Realtime occlusion in Augmented reality

SURAJ* and HARSHIT PANWAR*, Indraprastha Institute of Information and technology

ACM Reference Format:
Suraj and Harshit Panwar. 2019. Realtime occlusion in Augmented reality. 1, 1, Article 1 (May 2019), 2 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In augmented reality one of the main problem is that the virtual object stays on top of the marker without interacting naturally with the real world. So to give more realistic feel this paper shows the approach for real time time Occlusion by comparing depth values of both real objects and virtual objects.

2 LITERATURE SURVEY

Occlusion is the one of the elusive pieces of the augmented reality puzzle, this is the ability to hide virtual objects in front of real objects. Figure 1 shows the virtual object on the top of the person which is giving it a very unnatural feel.

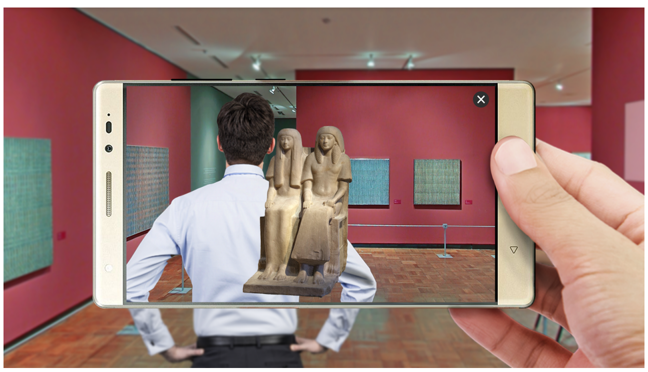

Fig. 1. Virtual Object without occlusion[3]

To solve this Problem there are so many approaches, some commonly used approaches are:

2.1 Model-based approach

This approach is very efficient and provide high quality of occlusion by the help of GPU. But for this high detailed model is needed and only applicable for static real objects. Therefore, fails in mixed/augmented reality.

*Both authors contributed equally to this research.

Authors' address: Suraj, harshit16043@iiitd.ac.in; Harshit Panwar, Indraprastha Institute of Information and technology, Delhi, Delhi, 110020.

XXXX-XXXX/2019/5-ART1 $15.00
https://doi.org/10.1145/1122445.1122456

2.2 Contour Tracking

This approach does not need high quality detailed model but need high power to tack silhouette contours by contour algorithm. The tracked counters are not that much accurate that's why this approach is only successful in case of opaque objects the occlusion doesn't work in case of translucent objects like glass. Also, this technique need a initialisation by user which is annoying and not acceptable.

3 ALGORITHM AND IMPLEMENTATION

We used matte and mask approach for implementing Real time Occlusion in AR, in this approach we are creating an alpha matte using alpha value in range [0,1] for every pixel, we assigned a m-value for every pixel which acts like an alpha value to get an alpha channel.

This approach is shown for kinect camera by using unity. Normal RGB cameras will won't work in this case, to compare depth values between real and virtual objects we need a RGBD depth camera like kinect.

For displaying virtual object select any marker (in our case we used our face as marker and virtual object as sphere).

After augmenting the object on marker, compare the depth of each pixel to the calculated depth value of object and marker, on basis of that pixels are categorized into 3 types:

- Infront: pixel whose depth is less then depth of virtual object, i.e. pixels infront of virtual object
- behind: pixel whose depth is more then depth of virtual object, i.e. pixels behind virtual object
- not known: for some pixels kinect does not give a depth value.

we assigned a new value m to every pixel, for infront pixels m=1 and for behind and not known pixels m=0 in figure 2 infront pixels are in white, for doing this we used an alpha matte and in which every pixel whose m value is 0 will be made transparent.

Fig. 2. Pixels with less depth are in white

4 RESULTS

All the listed tasks are completed:

- Ar with head as a marker.
- Depth values of pixels.
- Occlusion detection.
- occlusion Culling.

The figure 3 shows the complete occlusion culling using kinect camera on virtual sphere.

Fig. 3. A demo of result

Github: *https://github.com/cse-570-virtual-reality-2019/occlusion*

5 LIMITATIONS

In Our algorithm the upper layer(matte) is 1/30 seconds behind the below layer so for moving objects with m = 1 it fluctuates. RGBD cameras has some noises and it does not have a precise depth value per pixel so the occlusion Culling might shift from it's original position.

Kinect camera have also limitation for sensing depth. It can only measure depth upto around 4.5 meters of distance.

6 REFERENCES

- https://dl.acm.org/citation.cfm?id=3139153
- https://hackernoon.com/why-is-occlusion-in-augmented-reality-so-hard-7bc8041607f9
- Google images

Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 559âĂŞ568.

O. Kahler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S Torr, and D. W. Murray. 2015. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. IEEE Transactions on Visualization and Computer Graphics (Proceedings International Symposium on Mixed and Augmented Reality 2015 22, 11 (2015).

Denis Kalkofen, Erick Mendez, and Dieter Schmalstieg. 2007. Interactive focus and context visualization for augmented reality. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 1âĂŞ10.

M-O Berger. 1997. Resolving occlusion in augmented reality: a contour based approach without 3D reconstruction. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on. IEEE, 91âĂŞ96.

David E Breen, Ross T Whitaker, Eric Rose, and Mihran Tuceryan. 1996. Interactive occlusion and automatic object placement for augmented reality. In Computer Graphics Forum, Vol. 15. Wiley Online Library, 11âĂŞ22.