SMPTE

MI

MOTION IMAGING JOURNAL

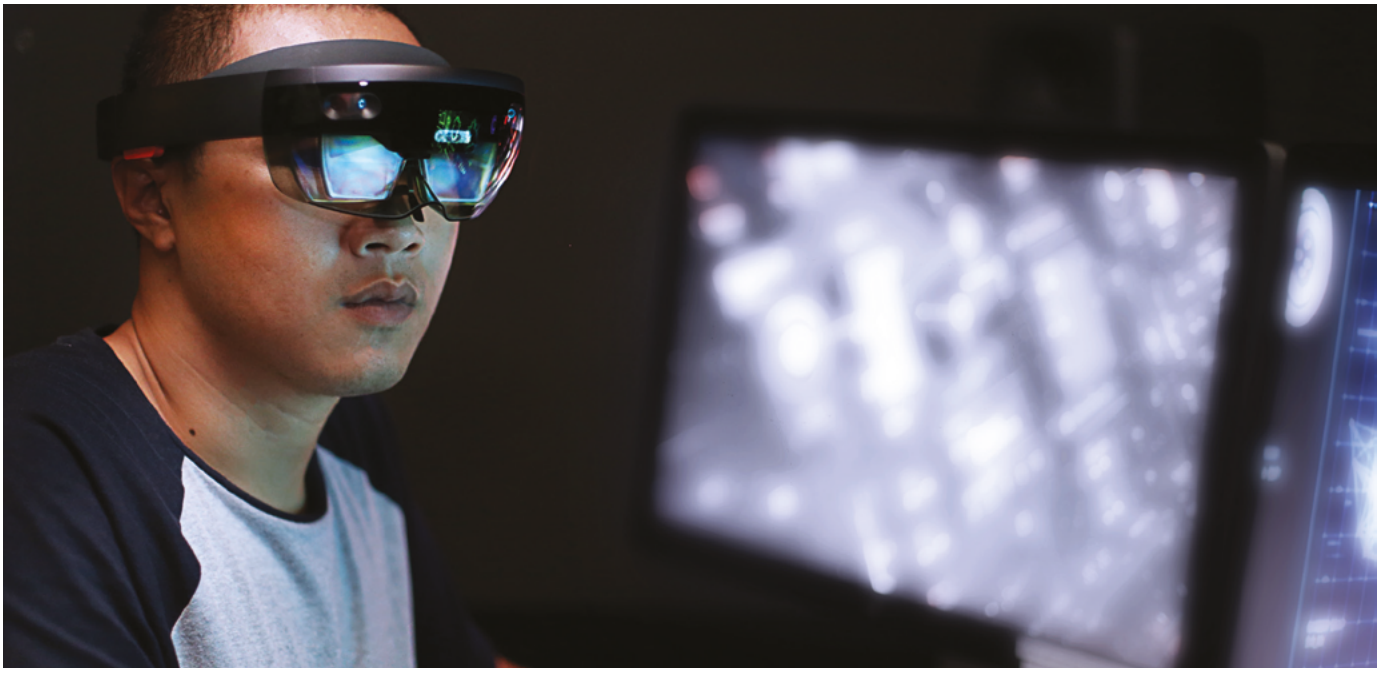
COVERING EMERGING TECHNOLOGIES FOR THE GLOBAL MEDIA COMMUNITY

TECHNICAL PAPER

KEYWORDS MIXED REALITY // UNREAL ENGINE // MICROSOFT HOLOLENS // AR GRAPHICS // VIRTUAL // PRODUCTION

O sistema desenvolvido pela Disney Star's Star Lab representa um avanço significativo na integração de Realidade Aumentada (AR) em transmissões, especialmente para esportes. A combinação de HoloLens, Unreal Engine e dados em tempo real permite interações naturais e *storytelling* aprimorado. Embora desafios técnicos persistam, a adoção de *engines* de jogos e o foco no metaverso indicam um futuro promissor para produções imersivas, isso o que mostra é o próprio teste realizado com usuários que deram notas altas para: Clareza - 4.6 /5 e Experiência- 4.5/5. Embora o realismo gráfico ainda deixe a desejar com 3.8/5. E você que nota daria para a experiência de Realidade Aumentada nas transmissões esportivas ? Responda-me depois de ler esta matéria.

Tom Jones Moreira



Immersive and Interactive AR Graphics and Environments for Broadcast Applications

By Caroline Stedman Mishra, Saurabh Ranjan, and Sanjyot Dahale

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.

Abstract

This paper presents a solution for visualization and interaction with broadcast AR (augmented reality) graphics with a multi-camera spectator camera solution. This solution was developed by Disney Star's Star Lab to provide a more immersive experience for audiences and presenters and aid in storytelling around sports data. It describes the inspiration for the project: the Hololens demos by Epic Games and Microsoft. It describes the technical specifications and workflow of the system. An evaluation details technical roadblocks encountered, steps taken to solve them, and possible improvements to the system. The paper concludes with a summary of the state of the industry, how our product is novel, plans for its next stages of development, as well as an outlook for the future of interactive AR graphics and virtual sets.

Broadcast augmented reality graphics are widely used to demonstrate data points that 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 must rely on comfort monitors that 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 must 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 true of traditional graphics systems like the earlier versions of VizRT[i] (real-time graphics and live production solutions for content creators). While presenter visualization is still not 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 controlling broadcast augmented reality graphics, which allows for real-time dynamic interactions with data within a multi-camera set-up and visualization of AR graphics by a presenter via an HMD (head-mounted device) 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 broadcasts.

The system uses the Microsoft Hololens 2[ii] (an untethered mixed reality headset) as the primary viewing and interaction device via holographic remoting controlled from within Epic Games' Unreal Engine[iii] (a 3D computer graphics game engine), integrated with Mo-sys[iv] camera tracking (virtual production solutions, virtual production training, camera tracking, image robotics and remote production) and Zero Density[v] (integrated solutions for virtual production and real-time motion graphics) real-time graphics rendering.

By utilizing physics functions available in the game engine, we can calculate ball trajectories based on moving data points produced by presenter interactions with matching-Hawkeye[vi] (a Sony company that provides optical tracking, vision-processing, video review, and creative graphic technologies) data, as well as the provision for biomechanics data and bat-ball trajectories from manipulation of 3D player mannequins.

The system also allows for cricket fielding analysis based on player positional data integrated into the system and templates of commonly used fielding configurations. These can be overlaid with the Hawkeye wagon wheel (a graphical representation of the cricket field, with lines showing the trajectories of the scoring balls hit by a batter) and data showing scoring zones for specific players. Additional features include

[i] https://www.vizrt.com/
[ii] https://www.microsoft.com/en-us/hololens
[iii] https://www.unrealengine.com/en-US
[iv] https://www.mo-sys.com/
[v] https://www.zerodensity.io/
[vi] https://www.hawkeyeinnovations.com/

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 to present 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 near future, additional data such as skeletal tracking and pose estimation data will be enabled to create 3D AR replays of sporting moments in near real-time. These dynamic 3D assets will be fully interactive so that the presenter can play, pause, rewind, and manipulate the model to understand the result if the player has 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 continue. While we are currently developing these features for the studio, interactive 3D data visualizations and 3D player replays can eventually be pushed out to audiences via mobile or HMD devices. Volumetric videos of entire sports arenas can become immersive virtual environments where presenters can move around and visualize via an HMD. These same environments can be provided to audiences to experience a match from within, at any angle or position. They also have the potential to be gamified and shaped by audience engagement data and feedback/inputs.

The technology also has further use cases outside sports broadcast and LBE (location-based experiences). As we explore the possible future of a tangible metaverse, we are creating the foundation technology that can be used to build out such future audience experiences, moving out of the television studios and into our audience's living rooms.

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 so that the system used a combination of two 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. A virtual camera was controlled inside Unreal Engine using a combination of these values. The real camera feed was fed into Unreal Engine, and using the Composure plugin, the mixed-reality (MR) graphics that were captured by the virtual camera were composited on top of the real camera feed in real-time. The real camera feed was set as the background in this composition, 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 Hololens 1. Microsoft showcased it in Hololens 2 demos, where the audience could 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 Unity Engine.[vii] We can integrate a secondary camera into the MR space to capture all the MR graphics with which the Hololens user interacts. This feature requires one Hololens device to be mounted on top of the secondary camera. A major limitation of this system was that the camera parameters, like the focal length and 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 to mount an 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 can integrate multiple cameras in the MR space.

The Mixed Reality Toolkit

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

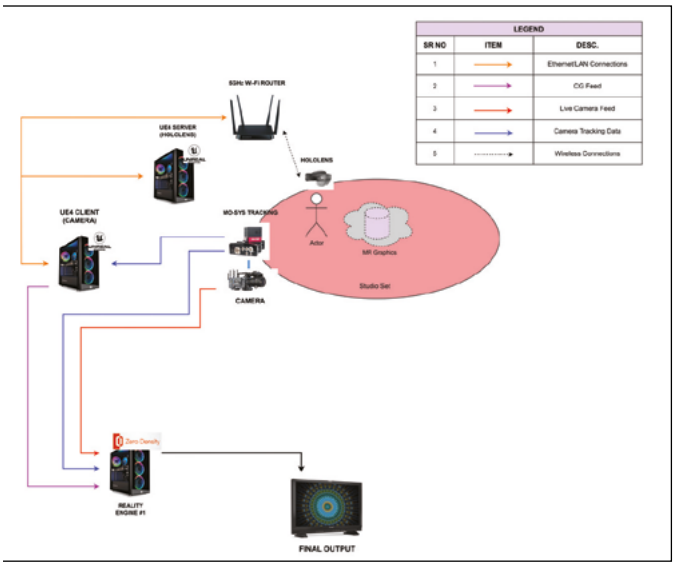
1. Spectator View / Third person view / Production system
2. Hololens operations
3. Pre-show operations & setups using interfaces for Operators.

Spectator View

The foundation of our Unreal Engine project is the multi-player system, through which we can separate the Hololens and the broadcast/spectator cameras as independent entities in the system.

The Hololens, as well as each individual spectator camera to be connected to the system, has its own 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 (**Fig. 1**).

[vii] https://unity.com/



GURE 1. Spectator view workflow.

The computers connected to the cameras receive live camera tracking data from their Mo-sys camera trackers, which provide the camera's positional and pan, tilt, and zoom data, 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 through 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 in two situations: one where the talent is standing on a physical set and the other where the talent is standing on a green screen virtual set. When the talent is standing on a physical set, there is no virtual background, and 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, the zero-density camera captures the virtual background, as the background and the talent captured by the real camera are keyed out from the green screen. Hence, we can 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 we wish to showcase and the position of the talent with respect to the MR graphics, we needed to switch the midground and foreground feeds in 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; in future updates, we plan to automate this switching using technologies like the Zero Density Talent S Traxis, which can provide us with the real-time 3D position of the presenters.

Hololens Operations

When the user wears the Hololens, it scans its surroundings 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 (WMR) plugin[ix] 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, to interact with the MR 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, e.g., menu, video wall, etc., the interaction is received by the server system to which the Hololens is connected (via the WMR plugin), and the result of the action is shared with the client computers connected to 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.

Pre-show Operations and Setups Using Interfaces for Operators

The MR 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 visualize specific analytical simulations.

We collaborated with Hawkeye to gather this data in JSON

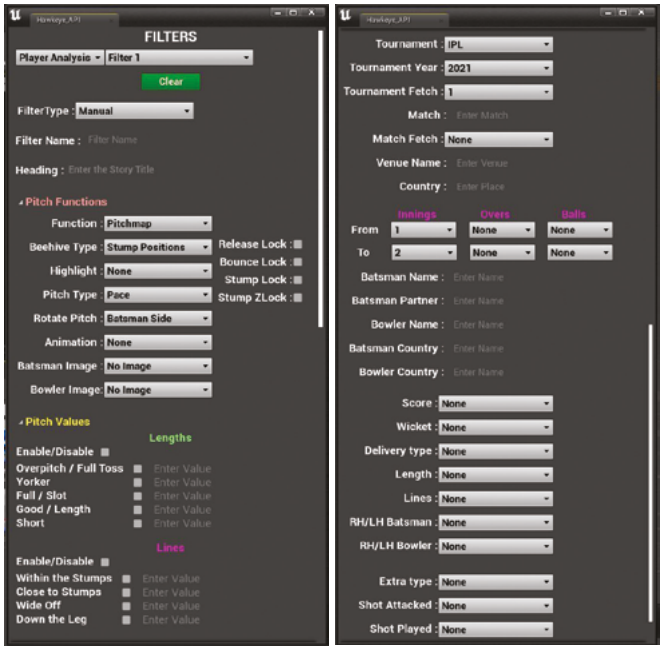


FIGURE 2. Data filters UI.

[viii] https://www.zerodensity.io/platforms/traxis/
[ix] https://github.com/microsoft/MixedRealityToolkit-Unreal

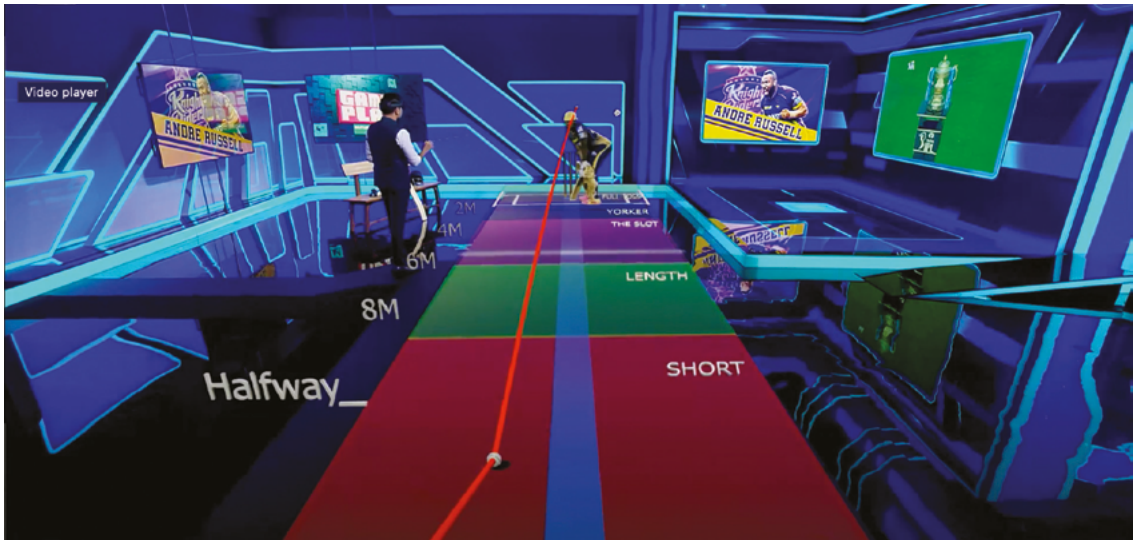


FIGURE 3. Screengrab of subjective user test.

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

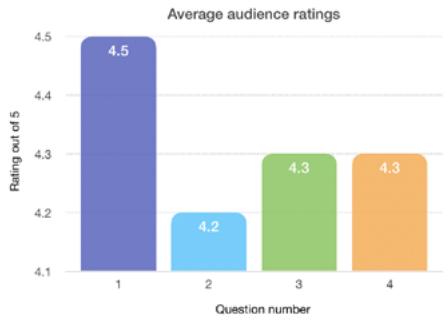


FIGURE 4. Subjective user test results—audience.

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

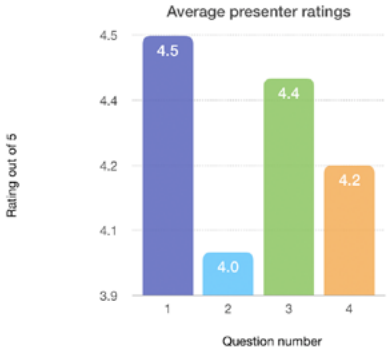


FIGURE 5. Subjective user test results—presenters.

files, which provided us an exhaustive library of parameters regarding each ball played during different matches.

The Hawkeye database combines historical and real-time data. The historical data contains information about each ball delivery since 2018 for all the Indian Premier League Cricket (IPL) tournament matches and several other tournament matches. The real-time database provides live match data with ~ 2-3 sec latency.

We created a custom API to communicate with a remote server that hosts thousands of these JSON files and receives only the data parameters 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 visualization (**Fig. 2**).

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

Once the Unreal Engine receives the data from the Hawkeye API, it processes it according to the graphics the operator selects. The operator can choose between four types of

[x] https://azure.microsoft.com/en-in/products/spatial-anchors
[xi] https://github.com/microsoft/Microsoft-OpenXR-Unreal
[xii] https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/
[xiii] https://software.mo-sys.com/docs/MoSysVPFree/MoSysVPFree_v5.pdf
[xiv] https://www.mo-sys.com/product/mo-sys-vp-pro/

graphics: ball trajectories, pitch maps (diagrams showing where several balls, usually from a particular bowler, have pitched), beehives (diagrams showing where several balls, usually from a particular bowler, have passed the batter), and 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. A cricket pitch measures 20.12m long and 3.05m wide. In the game of cricket, two teams of 11 players take turns batting and bowling.

When one team is batting, they must score as many runs as they can by hitting the ball and then running between two sets of stumps. The opposing team must get them out by bowling the ball overarm at the stumps and hitting them at either end of the cricket pitch or by catching the ball. Once the batting team is all out, they swap over and become the bowling side. Each time a team bats it is known as their innings. Teams can have one or two innings depending on how long there is to play. Whoever scores the most runs wins. But a cricket match can be drawn when the team bowling last is unable to get all the batsmen out.'

We have created multiple pitches for visualizing different types of bowling scenarios, i.e., pitch for pacers, pitch for spinners, etc. The bowling data GUI allows the user to choose bowling deliveries bouncing at a range of distances from the bowler (the "length" from a short delivery designed to bounce near head height to a full toss, which doesn't bounce at all) and with a range of aiming points (the "line" from behind the batter's feet to 1m or more in front of his feet).

If the operator has selected ball trajectories, multiple parameters are extracted from the JSON files for each ball delivery, 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 delivery that is being showcased i.e. full toss, bouncer, etc. Once the trajectory is generated, the Hololens user can 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 ball deliveries are re-calculated based on the different parameters of the ball deliveries from Hawkeye and Unreal Engine's physics-based calculations, thus giving the ability to the user to not just visualize the trajectory as per the actual ball delivery information, but to also interact and change the trajectory according to the requirements of the story that they are trying to convey.

Evaluation/Key Findings

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

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[x] while using Holographic remoting. The Unreal Engine connected to the Hololens crashes when we place or load any Azure anchors. We have reported this issue to Epic Games and Microsoft Support. We hope that it will be resolved in future updates of the Hololens plugins.
2. When we use Holographic Remoting with the Windows OpenXR plugin,[xi] the Anchors system, including Local Anchors or Azure Spatial Anchors, does not work in OpenXR. We currently use the Windows Mixed Reality plugin that 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 observed that the output stops abruptly, and Unreal does not realize this, as the recording buttons/icons in the Composure panel stay active. Hence, we must manually disable and enable the output using a Blueprint script.[xii] We believe this is a minor bug that would have been fixed in Unreal Engine 5, and we informed Epic Games of the issue.
4. If we replicate the UXTPessableButton component or any of its child components (e.g., 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 Microsoft will fix this issue in future iterations, and we have made them aware of it.
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 recognizing the user's hands. Hence, hand-based interactions are not possible for the second Hololens device. We can add multiple spectator cameras in the current system, but only one Hololens, which acts as a server. We have reported this critical issue 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[xiii] that we currently use to get the Mo-sys camera tracking information inside Unreal Engine does not provide us with accurate focus and lens distortion data of the camera. We also tried using the Mo-sys VP Pro plugin,[xiv] which provides us accurate data on 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 reported this issue to Mo-sys, and they assisted us in creating a system to receive lens distortion data, but we still haven't received accurate focus data. Hence, we have manually set the focus of our virtual cameras at infinity to avoid any issues.

Subjective User Test

Five presenters were given the toolkit and instructed to present an analysis segment of 5-10 minutes using the toolkit.

[xv] https://www.pixotope.com/

The presenters then answered a series of subjective questions to ascertain how useful the toolkit is in terms of clarity of information, realism of graphics, and meaningful and enjoyable experience.

The same segment was shown to six audience members, and similar questions asked to gauge the quality of the experience (**Figs. 3-5**).

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?

Conclusion and 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 work 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 further facilitate this goal and provide 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 additional functionality that will be added to our system soon. The limitations in the commercially available systems, however, are that the talent still has no way of directly visualizing 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 visualization 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 to key 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 using 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 that have AR graphics and virtual sets where no holographic visualizations/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 the presenter's storytelling abilities and the audience's experience and understanding of the content produced using the toolkit. As such, we see value in developing the solution further and adding 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 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 regarding the experience being meaningful and enjoyable, however there is some room for improvement.

One 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 of what the presenter was selecting. This is something we implemented right away to a positive reception. Regarding the roadblocks encountered and 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 the 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 will all incorporate 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 graph-

ics 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.

Acknowledgments

I would like to thank the Liminal team for their hard work and expertise in helping the Lab team realize 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 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 input from a talent point of view.

Thanks also to the Disney Star IP, legal, and PR teams for enabling the project patent and whitepaper.

Bibliography and Further Reading

Avid Technology, Accessed 17 Jun. 2022. "Avid Maestro Virtual Set Setup Guide Version 2020.3." Avid Technology, 2020. [Online]. Available: https://resources.avid.com/SupportFiles/attach/Maestro_Virtual_Set/Maestro_Virtual_Set_Setup_Guide_v2020.3.pdf

Business Wire, "Pixotope Acquires TrackMen Real-Time 3D Tracking Solutions," 21 April. 2022. Accessed 17 Jun. 2022. [Online]. Available: https://www.businesswire.com/news/home/20220421005180/en/

Epic Games, "Announcing a $1Billion Funding Round to Support Epic's Long-Term Vision for the Metaverse," Unreal Engine, Epic Games, 13 Apr. 2021. Accessed 17 Jun. 2022. [Online]. Available: https://www.epicgames.com/site/en-US/news/announcing-a-1-billion-funding-round-to-support-epics-long-term-vision-for-themetaverse

Epic Games, "Developing for Hololens," Unreal Engine, Epic Games. Accessed 17 Jun. 2022. [Online]. Available: https://docs.unrealengine.com/5.0/en-US/develping-for-hololens-in-unreal-engine/

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 Apr. 2022. Accessed 17 Jun. 2022. [Online]. Available: https://www.epicgames.com/site/en-US/news/the-lego-groupand- epic-games-team-up-to-build-a-place-for-kids-to-play-in-the-metaverse

Epic Games, "Microsoft Hololens Development," Unreal Engine, Epic Games. Accessed 17 Jun. 2022. [Online]. Available: https://docs.unrealengine.com/4.27/en-US/SharingAndReleasing/XRDevelopment/AR/ARPlatforms/HoloLens/

Izmir, TR, "Zero Density Releases AI-powered Talent Tracking Product," Zero Density, 16 Nov. 2021. Accessed 17 Jun. 2022. [Online]. Available: https://www. zerodensity.tv/zero-density-releases-ai-powered-talent-tacking-product/

Kerawala, Sean; Sherer, Tim; Summer, Wu; Ferrone, Harrison; Coulter, David; "Unreal Development Overview," Microsoft, 3 November 2022. Accessed 17 Jun. 2022. [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unreal/unreal-development-overview?tabs=ue426%2Cmrtk%2Casa-%2CD365

Microsoft, "Mixed Reality UX Tools," Microsoft. Accessed 17 Jun. 2022. [Online]. Available https://microsoft.github.io/MixedReality-UXTools-Unreal/README.html

Mo-Sys, "Mo-Sys VP Free," Unreal Engine, Epic Games, 7 Jan. 2022. Accessed 17 Jun. 2022. [Online]. Available: https://www.unrealengine.com/marketplace/en-US/product/mo-sys-tracking

Pixotape, "Body Pose What?" Pixotope, 29 Dec. 2021. Accessed 17 Jun. 2022. [Online]. Available: https://blog.pixotope.com/body-pose-what

Pixotope, "Trackmen Acquisition FAQ," Pixotope, 21 Apr. 2022. [Online]. Available: https://www.pixotope.com/trackmen

Sport Cricket, "The aim of Cricket," Accessed 16 May 2024. [Online]. Available: http://news.bbc.co.uk/sport2/hi/cricket/rules_and_equipment/4183172.stm

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

Sullivan, Mark, "Epic Games CEO Tim Sweeney Talks the Metaverse, Crypto, and Antitrust." Fast Company, 25 Apr. 2022. Accessed 17 Jun. 2022. [Online]. Available: https://www.fastcompany.com/90741893/epic-games-ceo-tim-sweeney-talks-themetaverse- crypto-and-antitrust

About the Authors

Caroline Mishra is a creative technologist and new media artist based in India.She started out in live sound and went on to work in event creative, technical production, and innovation. She joined Disney Star in 2019 to head R&D and now consults on emerging technology in entertainment.

Saurabh Ranjan is a creative technologist, entrepreneur, and product builder dedicated to blending reality and imagination through art, technology, and storytelling. Ranjan founded Liminal in 2015 and has spent seven years developing new products.

Sanjyot Dahale is a programmer and developer by passion. He has more than five years of experience in XR development, virtual production, and live broadcasting. At Liminal, he heads the Tech and R&D division as a technical solutions architect, implementing creative concepts to create immersive experiences.

This paper first appeared in the Proceedings of the NAB 2023 Broadcast Engineering and Information Technology (BEIT) Conference. It is reprinted here with permission of the author(s) and the National Association of Broadcasters, Washington, D.C.

DOI: 10.5594/JMI.2024/IHPC9661
Date of publication: 15 July 2024