EPiC Series in Computing

Volume 107, 2025, Pages 354–363

Proceedings of EUNIS 2025 annual congress in Belfast

Innovating Educational Technology: A Beta Implementation of Hybrid Production Environments in the AUTh Studio

Evangelos Christopoulos[1], Georgios Roussos[*], Angeliki Agorogianni[1]

[1] Digital Governance Unit (DGU) - Aristotle University of Thessaloniki (AUTh), Greece

christopec@it.auth.gr, grou@it.auth.gr, aagorogi@it.auth.gr

Abstract

The paper presents a pilot implementation of hybrid production environments at the Aristotle University of Thessaloniki (AUTh) studio, using Unreal Engine (UE) 5 and Vizrt NDI technology. The project focuses mainly on synchronizing physical and virtual environments to improve educational content creation. The main areas of development of the entire research and implementation include camera synchronization, depth composition and virtual reality integration, aiming to improve interactive learning experiences. Furthermore, taking into account ensuring compliance with GDPR and the protection of personal data, the project presents ideas and implementations for anonymous depth-based visualizations for interviews and voice recordings, using silhouette representations instead of real face or body images. This approach offers privacy-focused solutions for disciplines dealing with sensitive data, such as psychology and social research. Finally, the project highlights interdisciplinary applications in healthcare, engineering, and media studies, demonstrating the potential for simulations and hands-on training. The findings contribute to advancing educational studio technologies, driving innovation in audiovisual content production, and providing a model for integrating emerging technologies into academic environments. This beta application lays the groundwork for future development, emphasizing scalability, privacy, and interdisciplinary collaboration.

[*] https://orcid.org/0000-0003-2311-5196

L. Desnos, R. Vogl, L. Merakos, C. Diaz, J. Mincer-Daszkiewicz and S. Mclellan (eds.), EUNIS 2025 (EPiC Series in Computing, vol. 107), pp. 354–363

1 Introduction

We live in a digital age where universities must embrace technology to improve education and research. As a university, we are focused on building a strong digital ecosystem that supports innovation and new teaching methods (Roussos et al., 2025). Work has shown that AR, VR and XR are shaping the future of education, offering new ways for students to engage with educational materials, even in primary education (Roussos et al., 2022). Digital leadership plays a key role in this transformation, helping universities adapt and effectively implement these developments (Brown et al., 2024; Roussos et al., 2025)

In recent years, virtual production and virtual reality have rapidly entered all forms of audiovisual content development, from cinema to education (Zhang & & Li, 2023; Wilson & & Clark, 2024). Studios equipped with such tools provide new opportunities for teaching and applied learning (Cremona & Kavakli, 2023). Recognizing these growing needs, we have explored methods for integrating these technologies into our university studio.

1.1 The Background

The rapid advancement of digital technology has transformed educational practices, enabling more immersive and interactive learning experiences. Virtual production, once limited to high-budget film and media industries, is now making its way into academic environments, providing new tools for teaching and research. Technologies such as Unreal Engine 5 (UE5), Vizrt NDI, and PTZ cameras allow for the seamless integration of real and virtual spaces, offering dynamic content creation possibilities. Universities worldwide are exploring these innovations to enhance remote learning, support interdisciplinary collaboration, and create scalable, high-quality educational content (Garcia et al., 2023; Chen & & Wang, 2024; Lee & & Kim, 2024).

In fact, virtual production tools and hybrid studio environments have expanded significantly in educational settings, driven by the need for more interactive and immersive learning experiences. This paper explores how integrating UE5 with Vizrt NDI enhances educational studio production at AUTh, allowing seamless interaction between real and virtual environments (Cremona & Kavakli, 2023; Anderson, 2023; Chen & & Wang, 2024)

1.2 Scope, Objectives and Limitations

In order to improve the creation of instructional material, the project focuses on establishing and beta testing a hybrid production studio at AUTh that combines virtual and real-world settings. The studio offers an adaptable and scalable digital environment for interactive education, research, and media creation by utilizing technologies including UE5, Vizrt NDI, PTZ cameras, and real-time depth estimation (Smith & & Jones, 2024; Garcia et al., 2023). This program enables educators and students to experiment with innovative teaching strategies and imaginative narrative tactics by bridging the gap between contemporary virtual production tools and conventional audiovisual education. The hybrid method improves the quality of digital material and remote and hybrid learning settings, facilitating multidisciplinary cooperation in sectors including digital arts, media studies, engineering, and healthcare.

Despite its potential, the project faces several technical and operational challenges that must be addressed for broader adoption. Previous research on educational studio infrastructure at AUTh has identified key constraints, including spatial limitations, technical integration issues, and the need for specialized training (Charidimou et al., 2025). One of the main challenges is overcoming latency in PTZ synchronization and inconsistencies in real-time depth estimation, which impact the overall quality of production. Additionally, scalability is another concern, as the current setup is a beta version that requires further testing and refinement before broader deployment. Furthermore, ensuring GDPR compliance and ethical considerations in using depth-based anonymization techniques for research purposes is essential.

2 Studio Infrastructure and Technology Overview

The research we conducted was based upon the pre-existing infrastructure of an educational studio in our university (Charidimou et al., 2025).

2.1 Studio Layout and Key Equipment

The studio comprises of two spaces: the production room, a 4x4m space where the capture takes place, and the control room (Charidimou et al., 2025). For the purposes of this paper, it is important to note the following features of the studio:

- **NDI & Tricaster Mini**: The audiovisual infrastructure of the studio is developed around the NewTek (Vizrt) Tricaster media production suite. This means all devices communicate via the NDI protocol over the local network.
- **PTZ cameras**: Two Vizrt PTZ (pan, tilt and zoom) cameras are the main infrastructure. They are controlled over NDI via the Tricaster.
- **Green Screen**: A green wall is available for keying.

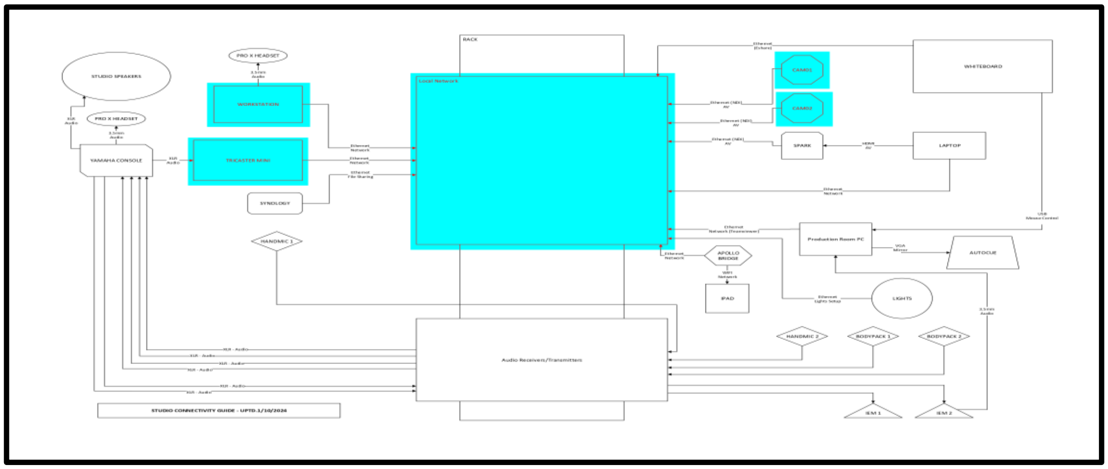

Figure 1: Connectivity diagram of the studio. Highlighted the components utilized in this study.

2.2 The Role of NDI and the Integration with UE5

NDI has been quickly gaining in popularity since its release by NewTek (now Vizrt) in 2015, especially during the Covid-19 pandemic, as a low-cost solution for small-scale production systems . The Proliferation of NDI in Live Production and Broadcast Workflows. This makes the protocol an exciting area of research that applies to smaller enterprises and individuals instead of being limited to large-scale productions. Since the release of the latest version UE5, it has helped boost virtual production capabilities for both professional and amateur applications (Cremona & Kavakli, 2023).

3 Building the Mixed - Hybrid Environment: Methods and Implementation

The main challenge when creating the mixed world was compositing the real and digital environments in a seamless and realistic way. This meant transferring several geometrical and camera parameters between the real scene and the digital environment.

3.1 Creating a Virtual Model of the Studio

During the research process, it became clear that we would need an accurate 3d replica of the geometry of the studio. The two dominant low-cost options for creating such models at this point are photogrammetry and LiDAR scanning. Photogrammetry requires capturing hundreds of RGB images of the object which are then processed to create the 3d representation. LiDAR on the other hand uses a LiDAR (Light Detection and Ranging) sensor which captures depth information alongside a regular RGB camera.

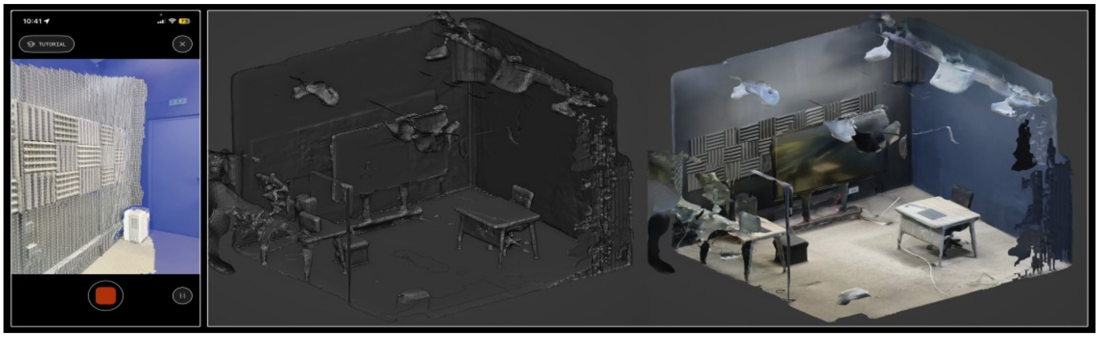

Figure 2: PolyCam user interface (left), 3d studio model (right)

LiDAR systems became widely available with the release of the iPhone 12 Pro which featured a built-in LiDAR sensor. Although relatively low-resolution, the sensor has great potential for room scanning applications. Utilizing other built-in sensors of the iPhone, apps like PolyCam now offer a user-friendly and fast approach to 3d scanning (Askar & Sternberg, 2023). Using Polycam's LiDAR mode on an iPhone 12 Pro device, we were able to capture an accurate 3d model of the studio in exact world units, as seen below.

3.2 Camera Synchronization and PTZ Calibration

A key aspect of integrating digital and physical scenes involves accurately aligning virtual and physical cameras in terms of position, orientation, and field of view. In UE5, the static placement of physical cameras was replicated using a 3D studio scan. Challenges with orientation and zoom arose due to the limitations of NDI's PTZ state query, necessitating a coordinated initialization and synchronized commands through Tricaster macros. Mechanical acceleration discrepancies in the physical cameras required adjustments to these macros. Although these issues are minor for static educational recordings, future enhancements could leverage a VIVE Tracker to achieve precise camera orientation, facilitating improved PTZ integration.

3.3 Depth Estimation and Compositing

The real-world scene and the digitally created environment are two separate plates. In order to correctly mix them together, we need data regarding the "depth" of both scenes. By depth we mean the distance from each pixel to the camera. This information can be represented in a grayscale image map. A brighter value signifies a pixel farther away from the camera, while a darker value signifies a pixel closer to the camera. This means that by comparing the corresponding values in each map, we can decide whether the digital or the real plate is closer to the camera for every pixel. This method is known as "deep compositing." When it comes to the digital environment, UE5 offers built-in tools and post-processing materials for calculating the exact pixel depth of the digital environment. This information can be streamed through NDI alongside the virtual environment.

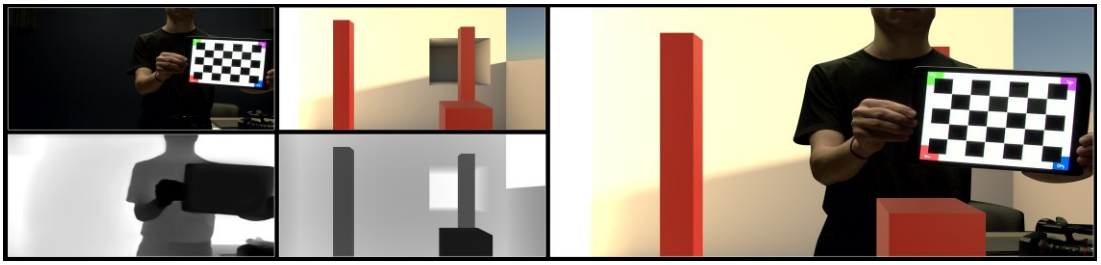

Figure 3: Deep compositing: The real scene and the estimated depth (left), the digital environment and its pixel depth as extracted from the renderer (middle) and the composited image (right)

3.3.1 Estimating the Depth of the Real-world Scene

Estimating the depth of the real scene can be achieved using various algorithms, machine learning models or specialized hardware like following.

Stereoscopic Depth Estimation: Estimates depth using data captured from two side-by-side RGB cameras, similarly to how human eyes perceive depth (Hirschmuller, 2007).

Monocular Depth Estimation: Machine learning algorithms can be utilized to estimate depth from a single camera input. Though inaccurate and probably unfit for the final project, the ease of use of this approach allowed us to experiment thoroughly with it (Ranftl et al., 2020; Yang et al., 2019).

Dedicated Depth Estimation Hardware: Several manufacturers offer specialized depth cameras that work using built-in stereoscopic cameras, ToF (Time of Flight) sensors etc. These solutions offer real-time, almost-perfect depth calculation without any computational cost.

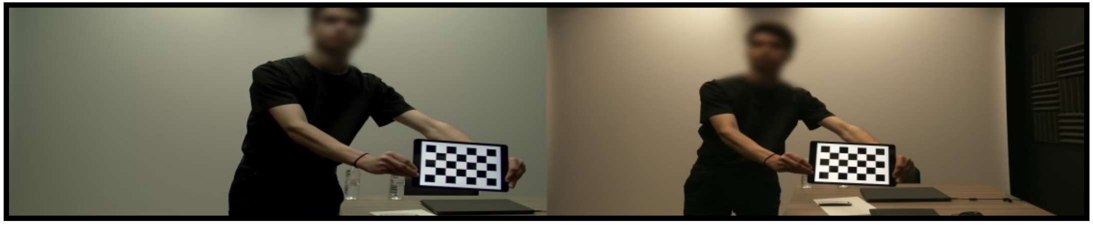

Figure 4: Camera Calibration for algorithmic methods (SGBM)

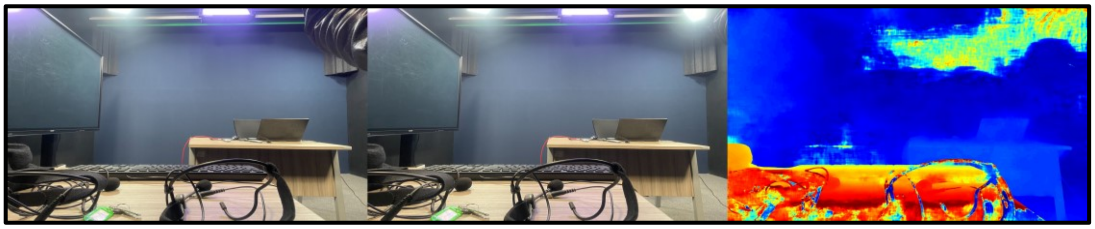

Figure 5: Draft test using the Middlebury-v3 dataset

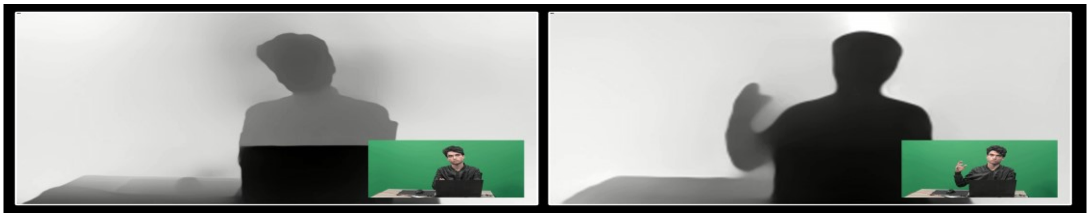

Figure 6: Experimenting with MiDaS: Large model (left), accurate but slow. Small model (right): Real-time, but inaccurate and temporally unstable

3.3.2 Integrating the Green Screen

In addition to depth estimation, the greenscreen can be utilized to improve edge detail.

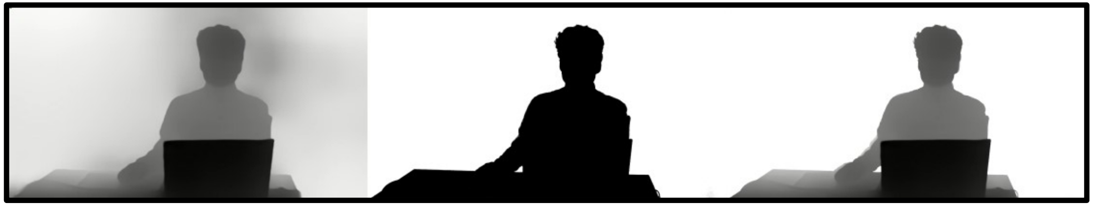

Figure 7: MiDaS large model estimation (left), green screen garbage matte (middle), combined result with refined edges (right)

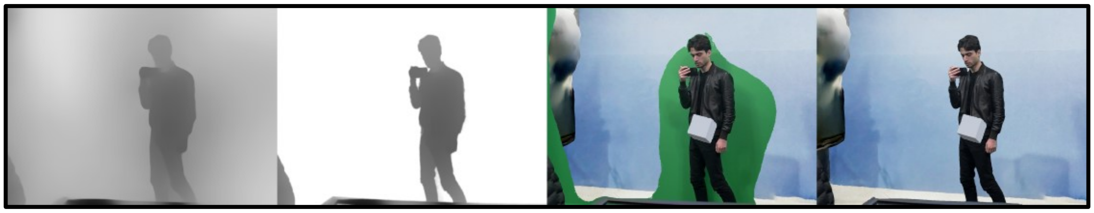

Figure 8: The greenscreen prevents the physical wall from showing in the final composite

Most depth estimation techniques, especially when used in real time, provide low-resolution results which lead to low edge detail. By using chroma keying we generated a "garbage matte" map to optimize edge detail. As the greenscreen can be considered of infinite depth, it can also help prevent the back wall from clipping through, as shown in this example from our demo using MiDaS.

3.4 Depth Estimation in a Real Time Pipeline

An interesting challenge here is to see which depth estimation methods are best suited for real time compositing. Besides allowing both the presenter and the production team to visualize the final result on the spot, real time depth estimation and compositing would offer advantages like reduced file sizes and lessen the work during the editing stage.

3.4.1 Offline Pipeline

First let's look at this diagram representing software-based, offline depth estimation. The system works as it would without the depth estimation, so the latency is that of X, where X is the lag introduced by an NDI video stream.

We will use this as a baseline to compare the performance of real-time options. Please note that three video files need to be stored and edited (if using stereoscopic methods with 2 cameras, 4 files would be needed). This adds to the complexity of the editing process.

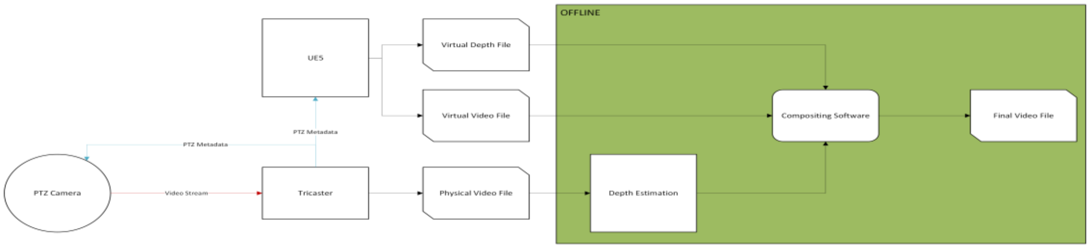

Figure 9: Diagram of offline depth estimation (latency X). The audiovisual streams (over NDI) are shown in red, as are all computational processes.

3.4.2 Real-Time Pipeline using Depth Cameras

As for the real-time options, we start with the depth camera alternative. As is shown below, the depth video feed is parallel to the video stream, meaning it does not add extra complexity to the system.

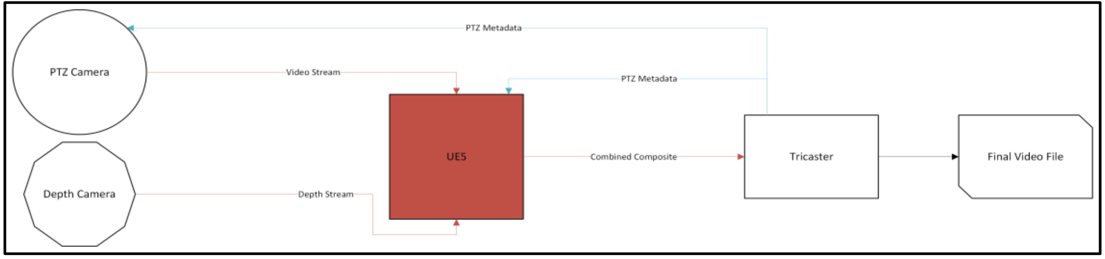

Figure 10: Real time depth estimation pipeline using a depth camera (latency 2X+K). Processing steps that slow the system down are shown as red squares.

The two feeds are processed together in UE5, which introduces lag K. The final composite is transferred over another NDI stream to Tricaster, meaning the total latency is 2X + K. It is obvious that the latency introduced is the bare minimum for the goal we are trying to achieve. By making sure to minimize the "K" component by optimizing the rendering process inside UE5, the latency comes down to miliseconds, and is not noticeable on set nor does it negatively impact the quality of the final composite.

3.4.3 Real-Time Pipeline using Software Tools

All methods requiring calculations (either algorithmic or AI-assisted) follow the same pipeline as shown below.

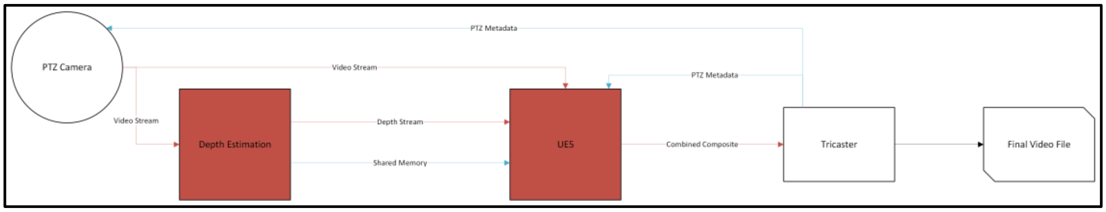

Figure 11: Real time implementation of approaches involving software. Latency of 2X+K+D for shared memory implementations, 3X+K+D for streaming depth over NDI

After depth estimation, there are two options to transfer the depth map to UE5: via an NDI stream (like a depth camera) or using shared memory. The shared memory method requires both UE5 and the depth estimation script to run on the same machine, allowing direct RAM access (e.g., via Spout), with minimal lag. However, real-time depth estimation is resource-intensive, making this setup likely to slow both processes.

Running depth estimation on a separate machine avoids performance issues but requires sending the depth stream to UE5 via NDI, adding latency (variable X). In the system diagram, the video stream reaches UE5 and the depth process simultaneously (latency X); depth estimation adds D, and transfer to UE5 adds another X, resulting in 2X+D. Therefore, total system latency becomes 3X+D+K.
Critically UE5 receives the depth stream with a delay of X+D relative to the video, causing desynchronization. Since this latency depends on depth estimation speed, without a complex timecode system, visual alignment may fail especially with presenter motion. The only fix is faster depth estimation to reduce D, but this risks lowering quality and accuracy.

3.4.1 Implementing a Real-Time Monocular Depth Estimation Pipeline

As shown above, software-based real-time depth estimation is not ideal for minimizing latency or ensuring high-quality composites but can serve for low-quality visualization or offline use. So, we primarily tested the MiDaS monocular AI model, chosen for its scalability and single-camera setup. However, even at high settings, its depth estimation lacks the robustness needed beyond testing. We used also the third pipeline option: the NDI (not shared memory) to transfer depth data from a separate machine running the estimation script. Both machines were connected via high-speed Ethernet to reduce network lag. Most depth estimation methods have Python implementations. Vizrt's Python wrapper for NDI (tested with Python 3.10) led us to use Python within Visual Studio Code for this setup. On the NDI side, we worked with four basic functions:

```
def displaySources(): # lists all NDI sources in the network
def openReceiver(sourceIndex): # creates a receiver for the selected source
def receive(): # returns every new frame received from the source
def broadcast(frame): # broadcasts frame in an NDI stream
```

Utilizing basic functions from the MiDaS library and the NDI functions, we created a command to read and process an NDI stream and create a real time stream with the estimated depth result.

```
def livePredictNDI(self, input, scale):
        ndiObj = NDI()
        ndiObj.displaySources()# display NDI sources
        ndiObj.openReceiver(input)# open desired source

        ndi_send = ndi.send_create()# create an NDI sender for broadcasting
        video_frame = ndi.VideoFrameV2()# frame readable by NDI
        while True:
            t, v, a, m = ndi.recv_capture_v2(ndiObj.receiverInstance, 300) # timeout for receiving data
            if t==ndi.FRAME_TYPE_VIDEO: # separate image data from audio and metadata
                    frame_data = np.frombuffer(v.data, dtype=np.uint8)
                    frame = frame_data.reshape((v.yres, v.xres, 2))
                    frame = cv.cvtColor(frame, cv.COLOR_YUV2BGR_UYVY) # format received frame to readable color
                height, width = frame.shape[:2]
                frame=cv.resize(frame, (int(width * scale), int(height * scale))) # downscale frame
                depthMap = self.predict(frame) # estimate depth using MiDaS

            video_frame.data = cv.cvtColor(depthMap, cv.COLOR_GRAY2BGRA)
            video_frame.FourCC = ndi.FOURCC_VIDEO_TYPE_BGRA # Ensure correct format for broadcasting
            ndi.send_send_video_v2(ndi_send, video_frame) # broadcast the depth information
```

Real-time results (see 3.3.1) showed slight mismatches between depth and RGB images, even with the fastest model and aggressive downsampling to reduce latency ("D"), due to the two NDI streams between the camera, depth estimation machine, and UE5. Offline tests used a modified script and the first pipeline (see diagram), with full resolution and highest quality settings. This greatly improved depth prediction, but temporal instability between frames caused flickering, making it unsuitable for production (see 3.3.2 and 3.5). This flickering is a known limitation of monocular models. Still, the method's low cost and simplicity suggest potential. For now, depth cameras remain the most reliable option.

3.5 Virtual Reality Mode and Presenter Interaction

Everything we have explained until this point relates to the point of view of the camera. However, the presenter also needs a way to perceive their digital surroundings. For this reason, we tested Unreal Engine's VCAM application, which allows the presenter to view the digital world through their phone.

This method both allows the presenter to interact with the digital world and gives us their point of view which can then be streamed to Tricaster via NDI. The application also runs on Apple Vision glasses for full VR. Other types of VR glasses can also work with UE5 with methods other than the VCAM app.

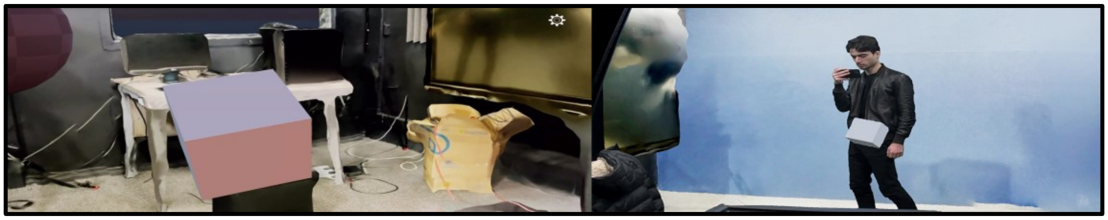

Figure 12: The VR perspective in the MiDaS demo

4 Results and Possible Use Cases

Summarizing the paper and mentioning the results and possible use cases, we consider that the beta implementation of hybrid production environments at AUTh Studio is a first research step and, of course, explores possible additional innovative applications in educational technology. For instance, virtual classrooms, interactive learning materials, and voice simulation enhance content creation while ensuring anonymity through depth mapping, replacing real faces with silhouette-based visuals to comply with GDPR. This is crucial for sensitive fields like psychology, legal studies, and social research. In media and film studies, virtual production serves as a training ground for students and supports film and broadcasting projects. Research in mixed reality XR focuses on VR and AR experiments, depth estimation, and cultural or historical reconstructions. Training and simulation applications extend to crisis response and emergency preparedness, using controlled environments for skill development.

5 Discussion and Future Work

The preliminary research is nearly complete, and the next step is to finalize a standardized workflow. At Aristotle University, we are committed to continuously enhancing our processes to achieve better outcomes (Roussos et al., 2025). This ongoing effort reflects our dedication to excellence and innovation. Key decisions include selecting a depth estimation method (hardware-based for realism or MiDaS for anonymity), integrating VR and PTZ tracking for immersive and accurate interaction, and developing a low-latency UE5 pipeline. Once the system is robust, outreach to university departments and exploration of cross-disciplinary applications will follow.

References

Anderson, J. (2023). Teaching Virtual Production: The Challenges of Developing a Formal Curriculum. *Film Education Journal*. https://doi.org/10.14324/FEJ.03.1.06

Askar, C., & Sternberg, H. (2023). Use of smartphone lidar technology for low-cost 3D building documentation with iPhone 13 pro: a comparative analysis of mobile scanning applications. *Geomatics, 3*, 563–579. https://doi.org/10.3390/geomatics3040030

Brown, T., Green, P., & & White, S. (2024). A Comparative Analysis of Virtual Education Technology, E-Learning Research, and the Digital Divide in Global South Countries. *Informatics*. https://doi.org/10.3390/informatics11030053

Charidimou, D., Roussos, G., Agorogianni, A., Patsinakidou, A., Nikolaidis, S., & Christopoulos, E. (2025). An educational studio for quality multimedia content-A successful delivery of IT practices. *Proceedings of EUNIS*, *105*, pp. 326–335. https://doi.org/10.29007/wf33

Chen, L., & & Wang, M. (2024). Design of an Optical Physics Virtual Simulation System Based on Unreal Engine 5. *Applied Sciences*. https://doi.org/10.3390/app14030955

Cremona, C., & Kavakli, M. (2023). The evolution of the virtual production studio as a game changer in filmmaking. In *Creating digitally: Shifting boundaries: Arts and technologies—Contemporary applications and concepts* (pp. 403–429). Springer. https://doi.org/10.1007/978-3-031-31360-8_14

Garcia, M., Patel, A., & & Nguyen, L. (2023). Experimental Results on Synthetic Data Generation in Unreal Engine 5 for Real-World Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* https://doi.org/10.1109/CVPR.2023.00123

Hirschmuller, H. (2007). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence, 30*, 328–341. https://doi.org/10.1109/tpami.2007.1166

Lee, H., & & Kim, J. (2024). Research on Virtual Education, Inclusion, and Diversity: A Systematic Review. *International Journal of Educational Technology in Higher Education*.

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2020). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence, 44*, 1623–1637. https://doi.org/10.1109/tpami.2020.3019967

Roussos, G., Agorogianni, A., Salmatzidis, I., Ferrell, G., & Kähkipuro, P. (2025). 2024 and Beyond: Navigating the Digital Shift-Leadership Strategies for the Future of HEIs in Europe. *Proceedings of EUNIS*, *105*, pp. 265–275. https://doi.org/10.29007/t67l

Roussos, G., Aliprantis, J., Alexandridis, G., & Caridakis, G. (2022). Augmented Reality in Primary Education: Adopting the new normal in learning by easily using AR-based Android applications. *Proceedings of the 26th Pan-Hellenic Conference on Informatics*, (pp. 347–354). https://doi.org/10.1145/3575879.3576016

Smith, L., & & Jones, M. (2024). Digital Education: Mapping the Landscape of Virtual Teaching in Higher Education. *Education and Information Technologies*. https://doi.org/10.1007/s10639-024-12899-2

Wilson, R., & & Clark, D. (2024). How Personalized and Effective Is Immersive Virtual Reality in Education? A Systematic Literature Review for the Last Decade. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-023-15986-7

Yang, G., Manela, J., Happold, M., & Ramanan, D. (2019). Hierarchical deep stereo matching on high-resolution images. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 5515–5524). https://doi.org/10.1109/cvpr.2019.00566

Zhang, Y., & & Li, X. (2023). Artificial Intelligence Pathfinding Based on Unreal Engine 5 Hexagonal Grid Map. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.10498463