Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Augmented Reality Project Report

Finn Voorhees

May 2021

Declaration

I hereby declare that this is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed: ____________________ Date: ____________________

Contents

1 Introduction

While there are plenty of practical application for augmented reality (1), AR can also provide a fun way to interact with software in the real world. Augmented reality games have become increasingly prevalent as camera and sensor technology progresses in mobile devices, and having an AR-enabled device in your pocket enables many new applications. The goal of this project is to create a real time collaborative 3D AR drawing app for Apple devices. This application will enable users to look through their device camera and draw lines in the air based on where their device is pointing. Instead of relying on a two-dimensional canvas, users will draw in three dimensions in the real world. Two devices in close proximity will be able to draw together in real time, with each person's drawings appearing on other devices in the correct position.

A collaborative AR drawing app can have many uses. It could be used in a commercial setting to train individuals in a factory and indicate where things are by drawing paths and lines in AR, or it can be used in an educational setting to write equations and lessons in the real world. AR drawing also provides a unique medium for creative expression, as typical drawing is done on a 2D surface.

The motivation for this project is to experiment with how AR can be used for three-dimensional drawing, and learn how AR frameworks can be used to easily create augmented reality experiences without needing to focus too much on the underlying mathematics and computer vision techniques.

2 Background

2.1 Collaborative Drawing

Collaborative drawing software can enable multiple people to draw on the same "canvas" on different devices. This allows students to work on assignments together, designers to create designs together in real time, and many more possibilities. Software has been created in the past for real-time collaborative whiteboard drawing. Ringe et al. (2) developed a web-based tool for collaborative virtual whiteboard drawing and found that this approach enables interactivity and dynamic behaviour beyond what a physical whiteboard can offer. A collaborative drawing tool allows people to interact with each other independent of their location or device. A software tool also enables a larger canvas that would be impractical as a physical whiteboard.

2.2 3D Drawing

3D drawing software has existed for creating 3D models and CAD drawings for a long time, however only recently with the advances in VR and AR technologies has 3D art-based drawing been developed further. Drawing lines and shapes in three dimensions can be difficult to represent on a 2D computer screen due to the lack of a third dimension to understand depth. Models can be rotated and moved, but it can be difficult to get a sense for scale and placement when looking through a 2D viewport. With the technological advancements in VR technology, however, 3D drawing has been explored further.

Tilt Brush[1] by Google is software available for mainstream VR headsets that enables painting in 3D space through VR. Tilt Brush contains many advanced drawing tools and brushes, and allows the environment around you to be changed to different scenes. Ho et al. (3) found that "immersive VR can promote students' motivation and interest in learning 3D animation. However, the practical application of this

[1] https://www.tiltbrush.com

technology requires solving problems related to hardware and space".

While VR drawing is an interesting medium, drawing in VR lacks environmental context (4). Drawing in AR can provide an important source of reference or inspiration. A study done by Chang et al. (5) found that projecting drawings into the real world using AR resulted in more accurate drawings than using a flat surface, and people preferred to draw in the air.

3 Implementation

The final result of this project is an application for Apple devices that allows collaborative AR drawing in the real world. The application was created using Apple's *ARKit* framework, which enables the usage of iOS device camera and motion features to create augmented reality apps. The app lets users look around using the device camera, and holding the draw button will start drawing at a predefined distance from the device in real world coordinates. Users can move and look around to see drawings staying at a set location in the real world, and lines can be drawn all around. These drawings are shared with connected devices nearby and are anchored to a world map so that drawings will appear in the same real-world position independent of the device they are shown on.

3.1 ARKit

The interface of the application consists of an `ARSCNView` and a few buttons. According to Apple, `ARSCNView` is a "view that blends virtual 3D content from SceneKit into your augmented reality experience"[1]. This is the main view of the app and handles the rendering of the device camera as well as 3D content on top of it. `ARSCNView` will show the device camera and allows 3D models to be placed in world-mapped coordinates using the *SceneKit* framework. The SceneKit framework provides easy generation of 3D models and assets and is used to create the 3D drawings placed in the world. On each frame of the `ARSCNView` rendering, it checks whether the draw button is held down. If the draw button is being held down, a cylinder is created of a pre-set radius. The cylinder model's material is set based on the selected color in the color picker. The renderer will then retrieve the `ARSCNView` point of view transform, which represents the current camera position in the `ARWorldMap`. The `ARWorldMap` maps numerical coordinates to the proper location in the camera display. The renderer then uses this transform matrix to compute a point that is a set distance away from the camera, and it will move the cylinder to this

[1]https://developer.apple.com/documentation/arkit/arscnview

point. The cylinder is then rotated and connected to past cylinders to create a smooth stroked line. Each cylinder is attached to an `ARAnchor`, which contains the position and orientation of the object relative to the `ARWorldMap`. ARKit allows objects to be placed in the real world without having to worry about view transforms or projections. When the device is moved around the world, these cylinders attached to `ARAnchors` will automatically be rendered in the correct position in the `ARSCNView`.

3.2 Multipeer Connectivity

When the application is first opened, the `ARSCNView` will attempt to construct a world coordinate system based on the camera feed, device sensors, and LiDAR data. This coordinate system will be stored in an `ARWorldMap` object, and once a device constructs this object it can be shared with devices around it to synchronize the locations of objects in 3D space. The `ARWorldMap` is shared with nearby devices using the *Multipeer Connectivity* framework, which uses Wi-Fi and Bluetooth to discover devices and share data. The Multipeer Connectivity framework provides a high level abstraction around complicated networking code, and it is relatively easy to use to discover nearby peers and share data. Once nearby devices are connected, the `ARWorldMap` is shared to establish a common coordinate system across devices. This allows 3D objects to appear in the same real-world position. When an `ARAnchor` is created on one device, the position, color, and attributes of the anchor will be shared with nearby devices in real time. When a device receives anchor data, it will automatically add it to it's local scene to be rendered on screen. This transmission of drawing data enables seamless interaction between multiple devices and displays drawings from other devices in real time.

4 Evaluation

4.1 Results

The end result of this project is an application that enables real-time collaborative 3D drawing in AR. The app can be evaluated based on world tracking accuracy and "feel" of drawing in 3D.

The world tracking accuracy of the app was evaluated by using the application and moving around the real world. Once the device captures a world coordinate system, lines placed using `ARAnchors` are anchored to a real world position. Walking away from the real world position will move the anchors and drawings out of view, however moving back to that position will display them again. In testing it was found that once a coordinate system is captured, the device is very accurately able to determine where anchors should be. The only situation in which this didn't work as well was when there were many objects around that confused the AR tracking. It can be difficult to determine many objects in a scene, however LiDAR data may improve this.

The shared world map for collaborative drawing was very accurate. It occasionally becomes out of sync as the device has trouble identifying key features, however re-sharing the world map between devices occasionally fixes any issues caused by this.

The "feel" of drawing in 3D using the application was experimented with by changing the transform matrix used to generate the line drawings and changing various parameters of the lines. Generating lines directly at the camera's point of view matrix results in lines taking up the entire viewport and clips the camera inside of them. While this does emulate drawing directly where the camera is, this did not provide a nice drawing experience. Because of this, the lines were generated based on an offset calculated from the direction of the camera transform. The distance along the direction vector was experimented with to result in a drawing experience that felt like the device was a pen or marker. The width of the generated lines was experimented with to find a value that made sense in the real world, however settings could be added to the app in the future to allow users to change the stroke width based on how they want to draw.

4.2 Limitations

While the final application largely achieves the initial project goal, there are still some limitations.

In order to make drawings visible from any direction, they are represented as a set of cylinders. Drawings are made up of many cylinders placed end to end, meaning small variations in direction appear as smooth curves in the drawing. Larger variations in direction, however, lead to gaps in the cylinders. This breaks the appearance of a smooth stroke, and should be developed further to match up the ends of cylinders.

The application also has trouble creating a world map in a very busy environment with complicated lighting. This is a limitation of AR in general, and will be improved with further versions of the ARKit framework. Advances in AR technology will open up the possibility for more accurate world tracking that may utilize sensor data beyond a simple camera stream.

4.3 Further Work

There are many possibilities for further work on this project. One thing that can be done to reduce the limitations described above is to create a custom stroke model instead of relying on cylinders. This model could be procedurally generated based on the points drawn by the user, and would allow more complicated features like end caps and line join styles. Different types of strokes and stroke widths could also allow more creative drawings to be made.

Another possibility for further work on this project is to implement realistic lighting or reflections. ARKit has support for estimated lighting, meaning generated objects can be shaded based on estimated scene lighting information associated with live video frames. This would improve the illusion of drawings appearing in the real world.

An option to export drawings from the app could be interesting to add in the future, and would allow people to save drawings for future use. This could be done by exporting the 3D models to a 3D file format such as `.obj`. Exporting models would, however, lose all sense of real-world placement.

4.4 Conclusions

While there have been 3D drawing tools developed in the past, many of them have used VR to draw in 3D space. A 3D drawing app that lets you draw in AR enables a new medium where drawings are tied to real world positions. Having an AR enabled device in your pocket at all times makes this easier than ever, however augmented reality has still not gained huge popularity. One reason for this may be the lack of a proper device for displaying augmented reality content. Holding a phone in front of you to view AR content limits interactivity with content, as your hands are not free to be used. A phone screen also has a limited size display, and must be pointed close to where you want to see.

A solution to these problems that is being actively worked on is an AR headset or glasses. These devices can be worn on the head and AR content is displayed directly in front of your eyes. The HoloLens[1] is one device that does this, and it is rumored that Apple is working on development of an AR headset. A benefit of an Apple-designed AR headset is that custom software written for it would most likely utilize the existing ARKit framework. This means that when a headset is created, the application developed for this project would most likely be able to run on the headset with little modification. Because an AR headset does not require the use of your hands, the app could be modified to create drawings where your hands are in the world. This would enable a whole new way of drawing and would be easier to make precise details.

A 3D AR drawing tool has many applications, from student learning to creative expression. Collaborative drawing can be used in an office setting to share ideas, or it can just be a fun way for people to draw together. As AR technology advances, the possibilities for a 3D AR drawing application only expand. It will be exciting to see how the world utilizes AR in the future, and some time in the future it may be normal for people to collaborate in 3D space.

[1] https://www.microsoft.com/en-us/hololens

Bibliography

[1] A.Y.C. Nee, S.K. Ong, G. Chryssolouris, and D. Mourtzis. Augmented reality applications in design and manufacturing. *CIRP Annals*, 61(2):657–679, 2012. ISSN 0007-8506. doi: https://doi.org/10.1016/j.cirp.2012.05.010. URL https://www.sciencedirect.com/science/article/pii/S0007850612002090.

[2] Swati Ringe, Rishabh Kedia, Anuj Poddar, and Sahil Patel. Html5 based virtual whiteboard for real time interaction. *Procedia Computer Science*, 49:170–177, 2015. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2015.04.241. URL https://www.sciencedirect.com/science/article/pii/S1877050915007504. Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15).

[3] Li-Hsing Ho, Hung Sun, and Tsun-Hung Tsai. Research on 3D Painting in Virtual Reality to Improve Students' Motivation of 3D Animation Learning. *Sustainability*, 11(6):1–17, March 2019. URL https://ideas.repec.org/a/gam/jsusta/v11y2019i6p1605-d214476.html.

[4] John J. Dudley, Hendrik Schuff, and Per Ola Kristensson. Bare-handed 3d drawing in augmented reality. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, page 241–252, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450351980. doi: 10.1145/3196709.3196737. URL https://doi.org/10.1145/3196709.3196737.

[5] Yun Suk Chang, Benjamin Nuernberger, Bo Luan, and Tobias Höllerer. Evaluating gesture-based augmented reality annotation. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 182–185, 2017. doi: 10.1109/3DUI.2017.7893337.

A1 Source Code

The source code for this project can be found at
`https://github.com/finnvoor/DrawAR`.