Star Lab Research & Development White Paper 001

Immersive & Interactive AR Graphics & Environments for Broadcast Applications

—---

April 2022

Authors
Caroline Stedman Mishra MA BSc, Saurabh Ranjan BTech, Sanjyot Dahale BE,

STARLAB
INNOVATIONS · SOLUTIONS

Abstract

This paper presents a Mixed Reality Toolkit solution for visualisation and interaction with broadcast AR graphics, with a multi-camera spectator camera solution so that the audience can see the same graphics that the presenter is interacting with. This solution was developed by Disney Star's R&D Lab, Star Lab. The purpose of the toolkit is to provide a more immersive experience for both audiences and presenters and to aid in storytelling around sports data in cricket broadcast programming.

This paper describes the background and inspiration for the project which started with the Hololens demos by Epic Games and Microsoft. It then goes on to describe the technical specifications and workflow of the system.

There is an evaluation which details the technical roadblocks encountered by the team and steps taken to solve them and/or possible future solutions that will provide improvements to the system. The evaluation also details the results of a subjective user test from both presenter and audience points of view.

The paper concludes with a summary of the state of the industry in the field of broadcast augmented reality graphics and how our product is novel in its capabilities and approach and plans for its next stages of development, as well as an outlook for the future of interactive AR graphics and virtual sets.

Additional keywords

Mixed Reality, Unreal Engine, Microsoft Hololens, AR graphics,Virtual production, Virtual sets, Interactivity, Metaverse, Sports broadcast.

Table of Contents

STARLAB
INNOVATIONS · SOLUTIONS

1. Introduction

Broadcast augmented reality graphics are widely used to demonstrate data points which are used for storytelling in programming, largely in factual based shows i.e. weather, news, sports, science etc. They are used in hard set environments as well as in combination with virtual studio green screen environments.
So far their use has been subject to technological restrictions whereby the presenters are unable to see the graphics and have to rely on comfort monitors which show their proximity to the virtual objects. Presenters have also been unable to interact directly with the graphics, so any dynamic changes to the graphics have to be implemented by an additional graphics operator or a hardware device controlled by the presenter, such as a clicker or gesture band worn on the hand or wrist.
In addition, the graphics generally appear as an alpha layer rendered in front of the presenter or a background layer in a green screen set-up, where the presenter is keyed in front of the graphic. If the presenter moves around the studio space, there is a risk of unwanted occlusion which 'breaks the magic' of the virtual object appearing to be a 3D physical object.
These restrictions on visibility to the presenter, interaction and free movement around a virtual object considerably limit storytelling abilities, natural looking interactions and other visual aesthetics.
The above is certainly true of traditional graphics systems like the earlier versions of VizRT. While presenter visualisation is still not currently a part of any commercial product, now with the integration of gaming engines into virtual production tools, virtual production software companies are beginning to address the challenges of interaction and unoccluded movement around virtual objects.

This paper describes a novel system for the control of broadcast augmented reality graphics which allows for real time dynamic interactions with data within a multi camera set-up and visualisation of AR graphics by a presenter via a HMD with holographic capabilities. The system has been designed with sports broadcast as the focus, specifically for the sport of Cricket. However, it is suitable for other types of non-fiction broadcast.
The system uses the Microsoft Hololens 2 as the primary viewing and interaction device via holographic remoting controlled from within Epic Games' Unreal Engine, integrated with Mosys camera tracking and Zero Density real time graphics rendering.
By utilising physics functions available in the game engine we can calculate ball trajectories based on moving data points produced by presenter interactions with match Hawkeye data, as well as the provision for biomechanics data and bat-ball trajectories from manipulation of 3D player mannequins.
The system also allows for fielding analysis based on player positional data integrated into the system as well as templates of commonly used fielding configurations. These can be overlaid with Hawkeye 'wagon wheel' data showing scoring zones for specific players.
There are additional features for calling up dynamic player stats and viewing/telestrating on multiple virtual video screens.

The short term goal of the application development is to enable new and enhanced tools for presenters for storytelling around sports data which are visually appealing for Disney Star's audiences, and an upgrade on what is currently possible with 'off the shelf' virtual production and AR graphics products.
In the longer term, the aim is to create the next generation of fully immersive and interactive environments using game engines for presenting television programming content. This would extend beyond singular AR graphics and we envision a fully interactive, tactile, virtual studio environment for immersive storytelling.
In the nearer future, additional data such as skeletal tracking and pose estimation data will be enabled in order to create 3D AR replays of sporting moments in near real time. These dynamic 3D assets will be fully interactive so that the presenter has the ability to play, pause, rewind and manipulate the model to understand what the result would have been if the player had been in a different pose. For example, we could show how if a bowler had moved his arm higher and made his elbow straighter, his delivery would have landed in an area of the pitch that would have likely got the batsman out.
The use cases of this technology do not stop here however. While we are at present developing these features for use in the studio, eventually interactive 3D data visualisations and 3D player replays can be pushed out to audiences via mobile or HMD devices. Volumetric video of entire sports arenas can become immersive virtual environments that presenters can move around in and visualise via an HMD, and these same environments can be provided to audiences to experience a match from within, any angle, any position. These same environments have the potential to be gamified and shaped by audience engagement data and feedback/inputs.
The technology also has further use cases outside of sports broadcast and LBE experiences.
As we explore the possible future of a tangible metaverse, we are creating the foundation technology now that can be used to build out such future audience experiences, moving out of the television studios and into our audience's living rooms.

2. Background

The Apollo 11 project created by Epic Games was used as an inspiration for our project. The development of the camera tracking system in this project was done in such a way that the system used a combination of 2 tracking data feeds, one coming from an optical tracking system and another from a mixed reality headset, both of which were mounted on top of a real camera. Using a combination of both these values, a virtual camera was controlled inside Unreal Engine. The real camera feed was fed into Unreal Engine and using the Composure plugin, the MR graphics that were captured by the virtual camera were composited on top of the real camera feed in real-time. In this composition the real camera feed was set as the background and the MR graphics were always set as the foreground. As this compositing system does not include a virtual background, switching the real camera feed to be in the foreground is not possible.

The spectator camera view for Hololens was first introduced in the Hololens 1. It was showcased in Hololens 2 demos by Microsoft where the audience were able to see the Hololens user interacting with the MR graphics from the point of view of a secondary camera.
Microsoft later introduced the spectator camera feature for the Unity Engine, using which we can integrate a secondary camera into the MR space to capture all the MR graphics that the Hololens user interacts with. This feature required one Hololens device to be mounted on top of the secondary camera. A major limitation with this system was that the camera parameters like the focal length & aperture of the secondary camera could not be changed once the calibration was done using a set of values.
The spectator view system was not introduced for the Unreal Engine. However, Epic Games created a similar system for the Apollo 11 project using a combination of a mixed reality headset and an optical tracking system mounted on top of the secondary camera. They used the data from both these tracking devices and set a virtual camera to capture the MR graphics from the secondary camera's point of view.
In our project we have eliminated the need of mounting a MR Headset on the secondary camera, instead we use a combination of mixed reality local anchors placed using the user's Hololens device and the Mo-sys camera tracking system to match the virtual camera with the real camera. This system also allows us to change the camera parameters such as focal length & aperture in real-time, as these values are also updated in the virtual camera. Additionally, as our project is designed on a multi-player system we also have the ability to integrate multiple cameras in the MR space.

3. The Mixed Reality Toolkit

The overarching idea of the entire system was to create a real-time interactive Mixed Reality experience with broadcast quality graphics, which can be showcased to an audience watching via linear television or OTT (without access to a Mixed Reality Headset). The system the we created to achieve this can be subdivided into three different parts as follows:

1. Spectator View / Third person view / Production system
2. Hololens operations
3. Pre-show operations & setups using interfaces for Operators.

3.1. Spectator View

The foundation of our Unreal Engine project is based on the multi-player system, through which we are able to separate the Hololens and the broadcast/spectator cameras as independent entities in the system.
The Hololens, as well as each individual spectator camera which is to be connected to the system, has their individual computer running the Unreal Engine project instance. The multiplayer system is based on a server-client model running on a Local Area Network in which the Hololens is always connected to the server computer and all the cameras are connected to the system as client computers.

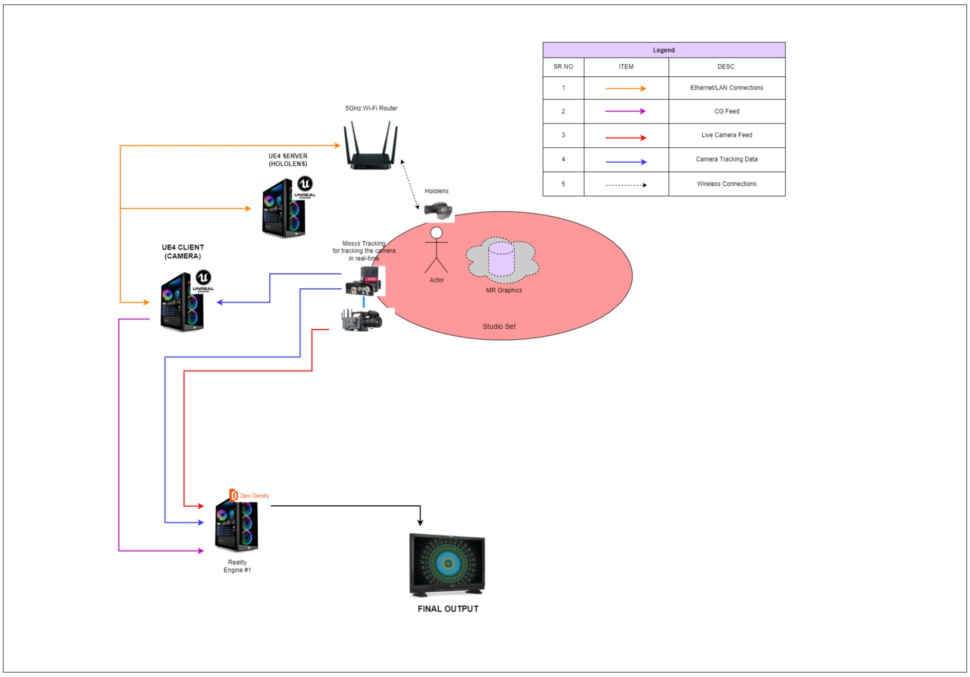

Figure 1 - spectator view workflow

The computers connected to the cameras receive live camera tracking data from their Mo-sys camera trackers which provide positional and pan, tilt & zoom data of the camera as well as all the intrinsic properties of the camera in real-time.

We use the camera tracking data to place a virtual camera in the MR simulation at the exact position, with all the properties of the real camera. Hence as the virtual camera moves in the MR space it captures the MR graphics with the right perspective and movements.
The captured visuals are processed by the Composure system in the Unreal Engine though SDI and the feed is finally sent to a Zero Density system designated for that camera for compositing with the RGB camera feed.

In this final composition we factored for two situations, one where the talent is standing in a physical set, and the other where the talent is standing in a green screen virtual set. When the talent is standing in a physical set, there is no virtual background & hence the real camera feed becomes the background and the MR graphics become the foreground.
When the talent is standing in a green screen set, we have the virtual background captured by the Zero Density camera, as the background and the talent captured by the real camera is keyed out from the green screen.
Hence here we have the ability to choose which feed becomes the foreground between the two remaining feeds, for example, the keyed out real camera feed or the keyed out MR graphics feed.

Depending on the kind of interactions that we wish to showcase, as well as depending on the position of the talent with respect to the MR graphics, we needed to be able to switch the midground and foreground feeds real-time.
To facilitate this, we created a script using Zero Density's custom scripting language called the Rgraph. Currently this switching between this midground and foreground is triggered on a keyboard button click, but in future updates we plan to automate this switching using technologies like the Zero Density Talent S Traxis, which can provide us the real-time 3D position of the presenters.

3.2. Hololens Operations

When the user wears the Hololens it scans the surroundings of the user and generates a mesh of its surroundings. It uses this mesh for calibration and detects the position of the user as well as Local Anchors in the real-world. Once the calibration is complete, the Windows Mixed Reality plugin allows us to track the user's hands including individual finger joints.
The MRTK plugin also allows us to use certain hand tracking gestures like palm detection and pinching, thus allowing the user to interact with the mixed reality graphics. We have used a combination of such gestures to execute different actions like opening the menu and interacting with the ball trajectories, etc.

When the user interacts with the MR graphics i.e. Menu, video wall etc. the interaction is received by the server system on which the Hololens is connected (via the WMR plugin), and the result of the action is shared with the client computers connected with the server system using multiplayer setup. Thus upon user interaction, we see the same results on the Hololens as well as on all the spectator cameras.

3.3 Pre-show operations & setups using interfaces for Operators

The Mixed Reality Toolkit is tailor-made for cricket Analysis and cricket based storytelling, hence it is important to integrate various databases which provide us with the critical data required to visualise specific analytical simulations.
We collaborated with Hawkeye to gather this data in the form of JSON files which provided us an exhaustive library of parameters regarding each ball played during different matches. The database from Hawkeye is a mix of historic, as well as real time data. The historic data contains information about all each ball delivery since 2018 for all the IPL tournament matches and several other tournament matches. The real-time database provides live match data with a latency of around 2-3 seconds.

We created a custom API to communicate with a remote server that hosts thousands of these JSON files and receives only those data parameters that are required for a particular simulation. To send these API requests from Unreal Engine, we created a front-end GUI for the operator, which they can use to set all the query parameters according to the requirement of the simulation and send an API request. Once the database receives the API request from Unreal, it runs a query that fetches each delivery and its parameters. This data is then sent back to Unreal for further processing of visualisation.

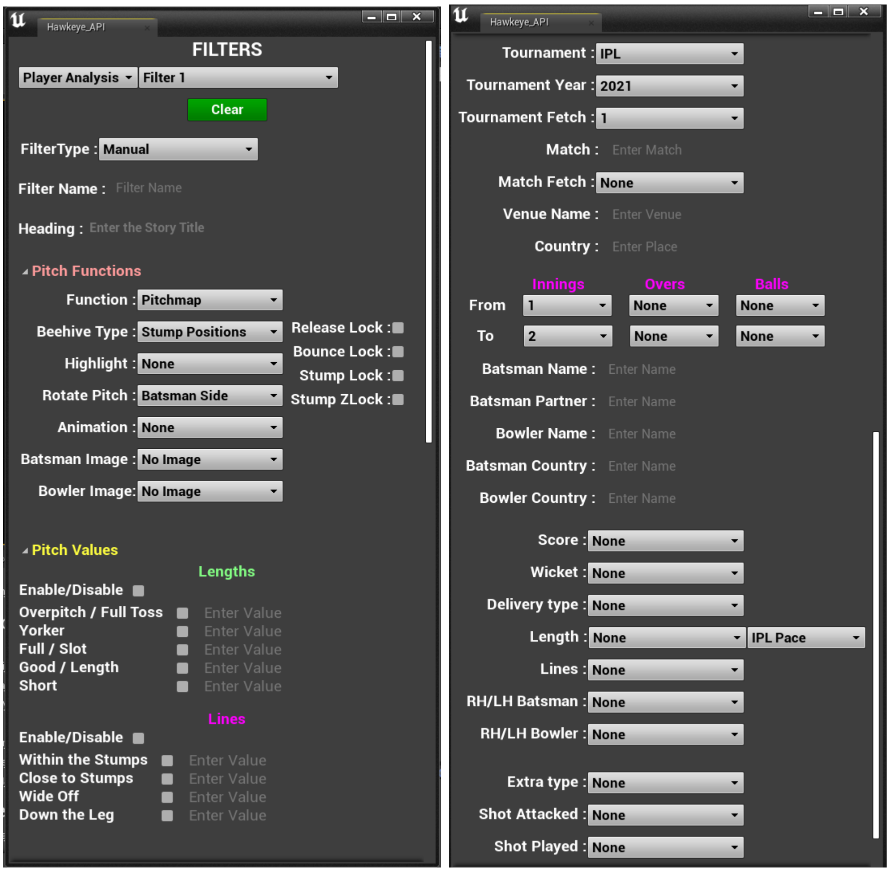

Figure 2. Data filters UI

The data received by Unreal as a response to the API request is then processed and simulated as MR graphics in the form of ball trails, pitch maps etc.

Once the Unreal Engine receives the data from the Hawkeye API, it processes the data as per the graphics that the operator has selected. The operator can choose between four types of graphics: ball trajectories, pitch maps, beehives & wagon wheels (with scoring areas).
The most important part of the Hawkeye graphics is the virtual pitch, which has been created using accurate real-world values. We have created multiple pitches for visualising different types of scenarios i.e. pitch for pacers, pitch for spinners, etc.
If the operator has selected ball trajectories, multiple parameters are extracted from the JSON files for each ball, for which a trajectory is generated between two or three points on the pitch. The number of points used for creating this trajectory depends on the type of ball that is being showcased i.e. full toss, bouncer, etc. Once the trajectory is generated, the Hololens user has the ability to select a point using hand gestures and manipulate it to modify the trajectory as per their requirements.

Based on the inputs of the user, the trajectories of the balls are re-calculated based on the different parameters of the ball from Hawkeye and Unreal Engine's physics based calculations, thus giving the ability to the user to not just visualise the trajectory as per the actual ball information, but to also interact and change the trajectory according to the requirements of the story that they are trying to convey.

4. Evaluation/Key Findings

In this section we identify any roadblocks encountered during development which may inhibit the short and long term progress of the project and present the results of a subjective user test (presenter perspective, audience perspective).

4.1 Roadblocks

Below are some roadblocks and challenges we faced while working on the Mixed reality toolkit:

1. No in-engine support for Azure spatial anchors while using Holographic remoting. The Unreal Engine connected to the Hololens crashes the moment we try to place or load any Azure anchors.
   We have reported this issue to both Epic Games and Microsoft Support, we hope that this issue will be resolved in the future updates of the Hololens plugins.
2. When we use Holographic Remoting with the Windows OpenXR plugin, the Anchors system including Local Anchors or Azure Spatial Anchors do not work in OpenXR. We are currently using the Windows Mixed Reality plugin which supports the Local Anchors.
   Currently the OpenXR plugin is only supported for native Hololens applications, we have reported this issue to Microsoft Support and we hope that this support will also be expanded for applications that use Holographic remoting.
3. We have been using Composure in our project to get an output of the MR graphics that the Hololens user is interacting with. During the play session we have observed that the output stops abruptly and Unreal does not realise this, as the recording buttons/icons in the Composure panel stay active. Hence, we have to manually disable and enable the output using a Blueprint script. We believe that this is a minor bug which would have been fixed in Unreal Engine 5 and we have informed Epic Games of the issue.
4. If we replicate the UXTPessableButton component or if we replicate any of its child components (ex. Static Mesh, Text Render, etc) in a Blueprint, Unreal crashes as soon as that Blueprint is spawned in the Level. As the MRTK plugin is currently in its Beta phase we believe that this issue would be fixed in its future iterations by Microsoft and we have made them aware of the issue.
5. In a scenario where we want to add multiple Hololens HMDs in the ecosystem, especially when the Hololens is connected as a client, the Hololens stops recognising the hands of the user, hence the hand based interactions are not possible for the second Hololens device. In the current system we can add multiple spectator cameras, but only one Hololens which acts as a server. This is a critical issue that we have reported to Epic Games and we are still in communication with them to find a way to solve this problem.
6. The Mo-sys VP Free plugin that we currently use to get the Mo-sys camera tracking information inside Unreal Engine does not provide us accurate focus and lens

distortion data of the camera. We also tried using the Mo-sys VP Pro plugin which provides us accurate data of all the camera parameters, but the format in which the plugin is created hinders us from connecting other devices such as the Hololens in the same environment. We have reported this issue to Mo-sys and they assisted us in creating a system to receive lens distortion data, but we still don't receive accurate focus data. Hence, we have manually set the focus of our virtual cameras at infinity to avoid any issues.

4.2 Subjective user test

Scale

Q1) 1 (very unclear) – 5 (very clear)

Q2) 1 (very unrealistic) – 5 (very realistic)

Q3) 1 (very meaningless) – 5 (very meaningful)

Q4) 1 (very unpleasant) – 5 (very enjoyable)

Presenter Q&A

Question

Q1) Could you please rate the clarity of the information in the scene you just presented?

Q2) How realistic were the graphics in the virtual environment?

Q3) How meaningful were the graphics/interactions in the virtual environment as a storytelling aid?

Q4) How much would you say you enjoyed presenting using the tools provided in the scene?

Audience Q&A

Question

Q1) Could you please rate the clarity of your understanding of the information in the scene you just viewed?

Q2) How realistic were the graphics in the virtual environment?

Q3) How meaningful were the graphics/interactions in the virtual environment as a storytelling aid?

Q4) How much would you say you enjoyed watching the scene?

Audience

PARTICIPANT SN	Q1	Q2	Q3	Q4
1	5.0	4.0	4.0	5.0
2	4.0	3.0	4.0	2.0
3	4.0	3.0	5.0	5.0
4	5.0	4.0	3.0	4.0
5	4.0	3.0	5.0	5.0
6	5.0	4.0	5.0	5.0

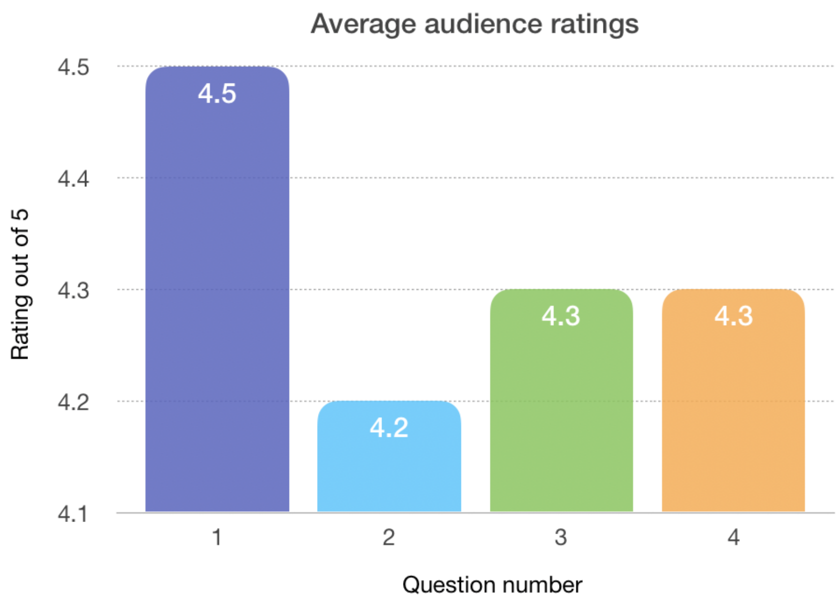

Figure 3. Subjective user test results - audience

Presenters

PARTICIPANT SN	Q1	Q2	Q3	Q4
1	5.0	5.0	5.0	5.0
2	5.0	4.0	5.0	5.0
3	4.5	4.0	4.0	3.0
4	4.0	4.0	3.0	4.0
5	4.0	3.0	5.0	4.0

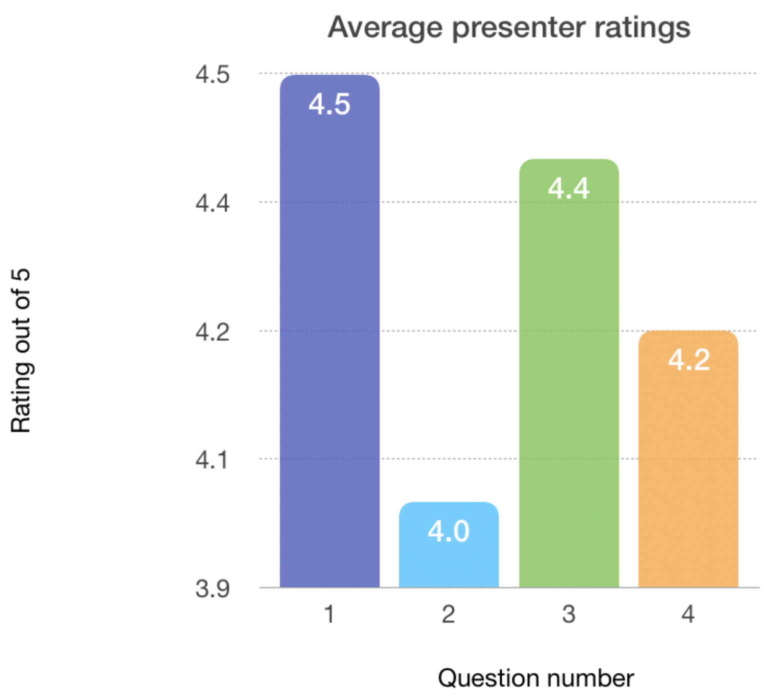

Figure 4. Subjective user test results - presenters

Conclusion & Outlook

Truly interactive graphics are an active area of development for many OEMs and software companies. Broadcast graphics middleware companies built on gaming engines like Zero Density and Pixotope (both based on Unreal Engine), are creating solutions that address talent (presenter) tracking in the 3D space, using either stereo cameras or computer vision. These products also are working on graphics interactions based on presenter body collisions and physics built in to trigger actions in the AR objects. Pixotope recently acquired Trackmen Gmbh, a real-time 3D camera and talent tracking company, to facilitate this goal further and have the provision for multi-person tracking.
These various talent tracking solutions allow for a graphics alpha layer to be switched from foreground to background and visa-versa according to the relative position of the talent to the AR graphic. While our system already has provisions for interactions based on collisions/physics and manual foreground/background switching, automatic switching is an additional functionality that will be added into our system in the near future. The limitations in the commercially available systems however, are that the talent still has no way of directly visualising the graphic that they are interacting with and most systems do not support fine interactions as they are based on a bounding box around the skeleton of the talent - whereas our system is novel in that it facilitates this graphics visualisation feature as well as fine hand/finger interactions.
A further limitation in the above OEM/software company's solution as well as our own, is that a green screen is required for keying the presenter against the graphic when it is placed as a background element. To solve this so that the system will work in this capacity in a hard-set environment, a real time matting/keying solution is proposed. This can be achieved through the use of depth sensing cameras for depth keying, or a more desirable solution of a computer vision based system, which must be robust enough to detect and segment fine features such as hands and fingers. This proposed system will also have value in regular scenes which have AR graphics and virtual sets where no holographic visualisations/interactions are present.
Our system is also currently limited in that it is restricted to the Microsoft Hololens 2 device. In the future we aim to develop functionality for other HMDs and even AR contact lenses.

The results of the subjective user test are promising and demonstrate that our system is successful in aiding in both the storytelling abilities of the presenter and audience experience and understanding of the content produced using the toolkit. As such we see value in developing the solution further and adding in the functionalities described above as and when the technology becomes available.
We can draw a few more detailed conclusions from the subjective user:
The clarity of information for both audience members and presenters scored the highest and realism scored the lowest. We can conclude that the realism of the graphics is something we need to work on as well as developing the ability to add dynamic lighting and shadows and different shaders (currently limited in our system due to software constraints).
Overall both presenter and audience members scored highly in regards to the experience being meaningful and enjoyable, however there is some room for improvement.

One piece of verbal feedback given by several audience participants was that the menu was difficult to see as it was a mirror image in the camera and it would be great to have a visual indication as to what the presenter was selecting. This is something we implemented right away to a positive reception.
In regards to the roadblocks encountered as well as the more general challenges of green screen-free keying/real-time matting, we remain optimistic that these will be solved in due course. The OEMs/software companies have been made aware of the issues we encountered during the project and we continue to work with industry to suggest features, beta test new ones and build new use cases for the technology.
With the release of UE5, OEMs/software companies are updating their plugins and Epic Games has a strong focus on building out the engine for Metaverse applications, so we anticipate they are all incorporating the functionalities required to enable projects such as this.
Our ultimate end goal that we continue to work towards, is for the presenter to be able to see and interact with graphics and virtual sets as naturally as if they were real physical objects and environments, with occlusion solves and physics built into the system, and for this experience to eventually become available for audiences also in the next generation of content, as active participants and not just passive viewers.

References and further reading

Sharon, Brian. “Apollo 11 Mission AR Project Sample Available Now on the Unreal Engine Marketplace - Unreal Engine.” Unreal Engine, Epic Games, 9 Dec. 2019, https://www.unrealengine.com/en-US/blog/apollo-11-mission-ar-project-sample-available-now-on-the-unreal-engine-marketplace. Retrieved on 25th May, 2022.

Epic Games. “Microsoft Hololens Development.” Unreal Engine, Epic Games, https://docs.unrealengine.com/4.27/en-US/SharingAndReleasing/XRDevelopment/AR/ARPlatforms/HoloLens/. Retrieved on 17th June, 2022.

Epic Games. “Developing for Hololens.” Unreal Engine, Epic Games, https://docs.unrealengine.com/5.0/en-US/develping-for-hololens-in-unreal-engine/. Retrieved on 17th June, 2022.

Kerawala, Sean. Sherer, Tim. Summer, Wu. Ferrone, Harrison. Coulter, David. “Unreal Development Overview.” Microsoft, 3 November. 2022, https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unreal/unreal-development-overview?tabs=ue426%2Cmrtk%2Casa%2CD365. Retrieved on 17th June, 2022.

Microsoft. “Mixed Reality UX Tools.” Microsoft, https://microsoft.github.io/MixedReality-UXTools-Unreal/README.html. Retrieved on 17th June, 2022.

Mo-Sys. “Mo-Sys VP Free.” Unreal Engine, Epic Games, 7 January. 2022, https://www.unrealengine.com/marketplace/en-US/product/mo-sys-tracking. Retrieved on 17th June, 2022.

Epic Games. “Announcing a $1Billion Funding Rount to Support Epic’s Long-Term Vision for the Metaverse.” Unreal Engine, Epic Games, 13 April. 2021, https://www.epicgames.com/site/en-US/news/announcing-a-1-billion-funding-round-to-support-epics-long-term-vision-for-the-metaverse. Retrieved on 17th June, 2022.

Epic Games. “The LEGO Group and Epic Team up to Build a PLace for Kids to Play in the Metaverse.” Unreal Engine, Epic Games, 7 April. 2022, https://www.epicgames.com/site/en-US/news/the-lego-group-and-epic-games-team-up-to-build-a-place-for-kids-to-play-in-the-metaverse. Retrieved on 17th June, 2022.

Sullivan, Mark. “Epic Games CEO Tim Sweeney Talks the Metaverse, Crypto, and Antitrust.” Fast Company, 25 April, 2022, https://www.fastcompany.com/90741893/epic-games-ceo-tim-sweeney-talks-the-metaverse-crypto-and-antitrust. Retrieved on 17th June, 2022.

Dot. “Body Pose What?” Pixotope, 29 December. 2021, https://blog.pixotope.com/body-pose-what, Retrieved on 17th June, 2022.

Pixotope. “Trackmen Acquisition FAQ.” Pixotope, 21 April. 2022, https://www.pixotope.com/trackmen. Retrieved on 17th June, 2022.

Izmir, TR. “Zero Density Releases AI-powered Talent Tracking Product.” Zero Density, 16 Novermber. 2021, https://www.zerodensity.tv/zero-density-releases-ai-powered-talent-tacking-product/. Retrieved on 17th June, 2022.

Avid Technology. “Avid Maestro Virtual Set Setup Guide Version 2020.3.” Avid Technology, 2020, https://resources.avid.com/SupportFiles/attach/Maestro_Virtual_Set/Maestro_Virtual_Set_Setup_Guide_v2020.3.pdf. Retrieved on 17th June, 2022.

“Pixotope Aquires TrackMen Real-Time 3D Tracking Solutions.” Business Wire, 21 April. 2022, https://www.businesswire.com/news/home/20220421005180/en/. Retrieved on 17th June, 2022.

Acknowledgements

I would like to thank the Liminal team for their hard work and expertise in helping the Lab team realise this project and co-authoring this paper. I would also like to thank Epic Games, Zero Density, Microsoft and Mo-sys for their technical support and advice regarding the integration of their products into the workflow.
Thanks are also given to my manager Prashant Khanna and leader Sanjog Gupta for supporting the project from its inception. Also thanks to Harshit Baranwal, Rahul Manglik, Rakesh Jha, Pramod Nagdeve, Mahesh Srikanta, Gary Burchett, Harsh Kalan, Saumitra Shankar and Hrushikesh Deo for their inputs and work on refining and testing the product and making it broadcast ready. Also thanks to Aakash Chopra for being our presenter-tester and giving valuable inputs from a talent point of view.
Thanks also to the Disney Star IP, legal teams and PR teams for their work enabling the project patent and whitepaper.