IEEE INFOCOM 2025 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 979-8-3315-4370-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOMWKSHPS65812.2025.11152969

Enabling Generative AI based Multi-sensory XR Applications with Mobile Edge Computing

Peng Qian*, Carlos Velez Redondo†, Ning Wang‡, Carl Udora†, JingXuan Men†, Rahim Tafazolli†
*Department of Engineering Science, University of Oxford, U.K.
†5GIC & 6GIC, Institute for Communication Systems, University of Surrey, U.K.
‡School of Electrical, Electronic and Mechanical Engineering, University of Bristol, U.K.
Emails: *peng.qian@eng.ox.ac.uk, †{cv00218, cu00029, j.men, r.tafazolli}@surrey.ac.uk, ‡n.wang@bristol.ac.uk

***Abstract*—With the rapid development of XR devices, holographic applications are expanding across various domains. However, it is a consensus that capturing and transmitting real-time holographic content still requires significant bandwidth. Even with the enhanced wireless capabilities of mobile systems, they still fall short of meeting the bandwidth and latency demands required for near-Gigabit per second interactive application scenarios. This paper proposes a network architecture that leverages MEC to address these challenges with the assistant of Generative AI. In this framework, the MEC server can leverage the power of the generative AI model to generate holographic objects with the input of user semantic commands, instead of requiring end-users to capture and transmit large raw holographic data. This approach significantly reduces uplink bandwidth requirements while enabling efficient real-time content generation. To validate this approach, we design an interactive and multisensory operational training scenario relying solely on semantic uplink transmissions from the end-users. The preliminary results based on the testbed implemented highlight the feasibility of deploying diverse holographic applications in wireless environments.**

***Index Terms*—Holographic Application, extended reality (XR), mobile edge computing, generative AI**

I. Introduction

In recent years, innovations in extended reality (XR) devices have spurred extensive research aimed at enhancing the user experience in holographic applications. Recent innovations have led to the proliferation of holographic video applications in domains such as remote healthcare, gaming, education, and industrial operations. Holographic technologies, with their ability to create highly immersive and realistic experiences, are poised to transform the way we interact with digital content. At the same time, the rapid expansion of 5G networks offers an opportunity to address these issues by providing lower latency, higher bandwidth, and enhanced network capabilities [5], [13]. These advances are crucial for enabling the transmission of real-time holographic content, which forms the foundation for practical and deployable metaverse applications [1], [19], [4].

Despite these advancements, the integration of holographic applications into mobile networks remains a challenging task. A critical challenge lies in designing a system capable of efficiently handling the capture, transmission, and rendering of holographic content in real time. The sheer size of raw holographic data, often consisting of millions of points in a point cloud, creates impractical demands on uplink bandwidth. In addition, the inherent instability of wireless networks exacerbates latency and packet loss, which can severely degrade the quality of user experiences. Current methods relying on the direct transmission of raw holographic data struggle to meet the dual demands of maintaining high perceptual quality and achieving real-time responsiveness [13].

To address these challenges, we propose a novel network architecture that leverages the computational power of Mobile Edge Computing (MEC) servers in conjunction with advanced generative AI models. By strategically deploying these generative AI models at the network edge, MEC servers can efficiently synthesise high-quality holographic content directly from compact semantic representations. This innovative approach significantly reduces uplink bandwidth requirements, thereby alleviating the burden on wireless networks. Moreover, it ensures ultra-low-latency holographic streaming by offloading resource-intensive tasks such as calibration, compression, and rendering to the edge, where they can be processed in real-time. On the downlink, apart from the necessary transmission of the initially generated point cloud data, the MEC server prioritises sending only the updated object coordinates and orientation changes whenever possible. This intelligent design eliminates the need for repeatedly transmitting large volumes of real-time scene data, thereby conserving bandwidth and enhancing overall system efficiency.

To demonstrate the practicality of our solution, we designed an interactive operational training scenario as a use case. In this scenario, trainees are required to real-time recognise and manipulate multiple objects generated based on the trainer's commands, leveraging a variety of sensory feedback mechanisms. Specifically, the interaction is enhanced through haptic feedback and holographic vision, which provide a highly immersive and intuitive experience. We implemented and evaluated the system in a real 5G test network by comparing the categories and generation times of point-cloud objects with varying densities produced by trained models in real time, alongside network transmission performance. These experiments validate the effectiveness of the solution, highlighting its ability to support complex interactive applications with scalable performance. The results also provide valuable information on the use of generative AI for holographic applications in mobile networks, paving the way for future advances in immersive experiences [1].

979-8-3315-4370-9/25/$31.00 ©2025 IEEE

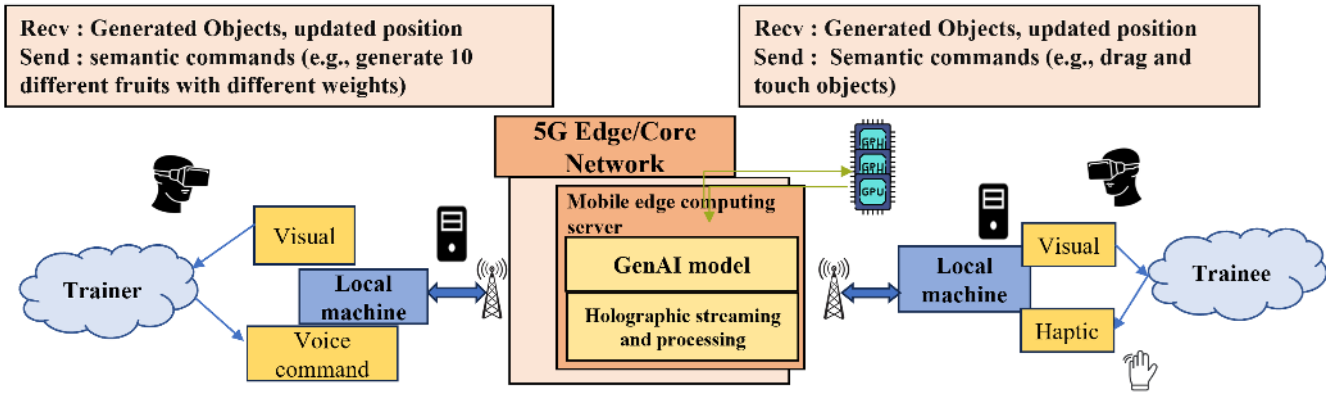

Fig. 1: Proposed GenAI assisted holographic streaming framework with mobile edge computing

II. Related Works

In this section, we provide a comprehensive review of the literature on recent advances in holographic and Generative AI applied for point cloud content.

A. Holographic application frameworks

The main challenge in live transmission of point-cloud videos is the substantial bandwidth required. To reduce traffic volume while maintaining a satisfactory user experience, existing solutions can be divided into two main categories. One approach uses machine learning to analyse the features of point cloud data, transmitting only the smaller-sized feature information and reconstructing the original content at the receiver's end. A notable example is [2], where the authors employ machine learning to divide a large frame into subframes from different viewpoints and selectively transmit based on the viewer's perspective. This method can improve Quality of Experience (QoE) by up to 85%. However, a common issue is the significant computation time required to encode and decode the features of the point cloud using machine learning algorithms [3], which limits their feasibility for real-time transmission. The other direction focusses on optimising the transmission architecture and process while maintaining general compression ratios [13]. For instance, the authors in [13] offload the computationally intensive point cloud processing tasks to a MEC server and investigate the impact of various congestion control algorithms and resolution levels. This approach provides an efficient solution for the deployment of holographic applications in mobile networks. Furthermore, the authors in [9] propose a collaborative framework called CoMIC, which aims to address the lack of infrastructure for multiuser XR applications by creating a collaborative, visual-first, and hologram-based computing environment. These enhancements help reduce the bandwidth required for real-time point-cloud transmission. Similarly, by integrating software-defined networking (SDN) and edge computing, TCPSbed [6] facilitates the development and testing of latency-sensitive applications, such as remote surgery and industrial automation. Although acknowledging that both solutions are still in the early stages of development, researchers have laid the groundwork for semantic communication. This includes defining its objectives, semantic encoding and decoding mechanisms, and transmission models [14]. By abstracting captured objects and scenes into semantic information, receivers can leverage generative artificial intelligence to reproduce the content from such information (e.g. text).

B. Generative AI for point cloud generation

Generative AI has emerged as a transformative technology across various fields, enabling the creation of complex and high-quality content in domains such as image synthesis, 3D modeling, and multi-modal applications. Different generative AI techniques exhibit unique methodologies and capabilities, making them suitable for distinct applications.

Diffusion models, such as Stable Diffusion [21], generate high-resolution images by iteratively refining noise through a latent diffusion process. This iterative approach enhances image quality and detail, making it particularly effective for tasks that require high fidelity, such as digital art and design. Additionally, optimisations in the latent space help reduce computational overhead while preserving visual quality.

Point-cloud generation models, such as Point-E [15], focus on 3D synthesis by leveraging a two-step process: converting a 2D image into a latent representation and subsequently generating a 3D point cloud. It relies on the attention mechanism, first introduced in "Attention is All You Need" [18], to process and generate multi-modal content. The self-attention mechanism allows the model to capture long-range dependencies in both textual and visual data, making it highly effective in generating images from natural language descriptions. 3D Gaussian Splatting [20] has recently gained attention as an alternative representation for efficient 3D scene reconstruction and rendering. Unlike traditional explicit mesh-based or implicit neural representations, Gaussian Splatting represents a scene as a set of anisotropic Gaussian kernels distributed in 3D space. This method enables real-time rendering with high-quality details, making it particularly promising for applications such as novel view synthesis, immersive virtual environments, and photorealistic scene reconstruction. Transformer-based architectures, including models like DALL-E [16], excel in multi-modal applications by generating images from textual descriptions. Their ability to integrate vision and language allows for flexible and creative content generation, making

them valuable in artistic creation, design automation, and interactive AI systems. However, these models rely heavily on large-scale attention mechanisms, which influence their computational complexity.

Each of these approaches presents unique strengths: diffusion models prioritise image detail and refinement, point-cloud generators focus on efficient 3D object creation, Gaussian Splatting enables high-quality real-time 3D rendering, and Transformer-based architectures facilitate multi-modal synthesis. The choice of a generative AI method depends on the specific application domain, whether it involves high-fidelity image generation, 3D modeling, real-time scene reconstruction, or text-to-image tasks."

III. FRAMEWORK DESIGN

In this section, we first introduce the real-time remote operation training scenario based on the proposed network architecture (see in Fig. 1) and explain how point-cloud objects generated by Generative AI are applied in this context. We then delve into the test network architecture that supports this application, as well as the design of its key components.

A. Use case: Generative AI-powered live interaction in mobile network

The proposed use case for Generative AI assisted metaverse application is remote training in industrial settings. For instance, the training procedure is initiated with both the trainer and the trainee wearing holographic headsets and the corresponding haptic gloves at trainee side. They connect to the same mobile network via their respective local mobile Customer Premises Equipment (CPE). During the initialisation phase, the trainer verbally communicates the specific content of the upcoming interaction to the trainee, for example, "Please select a 100 g apple from the various fruits you are about to see and place it on the workbench". The real-time voice instructions are transmitted to the MEC server, which then synchronously distributes them to all trainees. During the same time, the MEC server performs a semantic analysis on the trainer's command, generating different types of fruits with assigned weights and initial position information.

Subsequently, the real-time generated point cloud content is transmitted in real time to the trainer and trainee via the pre-established downlink connection of the mobile network and displayed through the headsets. Meanwhile, the trainee attempts to use the haptic gloves to touch the objects in the virtual space in order to identify and grasp the 100 g apple and drag it to the designated location. During this process, since each displayed object has been assigned a weight attribute, the corresponding vibrations and force feedback are correctly triggered when the local application detects an overlap between the glove coordinates and the object.

The changes in an object's position and orientation are represented by the differences in its coordinates over time. This information is embedded in TCP messages in the form of coordinates and transmitted to the trainer's end, where it is updated in real time. Since the size of these types of messages is usually very small (e.g., several KB), they do not consume much bandwidth and do not create a transmission performance bottleneck. This updating method will become a primary content update mechanism in remote training scenarios and will be repeatedly executed.

B. Key components of the proposed framework

Generative AI assisted application content generation: Next, we introduce the core components on the MEC server, specifically for generating the holographic objects. First, it is important to note that MEC servers (or GPU cards) are typically deployed as a cluster at the edge of the mobile network. In our use case, one MEC server is selected as the control server which is responsible for receiving semantic information from users. During the application, user instructions can be transmitted to the control server in the form of voice packets or text-based message packets. The control MEC server then distributes different text instructions to various generation servers (or GPUs) to perform parallel and distinct object generation. This generation process is based on the deployment and training of the Point-E model. Since the Point-E model comes with a built-in text prompt-based generation interface, no additional interface adjustments are required. In our test framework, we set the size of the individual object to be 4096 points, and the MEC server will determine the display size of individual points rendered on the user's holographic helmet, attempting to maximise the likelihood that the generated object is accurately perceived and understood by the end user.

The reason for adopting a parallel generation server design is that generating point cloud objects requires a considerable amount of time. Using parallelism, the total generation time cost can be effectively reduced. The control server distributes the generation tasks for multiple objects in the scene to parallel servers equipped with GPU cards. Once generated, it collects these objects, merges them, and maps them to a predefined physical space with coordinate attributes. Additionally, since the objects to be identified in the scene may belong to the same category but have different weights, it is necessary to properly store and trigger the corresponding haptic feedback programme at the control server, which need to be accurately implemented and deployed on the user's laptop. For the haptic feedback component, both the MEC server and the user side must have a consistent knowledge of haptic design. Specifically, the haptic grip feedback on the MEC server is configured at discrete levels 0, 1, and 2, which the user terminal must accurately translate to 0 Newtons, 10 Newtons and 20 Newtons, respectively, on the smart gloves [7].

Once the MEC server completes object generation, it transmits the generated content, such as images, 3D coordinates, and object properties (e.g., weight), to each trainee's local device, enabling a multisensory experience to be processed and displayed. Meanwhile, the system deployed on the user side continuously captures hand movements and position feedback, ensuring seamless transmission and interaction.

Integration of different data types: Then we describe how different data types can be supported, transmitted, and activated realistically in the test framework.

First, in cooperation with the multisensory equipment deployed on the user side, the MEC is able to process different data types required by the application. These data types integrate video, localisation, and haptic data to enable immersive user experiences. The holographic image of a given object or scenario is represented as a point-cloud volume with RGB and depth data. These data are integrated and rendered, which can be displayed on modern XR devices below 40 ms for low-resolution videos [12]. Each point leverages a 3D coordinate system (X, Y, Z) to enable the representation of 6-DoF objects. However, detailed holographic images can require hundreds of millions of points, resulting in bandwidth demands of hundreds of MB/s to GB/s, placing a significant challenge to be timely delivered via wireless uplink. User localisation data (e.g. for tracing position or hand move) is collected using the HTC VIVE Tracker v3.0 and Steam Base Station [8], which offers millisecond temporal granularity and millimetre spatial accuracy. A predefined physical space (for example, a cubic area of 2 meters) is assigned to the virtual coordinates X, Y, Z. Multiple trackers attached to body parts enable precise motion capture within the space. Trackers detect infrared signals from Steam tracking base stations to compute real-time 3D positions, updating every 22 milliseconds. Studies show that measurement errors are at millimetre levels, further reduced with more than two base stations [10], [11]. To realistically support haptic on the user side, haptic data is captured using modern gloves such as SenseGlove Nova [7], equipped with sensors and actuators that track movements of the finger joints and provide force feedback. Magnetic brakes generate up to 20 newtons of force across ten levels, simulating realistic interactions like handshakes or hugs. This vibration feedback enhances realism for multi-user or object interactions in shared virtual spaces. Together, these systems ensure precise, responsive, and immersive interactions for users. The transmission of these multisensory data at the user side relies on a haptic-integrated holographic video client running on a laptop. This client actively retrieves various types of data from the respective hardware devices, ensuring that each type is collected evenly and temporarily stored locally. Regarding the clock system, both the local clients and the MEC server are connected to a public Network Time Protocol (NTP) server to maintain global synchronisation of timestamps. On the client side, data types that share the same timestamp are assigned a unique frame ID. This ensures that multi-type data can be accurately identified as part of the same semantic action during network transmission and processing. Once the same frame ID is assigned to each data type, haptic and positional information is combined into a single frame, as they are small packets and operate within an integrated hardware system (e.g., SenseGlove Nova integrated with HTC VIVE Tracker). Meanwhile, holographic video content is encapsulated separately. These frames are then packaged into TCP packets, ready for transmission to the MEC server. Additionally, the client can relay the received haptic-integrated holographic video from the MEC server to the corresponding hardware devices. This includes forwarding vibration and force trigger commands to the gloves and delivering holographic video to the XR headset, relying on the hardware's development libraries.

IV. Implementation and Evaluation results

In this section, we present the evaluation results of the use cases based on the implemented test framework.

A. Implementation and evaluation tested

User-side settings: In terms of capturing live real-time holographic data, we simplify the setup by removing local cameras. A local laptop serves as the central processing unit, integrating the necessary hardware libraries and managing data streams to communicate with the edge server. Each trainer and trainee wears the Microsoft HoloLens 2. To track the trainee's real-time body position, two SteamVR base stations are deployed in each room to establish a coordinate system. The trainee also wears an HTC VIVE 3.0 tracker on their wrist (or other body parts) and a SenseGlove Nova, allowing for both force and vibration interaction. A data collection software implemented in C++ integrates the necessary hardware libraries, running on the local laptop to facilitate communication with the edge server.
Network entity implementation and deployment: The user-side client connects to a 5G CPE device, accessing the 5G standalone (SA) network via eMBB and establishing a TCP connection to the MEC server. The MEC server, deployed at the network edge, features dual NVIDIA A6000 GPUs, 256GB DDR4 memory, and a 2TB NVMe SSD for high-speed processing. Running Windows Server 2022, it hosts an enhanced Livescan3D codebase [13], which integrates haptic processing, delay monitoring, content filtering, and calibration for seamless operation.

Performance measurement: To measure the time to generate the point cloud, we added a timestamp to the generation model component to record the duration between the arrival of the text instruction and the completion of the object generation. To measure the transmission time of the generated object over the network, we recorded the time at the MEC server by tracking the interval between the frame being sent and receiving the corresponding acknowledgement. In the following experiments, we typically repeat the process twenty times and record the value for each iteration (30 iterations per evaluation).

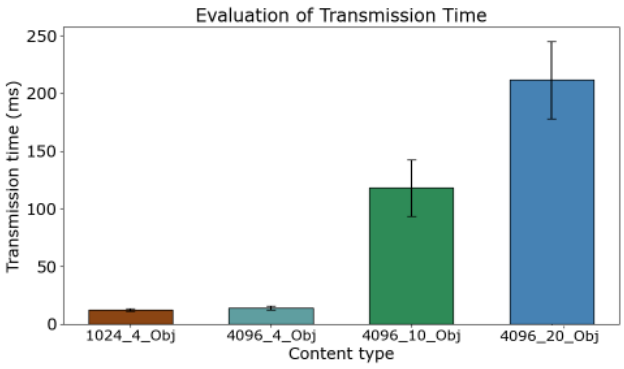

Fig. 2: Evaluation of transmission time

B. Implementation and evaluation tested

First, let us summarise the performance of real-time holographic object transmission over the network. We considered two point cloud densities: 1024 points (approximately 15KB) and 4096 points (approximately 60KB). Based on these two types, we transmitted different quantities and measured their downlink transmission latency, including 4×1024, 4×4096, 10×4096, and 20×4096 points, as shown in Fig. 2. As mentioned above, the generation of multiple objects can be accomplished using a GPU array. Specifically, our generation method involves two parallel GPU cards that each generate an object of a specified density upon receiving a generation request. During transmission, we simply simulate different objects by modifying the RGB colours and rotating some points (operations that are almost instantaneous). For the cases of 10 and 20 objects, we pregenerated the point clouds for direct transmission, aiming only to measure the upper limit of transmission performance. Figure 2 clearly shows that with a smaller number of objects, point cloud content can be transmitted within 1 to 2 Round-Trip Times (RTTs), providing a basis for designing more complex scene interactions. However, as the number of objects increases, the added transmission delay significantly reduces real-time performance. In particular, if a single frame contains 20 objects, its transmission time exceeding 200 ms and the performance fluctuation confirm that this scenario is only suitable for static scenes.

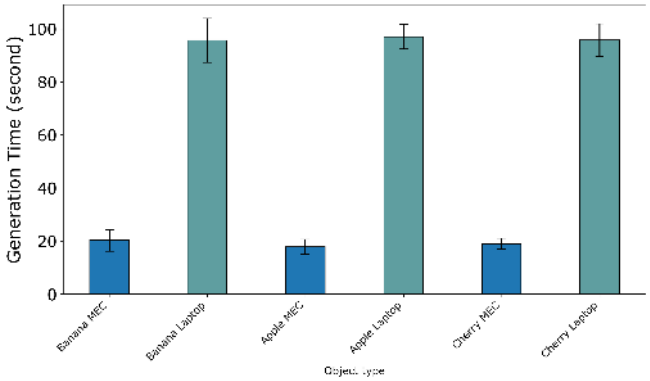

Fig. 3: Comparison of generation time between MEC server and user laptop

Next, we compared the speed of generating simple objects between a laptop with a discrete graphics card on the user terminal side and an MEC server. In this specific use case, the objects used in the training scenario belong to the ”fruit” category, with a point-cloud resolution of 4096 points. The comparison results are illustrated in Fig. 3.

Our findings reveal that even with an NVIDIA A6000 graphics card and matching CPU memory units, the MEC server requires nearly 18 seconds to complete the generation of a single object. Although not instantaneous, this speed is sufficient for semi-real-time interactive scenarios. For example, in a training session, a trainer can initiate the content generation command while introducing the training objectives. By the time the oral instruction is complete, the corresponding object would have been generated and ready for transmission.

In contrast, the laptop on the user side, equipped with a GTX 1060 GPU, took nearly 100 seconds to generate the same object. This stark difference underscores the necessity of using MEC servers for point cloud content generation in interactive applications. The computational limitations of end-user devices make them unsuitable for handling the intensive workloads required for real-time or semi-real-time object generation. By offloading this task to the edge, we ensure that computationally heavy processes do not hinder the user's interactive experience.

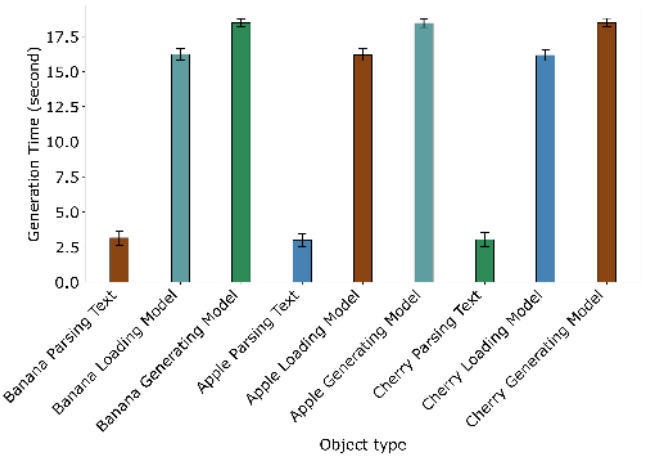

Fig. 4: Time consumption of different generation types

To further analyse the object generation performance of the MEC server, we break down the object generation time into three cumulative components: processing the text-based generation instruction, loading the corresponding object generation model, and completing the final generation task, where each stage includes the time from all preceding stages. The cumulative results, shown in Fig. 4, illustrate how time is distributed between these stages.

The results indicate that as long as the number of points in the point cloud remains constant, the generation speed remains largely consistent. This is attributed to the stable diffusion-based generation algorithm, which uses a point-set position diffusion method rather than feature-specific generation. This approach not only ensures greater consistency in generation times, but also enhances the algorithm's controllability when deployed in edge environments. The uniformity in performance highlights the suitability of this method for edge applications where reliability and predictability are crucial.

Interestingly, our analysis shows that the model loading stage consumes the majority of the generation time, often exceeding 10 seconds. This latency is a significant bottleneck as it limits the overall speed of the system. Compared to model loading, the time required for point cloud generation and text instruction processing is relatively minor. This finding suggests several avenues for optimisation. For example, exploring more efficient model file formats, such as reducing file sizes or preloading models, could significantly improve performance. Additionally, optimising the deployment code for the edge environment and employing lightweight model architectures could further reduce latency.

Finally, we reduced the density of point clouds to 1024 points and compared the generation time in Fig. 5. It was

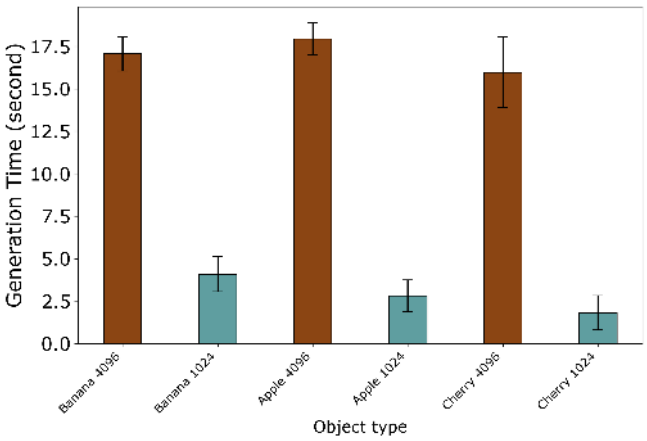

Fig. 5: Generation time of different point density

evident that while the point cloud density was reduced by a factor of 4, the generation time decreased by almost 6 to 7 times, taking less than 4 seconds. At this lower point cloud density, differences in generation speed between object categories became apparent. For example, we observed that the generation speed for "cherry" was significantly faster, taking less than 2 seconds.

These findings provide application designers with greater flexibility, such as creating scenes with a clear hierarchy by combining a single high-density object with multiple low-density objects. Furthermore, we plan to leave the investigation of generation speed across different object types and densities to the future work. In summary, the evaluation demonstrates the effectiveness of using MEC servers for point-cloud content generation in semi-real-time applications. Although the current generation speed is sufficient for the designed training scenario, our future work will focus on optimising the model loading process to further reduce latency and improve scalability. These improvements will improve the responsiveness of AI-based generative systems, making them even more suitable for diverse industrial and educational use cases.

V. CONCLUSION

In this work, we proposed a test network architecture that integrates MEC with generative AI models to address the challenges of real-time holographic streaming in mobile networks. By generating high-quality holographic content from compact semantic inputs at the network edge, our approach significantly reduces uplink bandwidth demands and ensures low-latency processing for tasks such as calibration, compression, and rendering. Additionally, bandwidth efficiency is further improved by transmitting only updates to object coordinates and changes during interactions, rather than full scene data. The system was validated through an interactive operational training use case in a real 5G test network. The results demonstrated that our solution supports scalable and efficient real-time holographic applications while conserving network resources. This work highlights the potential of integrating MEC and generative AI to enable immersive and responsive holographic experiences, providing a solid foundation for future advancements in this field.

VI. ACKOWLEDGEMENT

This project is sponsored by the EU Horizon Europe SPIRIT project (Grant Agreement 101070672, https://www.spirit-project.eu/). The authors would also like to acknowledge the support of 5GIC industry members (5GIC) (http://www.surrey.ac.uk/5gic) for this work.

REFERENCES

[1] Lema, Maria A., et al. "5G case study of Internet of Skills: Slicing the human senses." 2017 European Conference on Networks and Communications (EuCNC). IEEE, 2017.
[2] Zhang, Anlan, et al. "YuZu:Neural-Enhanced Volumetric Video Streaming." 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22). 2022.
[3] Yang, Mengyu, et al. "A Comparative Measurement Study of Point Cloud-Based Volumetric Video Codecs." IEEE Transactions on Broadcasting (2023).
[4] Huang, Yakun, et al. "Toward holographic video communications: a promising AI-driven solution." IEEE Communications Magazine 60.11 (2022): 82-88.
[5] Babaei, Aptin, Parham M. Kebria, and Saeid Nahavandi. "5G for Low-latency Human-Robot Collaborations; Challenges and Solutions." 2022 15th International Conference on Human System Interaction (HSI). IEEE, 2022.
[6] Polachan, Kurian, et al. "TCPSbed: A Modular Testbed for Tactile Internet-Based Cyber-Physical Systems." IEEE/ACM Transactions on Networking 30.2 (2021): 796-811.
[7] SenseGlove Nova 1.X https://senseglove.gitlab.io/SenseGloveDocs/nova-glove.html?highlight=force%20feedback
[8] Caserman, Polona, et al. "Real-time body tracking in virtual reality using a VIVE tracker." Virtual Reality 23 (2019): 155-168.
[9] Han, Bo, et al. "CoMIC: A collaborative mobile immersive computing infrastructure for conducting multi-user XR research." IEEE Network (2022).
[10] Weber, Mitchell, et al. "Dynamic Pose Tracking Accuracy Improvement via Fusing HTC VIVE Trackers and Inertia Measurement Units." International Journal of Precision Engineering and Manufacturing 24.9 (2023): 1661-1674.
[11] Lwowski, Jonathan, et al. "HTC VIVE tracker: Accuracy for indoor localization." IEEE Systems, Man, and Cybernetics Magazine 6.4 (2020): 15-22.
[12] Long, Zijian, Haiwei Dong, and Abdulmotaleb El Saddik. "Interacting with New York city data by HoloLens through remote rendering." IEEE Consumer Electronics Magazine 11.5 (2022): 64-72.
[13] Qian, Peng, et al. "Remote Production for Live Holographic Teleportation Applications in 5G Networks." IEEE transactions on broadcasting 68.2 (2022): 451-463.
[14] Zhu, Yuanwei, et al. "A semantic-aware transmission with adaptive control scheme for volumetric video service." IEEE Transactions on Multimedia (2022).
[15] Nichol, A., Pohl, T., et al. "Point-E: A System for Generating 3D Point Clouds from Images." OpenAI Research, 2022.
[16] Ramesh, A., Dhariwal, P., Nichol, A., et al. "DALL-E: Creating Images from Text Descriptions." OpenAI Blog, 2021.
[17] Nichol, A., Dhariwal, P. "Efficient 3D Object Generation with Point Clouds." OpenAI Technical Report, 2022.
[18] Vaswani, A., Shazeer, N., Parmar, N., et al. "Attention is All You Need." NeurIPS Proceedings, 2017.
[19] Huynh, Vu San Ha, et al. "User-Intent Aware Transport-Layer Intelligence for Frame Synchronisation in Multi-Party XR Application." 2024 IEEE Gaming, Entertainment, and Media Conference (GEM). IEEE, 2024.
[20] Fei, Ben, et al. "3d gaussian splatting as new era: A survey." IEEE Transactions on Visualization and Computer Graphics (2024).
[21] Luo, Shitong, and Wei Hu. "Diffusion probabilistic models for 3d point cloud generation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.