Augmented Reality Hologram

Jia Jun Gan
School of Computer Science and Engineering
Nanyang Technological University (NTU)
Singapore 639798
ganj0014@e.ntu.edu.sg

Owen Noel Newton Fernando
School of Computer Science and Engineering
Nanyang Technological University (NTU)
Singapore 639798
ofernando@ntu.edu.sg

***Abstract*— This paper presents the development of a mobile application that enables the creation of holographic objects and displays holographic objects on mobile devices. This project aims to demonstrate how the functionalities are implemented, and testing would be conducted to ensure the accuracy of creation and exhibitions of the holographic objects. During the app testing, there would be two approaches, which were tested based on two different lightings conditions. Users would be able to create their holographic objects with minimum lighting changes based on the results of the two approaches.**

Augmented Reality; Green Screen; Holograms; Interaction;

I. INTRODUCTION

In today's modern technology, the presence of viewing the virtual world is rapidly growing in the real world. The virtual world has become more vibrant and realistic. With the rapidly evolving technology, Augmented Reality, it has been rapidly changing and marking its presence in the technology landscape. Augmented Reality (AR) can help with inspecting holograms. However, such an approach poses unique challenges regarding image capture, matching and, in particular, user guidance, resulting in high temporal and cognitive effort [1].

The creation of the AR objects is often from the companies, which are in the media industry, with professional lightings and cameras. Consumers would like to create their holographic object without actually spending money to purchase expensive equipment. Therefore, the objectives of this project are to develop mobile hologram application for Android by making use of the latest AR frameworks such as Google ARCore. This app should achieve to allow users to:

- Create own holographic models from the app through green screen
- Place and interact with a holographic model or figure into a scene for video capture [2]

The following sections, Related Work, Implementation, App Integration, Discussion and Conclusion, would be further described.

II. RELATED WORK

There are various mobile applications which can place holographic models in the camera like Holo [3, 4]. Besides, there are a few green screen chroma key mobile applications, which could shoot videos and photos using chroma key effects like ChromaVid [5].

Firstly, for Holo [3], it is capable of adding holograms to the real world through the camera, as shown in Figure 1.

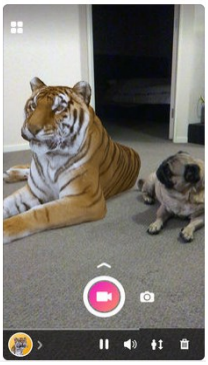

Figure 1. Holo Mobile App [1]

Figure 2 explains that the 8i team (Holo Creator) [6] has been continuously creating their 3D models in their studio by using photographic equipment for professional video taking behind a green screen as shown in Figure 2. The data will be processed and converted into a video asset [6].

Figure 2. 8i 3D Models Creation Process [6]

However, even though Holo is able to produce new holographic models constantly, Holo App users are unable to create their holographic models and use it on the app. Secondly, for ChromaVid [5] as shown in Figure 3, it acts as a photo or video editor which crops out the Chroma colour (blue, green, yellow or red) background and change to any background of the user's choice.

978-1-7281-2297-7/19/$31.00 ©2019 IEEE
DOI 10.1109/CW.2019.00064

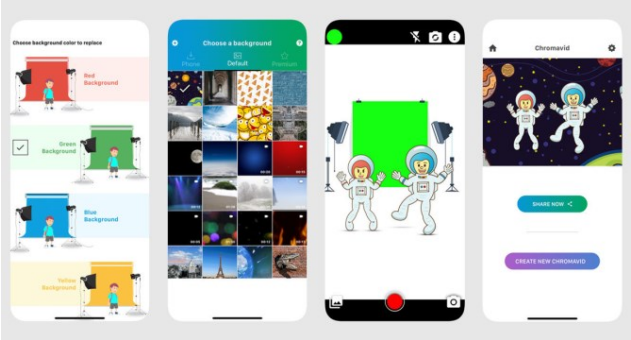

Figure 3. ChromaVideo App Walkthrough [5].

However, according to a review in Google Play [7] as shown below in Figure 4, the Chroma background colour must be exact, which make it unusable in most environment. There might be many lighting issues, which causes the Chroma colour to variate and in turn unable to crop out the images/videos properly.

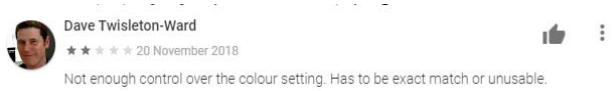

Figure 4. Google Play Review on ChromaVideo App [7].

Therefore, this project focuses on overcoming those issues by creating functions, which allow users to create their holographic model behind a green screen in the most environment with inconsistent lighting.

III. IMPLMENTATION

AR Hologram has been created using Android Studio version 3.1 or higher and Google ARCore [8]. ARCore is Google's Augmented Reality Platform for Android [7], which allows augmented reality applications to be built. ARCore uses Sceneform SDK, which is a 3D framework that provides a rendering of 3D models [9]. In order for the green screen to work, it should have the sceneform asset files, which consists of Sceneform Asset Definition File (*.sfa), Source Asset File (*.obj) and Custom Material File (*.mat). Sceneform Asset Definition File defines the source asset file and custom material file, which will be called from the Activity Class. Source Asset File indicates the vectors of the plane (AR Object), which is generated from Blender. Custom Material File defines the chroma key video material, which indicates the threshold of the "Green" being filtered out as transparent.

For the features of AR Hologram, it allows users to create their own holographic objects (behind a green screen), place and interact with the newly or existing holographic object.

Firstly, for own creation of holographic objects, the app will have a camera view which allows the users to record the video of any object behind a green screen. The video will then be saved into the "My Holos" list, as shown in Figure 5 (Left).

Next, for the placement and interaction of holographic object, the users could choose to display the holographic object from a list of tabs (My Holos, Featured, Characters and Animals), as shown in Figure 5 (Left). The holo version of the subject would then be displayed in camera view, as shown in Figure 5 (Right).

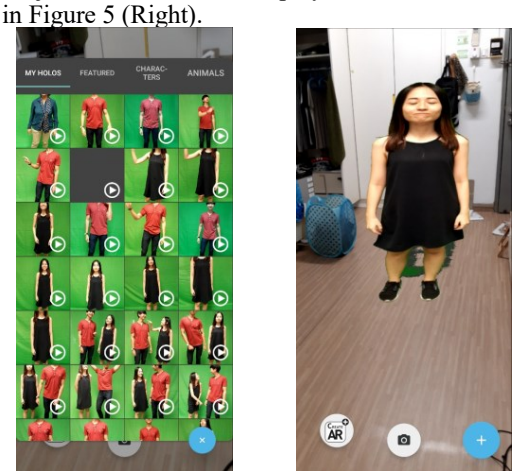

Figure 5. Choose Holo Objects. (Left) and Display of Own Holo created. (Right)

IV. APP INTEGRATION WITH GREEN SCREEN FILTER

A. *Green Screen Set-up*

There are a few things to take note when setting up the green screen [11], the overall lightings around the area and the best area to set up the green screen and so on. In every part of the green screen, it must have consistent light. The shades of green are consistent, which means that when cropping out the person from the green screen, it could crop out the object from the Chroma green completely.

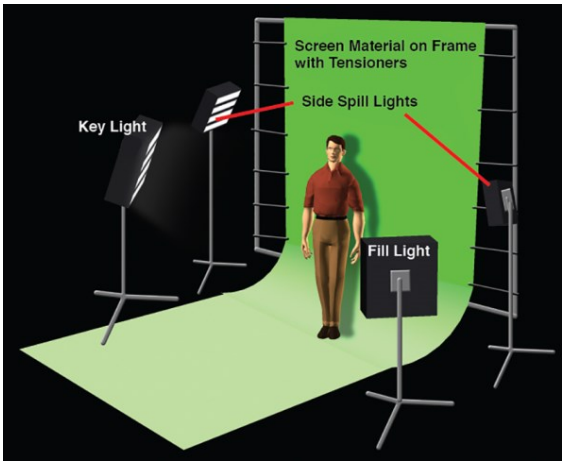

Figure 6. Green Screen Setup [10]

The setup of lighting must have a key light, two side spill lights and fill light, as shown in Figure 6. The key light is shown directly on the subject to light up the subject. The fill light is to fill in the light around the green screen and the

subject. It is used to make sure that the area, which is not lit well by the other lights, could have an equal spread of light around the area. The side spill light plays an essential role in ensuring that the shadow behind the subject is less prominent.

B. Problems faced in Green Screen Filter

Two issues were faced during the development of the app, Chroma Green Colour Issue and The Orientation of AR Object is not aligned correctly.

Under Chroma Green Colour Issue, the app is unable to filter out the green completely, leaving some green on specific areas with least lighting as shown in Figure 5 (Right). The AR object on the screen should be in an upright position. This is due to the original aspect size of this video is in landscape mode, and the app could only read portrait mode. This causes the misalignment in the AR Object. This can be shown in Figure 7 (Left).

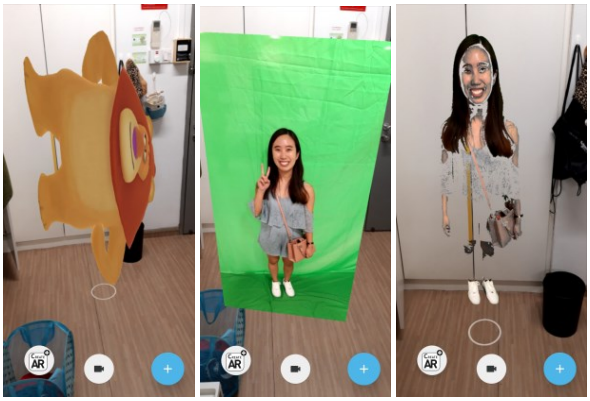

Figure 7. Misaligned AR Object (Left), Threshold at 0.3 (Center) and Threshold at 0.9 (Right)

1) Correction method for Chroma Green Colour Issue

The solution to this is to change the code to the following algorithm.

In Figure 8, the value in the threshold [12, 13] will determine if more or fewer shades of green would be filtered out. The higher the threshold, the more shades of green would be filtered out. By testing the AR object on the screen at a different threshold, 0.3, 0.675 and 0.9, the results are shown in Figure 7 (Center), Figure 5 (Right) and Figure 7 (Right) respectively.

Algorithm 1: chroma_key_video_material.mat

```
1   Calls: prepareMaterials( ) method for layout initialisation
2   Initialisation: uv ← current UV Texture Coordinates from camera
3   if object is not facing the front then
4       uv.x ← 1.0 – uv.x //invert the x axis of the UV Texture Coordinates
5   Initialisation: colour ← texture colour (rgb) of input video material
6   Initialisation: keyColor ← colour to filter out (chroma green colour)
7   Initialisation: threshold ← 0.675
8   Initialisation: slope ← 0.2
9   Initialisation: distance ← difference in colour shade between colour and keyColor
10  Initialisation: edge0 ← threshold * (1.0 – slope)
11  Initialisation: alpha ← get step value between edge0, threshold and distance
12  colour ← desaturate colour
13  Set alpha value to output material
14  Inverse colour and assign to RGB (colour) of layout //make it transparent
15  Output material ← Combination of the RGB (colour) of layout with inverse colour //place transparent layer
```

Figure 8. Correction method for Chroma Green Colour Issue

2) Correction of the alignment of the Orientation of AR Object

Two different sfb files were created in landscape and portrait mode to ensure the alignment that change accordingly to the video aspect ratio. A conditional statement, if else statement was added to check if the video width is less than the video height (portrait mode). If it is a portrait, then chroma_key_video_2.sfb will be used as the source and vice versa. This can be shown in Figure 9.

Algorithm 2: Solution to misalignment of AR object

```
1   Initialisation: videoWidth ← get video width of the selected AR object
2   Initialisation: videoHeight ← get video height of the selected AR object
3   if videoWidth < videoHeight then //Video is in portrait mode
4       Build model by setting source with the formatted sfb file (chroma_key_video_2.sfb)
5           Set external texture of the model as videoTexture
6           Set filter color as CHROMA_KEY_COLOR
7   else //Video is in landscape mode
8       Build model by setting source with the formatted sfb file (chroma_key_video.sfb)
9           Set external texture of the model as videoTexture
10          Set filter color as CHROMA_KEY_COLOR
```

Figure 9. Solution to misalignment of AR Object

V. DISCUSSION

In two different occasions, there are two different approaches in setting up the green screen at the same location. The first approach (as shown in Figure 10 (Top-Left)) is making use of the different lightings, a main light at the ceiling and two side spill lights (external lightings). The second approach (as shown in Figure 10 (Bottom-Left)) is

making use of the main light at the ceiling without any external lighting as shown in Figure 10 (Top-Right).

Both of the approaches were implemented using the same phone, Samsung A9 and the same version of codes. For the first approach, even though the spill lights could help to light up the green background, there are some shadows around the leg area, as shown in Figure 10 (Top-Left). The result of Approach 1 is shown in Figure 5 (Right). It shows green at around the shadow area. The result of Approach 2 is shown in Figure 10 (Bottom-Right). It shows fewer green than Approach 1.

To conclude the two approaches, approach 2 is much more cost effective than approach 1. The result from approach 2 created less shadows than the result from approach 1. Even though the green screen background in approach 1 looks consistent in the photo, the shadows are more prominent when the subject does certain hand gestures or moves around. The reason behind this is that the main light, which is at the ceiling, was not bright enough to be able to shine on the subject. Approach 2 has a better main light, which is also at the ceiling as it was brighter and able to shine on the subject with the least shadow even if the subject moves around.

Figure 10. Approach 1 – Green Background (Top-Left), Approach 2 – Green Screen Setup (Top-Right), Approach 2 – Green Background (Bottom-Left) and Approach 2 Result (Bottom-Right)

VI. CONCLUSION

This paper introduced the AR Holograms, which has two main features, Create Own Holo – Taking Videos behind Green Screen and Add Holo on AR Scene – Move/Resize AR objects around. Two of the features are integrated to ensure that every users' are able to create their own holograms and show to the world immediately within the same application. The process from planning to implementation of AR Hologram was listed in this paper. The application was designed with thorough planning and consideration. Even though there are still some minor issues which were yet to be discovered, the app has gone through testing in two significant events, Prospectus Photo Shoot and Open House 2019. This application could also be used in multiple industries like entertainment and education, which shows the popularity potential of this application.

REFERENCES

[1] A. Hartl, J. Grubert, D. Schmalstieg, and G. Reitmayr, “Mobile interactive hologram verification”, in Proc. IEEE Int. Symp. Mixed Augmented Reality, 2013, pp. 75–82.

[2] Billinghurst, M. and Kato, H., “Collaborative Mixed Reality”, in Proc. IEEE 1st Intl. Symp. Mixed Reality (ISMR99), 1999, pp. 261-284.

[3] Holo - Mix Holograms into Photos & Videos - Holo. Retrieved January 22, 2019 from http://www.thisisholo.com/

[4] A. D. Hartl, C. Arth, J. Grubert and D. Schmalstieg, "Efficient Verification of Holograms Using Mobile Augmented Reality", in IEEE Transactions on Visualization and Computer Graphics, vol. 22, no. 7, pp. 1843-1851, 1 July 2016.

[5] Chromavid - The best Chroma key, green screen Mobile application. Retrieved January 25, 2019 from http://www.chromavid.com/

[6] 8i - Real human holograms for augmented, virtual and mixed reality. Retrieved January 24, 2019 from https://8i.com/technology/

[7] Google Play - Chromavid – Chroma key App. Retrieved January 25, 2019 from https://play.google.com/store/apps/details?id=com.appsformobs.chromavid&hl=en_SG&showAllReviews=true

[8] Package Index | ARCore | Google Developers. Retrieved March 18, 2019 from https://developers.google.com/ar/reference/java/arcore/reference/packages

[9] Sceneform | ARCore | Google Developers. Retrieved Match 18, 2019 from https://developers.google.com/ar/develop/java/sceneform/

[10] Basic Green Screen Lighting Techniques. Retrieved March 24, 2019 from https://www.newtek.com/blog/tips/basic-green-screen-lighting-techniques/ Ronald E. Anderson. 1992. Social impacts of computing: Codes of professional ethics. Soc Sci Comput Rev 10, 2: 453-469.

[11] J. Foster, “The Green Screen Handbook: Real-World Production Techniques, 1st ed. Indianapolis”, in Wiley Publishing, 2010.

[12] D. Rajan, "Weighted color and texture sample selection for image matting", in Proc. of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 718-725, 2012.

[13] W. Wang, J. Zhao, "Robust image chroma-keying: A quadmap approach based on global sampling and local affinity", Broadcasting IEEE Transactions on, vol. 61, no. 3, pp. 356-366, Sept 2015.