applied sciences

MDPI

Article

Enhancing the Performance of XR Environments Using Fog and Cloud Computing

Eun-Seok Lee [1] **and Byeong-Seok Shin** [2,*]

[1] Department of Computer Science & Engineering, Yuhan University, Bucheon 14780, Republic of Korea; elflee77@gmail.com

[2] Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea

* Correspondence: bsshin@inha.ac.kr; Tel.: +82-860-7452

Abstract: The extended reality (XR) environment demands high-performance computing and data processing capabilities, while requiring continuous technological development to enable a real-time integration between the physical and virtual worlds for user interactions. XR systems have traditionally been deployed in local environments primarily because of the need for the real-time collection of user behavioral patterns. On the other hand, these XR systems face limitations in local deployments, such as latency issues arising from factors, such as network bandwidth and GPU performance. Consequently, several studies have examined cloud-based XR solutions. While offering centralized management advantages, these solutions present bandwidth, data transmission, and real-time processing challenges. Addressing these challenges necessitates reconfiguring the XR environment and adopting new approaches and strategies focusing on network bandwidth and real-time processing optimization. This paper examines the computational complexities, latency issues, and real-time user interaction challenges of XR. A system architecture that leverages edge and fog computing is proposed to overcome these challenges and enhance the XR experience by efficiently processing input data, rendering output content, and minimizing latency for real-time user interactions.

Keywords: extended reality; cloud computing; edge computing; LiDAR; point cloud; compression; real-time interaction

check for updates

Citation: Lee, E.-S.; Shin, B.-S. Enhancing the Performance of XR Environments Using Fog and Cloud Computing. *Appl. Sci.* **2023**, *13*, 12477. https://doi.org/10.3390/app132212477

Academic Editors: Joon-Min Gil and Jisu Park

Received: 16 October 2023
Revised: 15 November 2023
Accepted: 16 November 2023
Published: 18 November 2023

1. Introduction

The rapid advances in digital technologies are fundamentally transforming daily life. One of the most notable paradigm shifts in this ongoing transformation is the rapid development of extended reality (XR) technologies, encompassing augmented reality (AR) [1,2], virtual reality (VR) [3], and mixed reality (MR) [4].

XR, as a technological domain, revolves around the fusion or alteration of the physical world and the digital domain, forging entirely new digital realities by obfuscating the conventional boundaries that delineate reality from the virtual world. VR achieves complete user immersion within virtual environments, enabling interactions within digitally synthesized settings entirely distinct from the physical world. AR overlays virtual objects or information seamlessly onto the real-world context, augmenting everyday experiences with rich layers of digital content. MR represents a dynamic convergence of the virtual and real worlds, allowing users to interact with virtual entities within the authentic physical environment.

These XR technologies are revolutionizing how people engage with digital information and are finding diverse applications across a spectrum of domains, from entertainment and gaming to the educational, healthcare, and industrial sectors. As XR continues its evolutionary trajectory, it has the potential to redefine perceptions and interactions within the tangible and digital dimensions, presenting novel avenues for innovation and creative exploration.

XR technology demands highly computational tasks, particularly in high-performance 3D rendering and processing extensive 3D point clouds [5]. These operations are particularly computation-intensive and must be executed in real time and be driven by user interactions, requiring seamless and uninterrupted rendering of virtual environments. Consequently, deploying such operations in a local environment was initially considered a reasonable choice regarding performance and stability. On the other hand, the limitations of local environments have become apparent with the advances in cloud technology and the exponential growth in the number of users. This has led to the necessity of transitioning to cloud-based solutions, driven by the following reasons.

First, a cloud-based XR environment [6] offers scalability, allowing it to accommodate a surge in users, with the advantage of accessibility from anywhere at any time. In contrast, local environments can be complex and expensive regarding hardware upgrades and maintenance. Leveraging the cloud enables swift expansion and upgrades, ensuring uninterrupted services for users. This scalability of cloud environments facilitates user collaboration and data sharing, particularly in scenarios involving numerous users collaborating or exchanging information within virtual spaces. Cloud-based systems are adept at handling such requirements seamlessly.

Second, in the cloud environment, it is feasible to collect and analyze large volumes of data to learn user patterns and offer personalized experiences [7]. This enables the provision of richer and tailored XR environments for users. In addition, cost savings are achieved because of the reduced expenditures associated with hardware procurement and management, such as storage and servers. Leveraging cloud services allows for the efficient management of costs by utilizing resources only when needed.

Lastly, cloud environments facilitate seamless updates and maintenance tasks through centralized management and automated processes [8]. Automation processes support the automatic updating of system components and software. Swift remedial actions can be taken when issues arise. Furthermore, cloud service providers can deploy new features and improvements promptly, addressing user requirements and security concerns. These advantages enhance the user experience and effectively incorporate the latest technologies in cloud-based XR environments.

Therefore, transitioning to the cloud is considered an essential step in constructing XR environments in a more modern and scalable manner, providing users with superior experiences. On the other hand, there are certain challenges involved in moving XR environments to the cloud. XR environments need to process and transmit large volumes of data in real time, posing significant network bandwidth and latency obstacles. Transmitting data to the cloud results in increased volumes of data. Moreover, the delays incurred while data are transferred to cloud servers can hinder the user experience. Addressing these latency issues requires network infrastructure optimization and data compression technologies.

Furthermore, real-time interactions with users must occur seamlessly in XR environments. Delays in data processing [9] and rendering within the cloud environment can hinder the delivery of a smooth user experience. Therefore, methods must be explored to ensure real-time processing and responsiveness, even within the cloud environment.

Therefore, developing new technologies and strategies for cloud-based XR environments is essential to address these challenges. Research focusing on network bandwidth optimization and real-time processing optimization plays a critical role in resolving these issues. Therefore, this paper proposes a novel XR system that leverages edge and fog computing to overcome the limitations of existing cloud-based XR environments. This approach aims to achieve network bandwidth optimization and real-time processing optimization, ensuring a seamless XR experience for the users without interruptions.

In the context of XR environments, the system was designed with a central focus on real-time user interactions. Consequently, the interaction component is considered one of the most critical elements within the system. Within this system, interactions play a prominent role in the input and output stages.

From the input perspective, numerous sensors generate extensive data, collecting information on the user's actions and environmental factors. Effectively processing and analyzing data to enable real-time interaction is paramount. Nevertheless, this necessitates efficient data compression and decompression processes, and minimizing network latency when transmitting data to servers.

In the proposed system, edge computing was designed to meet these requirements. Edge computing applies data compression techniques to reduce the volume of data, minimizing data transmission and processing delays. Typically, in XR environments, the devices used for user interactions are installed at specific locations, designed to provide users with a seamless interaction between the physical world and virtual environments. These devices, which include various sensors and displays, are controlled by local edge computers. These edge computers must process operations requiring rapid local processing, such as the real-time handling of extensive input data. Synchronizing the time among various hardware components ensures a natural interaction for users.

On the output side, real-time rendering of visuals and sound is crucial. Multiple edge computers are used to provide users with a smooth and realistic virtual environment. These edge computers are integrated to harmoniously coordinate the synchronization of input data and rendering tasks for output data. Hence, a dedicated server operates within the fog environment to guarantee the real-time performance of the system.

The proposed fog system is crucial for optimizing the interaction and user experience. The system functions as a dedicated server, integrating data collected from edge computers and input data while performing time synchronization tasks for the input data. Furthermore, it ensures precise timing for rendering visuals and sound based on the synchronized data, coordinating with the edge computers responsible for rendering. After processing the data collected by the fog system, it consolidates the key information and forwards it to the central cloud system. This central cloud system efficiently manages multiple users and virtual worlds, enabling natural interactions through synchronization. This architecture allows the system to provide users with realistic, real-time interaction and effectively establish a cloud-based XR environment. This design is critical for delivering a smooth and real-time interactive user experience.

This paper reports on a proposed system that utilizes edge and fog computing in a cloud-based XR environment. Section 2 reviews the related research and examines trends in the design of cloud-based XR systems and interaction-related research. Section 3 provides details of the proposed system architecture and core technologies. Section 4 presents the experimental results and performance tests. Section 5 concludes the paper and discusses future research directions, systematically introducing the present research and emphasizing the significance of cloud and edge computing in XR environments.

2. Related Works

VR [10,11] immerses the users entirely in a virtual world, enabling interactions completely detached from reality. AR [12] enriches everyday experiences by overlaying virtual objects or information onto the real world, often through smartphone applications (apps) or AR glasses. MR combines virtual and real worlds, allowing interactions with virtual objects within the real world. XR technologies are revolutionizing various fields, including gaming, education, healthcare, industrial automation, arts, and entertainment. These technologies continue to evolve and expand.

XR environments blur the boundaries between the physical and digital realms, overcoming the physical space limitations [13]. This convergence provides new opportunities for users to interact with diverse digital content daily. XR shapes the future of computing and interaction paradigms. Research efforts are currently underway to enable XR experiences in cloud environments, aiming to overcome the constraints related to the physical locations and computational performance.

Various research topics around cloud-based XR system design and edge computing in XR systems exist. This section introduces these research topics. Research related to cloud-based XR system design [7] focuses on creating virtual environments centered on cloud servers and delivering them to the users. Although these systems can provide high-performance computing and the ability to handle large-scale data, they are constrained by network latency and bandwidth issues, particularly in achieving real-time interactions.

Research on XR system design using edge computing [14,15] has explored methods of improving user interactions by deploying edge computers in local environments. Edge computers can handle large-scale data processing and real-time rendering locally, enhancing the user experience. Research on hybrid cloud–edge environments [16] explores methods to combine cloud and edge computing and provide optimal performance and user experiences. In methods involving mobile devices or VR equipment, there are ongoing efforts to address the latency issues, which can be problematic when using direct streaming or edge computing solutions [17,18]. This study addressed network latency issues, efficiently processing large datasets, and enabling real-time interactions.

Research on real-time data processing and compression technologies has investigated methods [19–21] to process and compress large datasets. These technologies reduced the volume of data, minimized network latency, and provided users with faster response times. These research efforts encompassed two approaches for compressing point cloud data. The direct compression approach compressed point cloud data directly using various data structures and compression techniques [19]. Alternatively, researchers have converted raw-point cloud data, such as packet data, into lossless range images. They applied different compression algorithms, such as those used for images and videos, e.g., MPEG, to reduce the volume of data [21]. Both methods aim to reduce the data size while maintaining essential information in point cloud datasets.

Research on the interactions and user experience has focused on enhancing the user's experience within XR environments [22–24]. This study evaluated various methods to improve user experience, emphasizing user interactions and feedback processing. These related studies are integral to enhancing XR environments through cloud-based XR systems and edge computing, providing valuable insights and inspiration for these research endeavors.

3. Proposed System Architecture

To provide a detailed explanation of the XR content slated for use as a testbed to validate the efficiency of the proposed cloud platform, the following description is provided. The experimental content facilitates a virtual environment wherein multiple users engage in collaborative interactions within a cubic space through an online platform. This virtual scenario takes the form of an interactive storybook.

Upon touching the virtual walls, users trigger immediate interactions. The touch interactions are detected through LiDAR sensors embedded in the walls. To promptly respond to user actions, laser projectors are employed to project visuals onto the four walls and the floor, while speakers simultaneously reproduce corresponding audio feedback. Consequently, users experience real-time responses to their interactions in the virtual world.

This content allows users to witness real-time reactions to their interactions in the virtual environment, where touching the walls or interacting with elements previously manipulated by other users leads to dynamic responses. The system is designed to offer users a multi-sensory experience, providing visual and auditory feedback in real-time.

In a cloud-based XR environment, generating and transmitting large-scale 3D graphics and sensor data are essential. However, this can lead to network bandwidth issues and the need to prevent overload due to data streaming. In recognizing these challenges, we propose an alternative approach that leverages edge and fog computing, which proves to be more efficient in designing and implementing XR environments. Edge and fog computing is more efficient in designing and implementing XR environments by considering these constraints rather than relying solely on cloud-based systems, such as the real-time transport protocol (RTP) [25]. This approach minimizes data transmission to the cloud, especially for equipment with a significant volume of data, such as light detection and ranging (LiDAR) sensors and projectors installed in the real world. These devices can seamlessly receive data in the local environment through edge computers.

Typically, in XR environments, content engages users through a process such as Figure 1. Users perform actions as required by the content in a designated space, and these actions need to be detected through sensing equipment provided in the XR environment (in our proposal, LiDAR sensors). Such equipment, like LiDAR sensors, typically transmits data such as point clouds. Based on this data, user actions are detected. These detected actions are then reflected in the XR content through character actions, enabling the content to provide feedback to users through screens and sound.

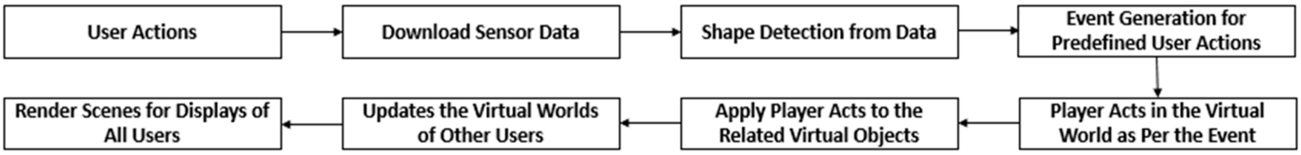
User Actions
Download Sensor Data
Shape Detection from Data
Event Generation for Predefined User Actions
Player Acts in the Virtual World as Per the Event
Apply Player Acts to the Related Virtual Objects
Updates the Virtual Worlds of Other Users
Render Scenes for Displays of All Users

Figure 1. Block diagram for proper XR interactions.

The distinct characteristics of XR environments are that, when executed on local computers, there are fewer constraints on data communication and data processing. However, there are limitations, such as higher costs and difficulties in facilitating interactions among multiple users. Consequently, our proposal aims to address real-time processing challenges that arise when building XR environments in a cloud setting. We achieve this by optimizing the volume of data and leveraging dedicated servers.

The input data for our system primarily consists of 3D point cloud data detected using a LiDAR sensor. These data are transmitted in real-time, with a scanning frequency of 20–30 times per second to determine whether the user's body is in contact with the wall. For touch interactions on the wall, multiple users may utilize various parts of their bodies, necessitating the detection of multiple touches from different positions. To achieve this, we deploy multiple sensors, as shown in Figure 2, to detect these touches with precision. However, it is worth noting that, while this sensor arrangement enables precise touch interactions, it can also result in data transmission delays due to the larger volume of data. These delays may ultimately hinder user interaction and cause delays in the XR environment's output.

Generating and transmitting large-scale 3D graphics and sensor data are essential in a cloud-based XR environment. This can lead to network bandwidth issues and the need to prevent overload due to data streaming. Edge and fog computing enhance local data processing and real-time on-site interaction, significantly improving the user experience.

Figure 2 shows the architecture of the proposed system. Unlike conventional cloud-based methods, the proposed approach minimizes data transmission to the cloud. Therefore, equipment with a significant volume of data to be transmitted, such as light detection and ranging (LiDAR) sensors and projectors installed in the real world, can receive data in the local environment through edge computers.

The input data for this system was the 3D point cloud data detected using a LiDAR sensor. These data were transmitted primarily in real-time, scanning the volumetric area as point cloud data at 20–30 times per second to determine if the user's body was in contact with the wall. For touch interactions on the wall, multiple users can utilize various parts of their bodies, requiring the detection of multiple touches from different positions. Therefore, multiple sensors are deployed to detect multiple touches, as shown in Figure 3. While this sensor arrangement enables precise touch interactions, data transmission delays can increase because of the larger volume of data. Such delays can ultimately hinder user interaction, delaying the output utilized as an element of the XR environment.

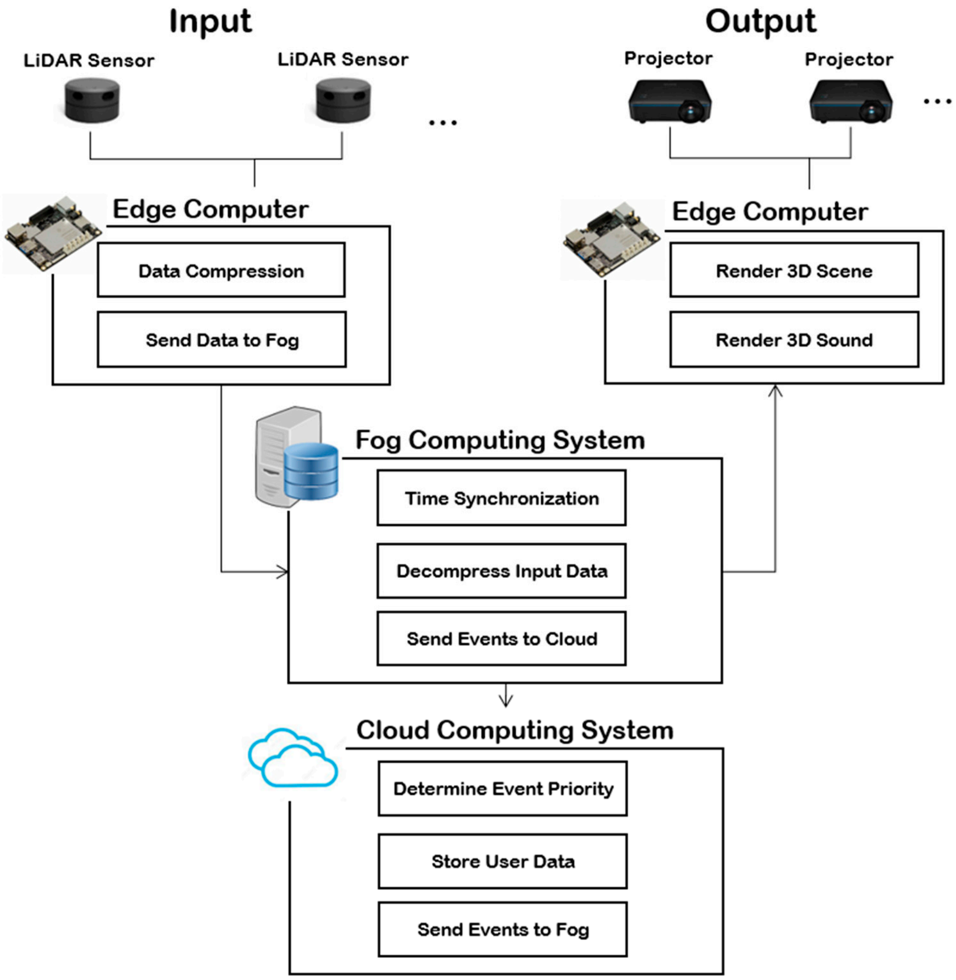

Figure 2. System architecture of the proposed method.

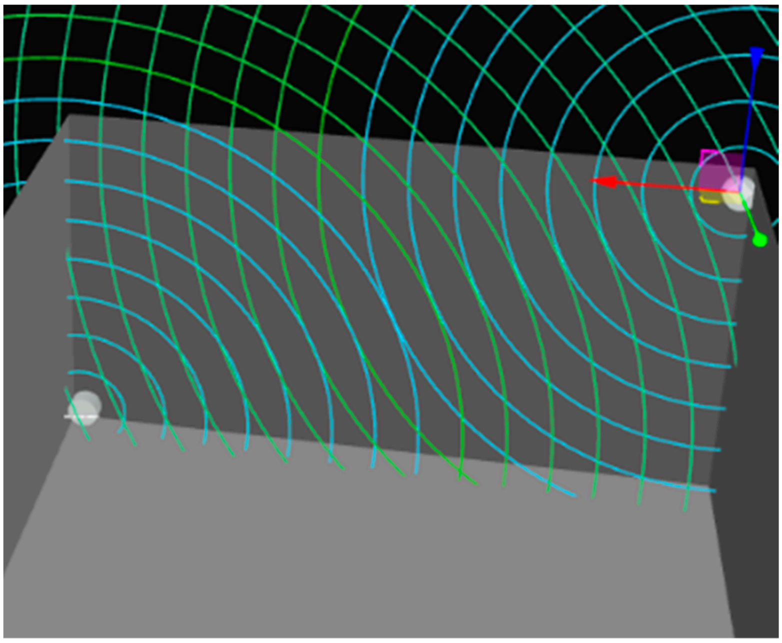

Figure 3. Simulating touch detection of the wall using two LiDAR sensors.

In the proposed system, edge computers are critical as they receive sensor data in real time in the local environment, making them essential devices for efficient data reduction. Typically, network speeds are slower than the CPU or GPU processing speeds. Therefore, compressing the data on these edge computers before transmitting it to the cloud can alleviate data bottlenecks. The system divides the target wall into minimum unit-sized segments to facilitate touch detection, as shown in Figure 4. Multiple points map to a single unit because most touches are concentrated in specific areas. For content execution, it is essential to know where touches are detected. Thus, it is advantageous to selectively send only the areas where touch detection occurs rather than transmitting all the points.

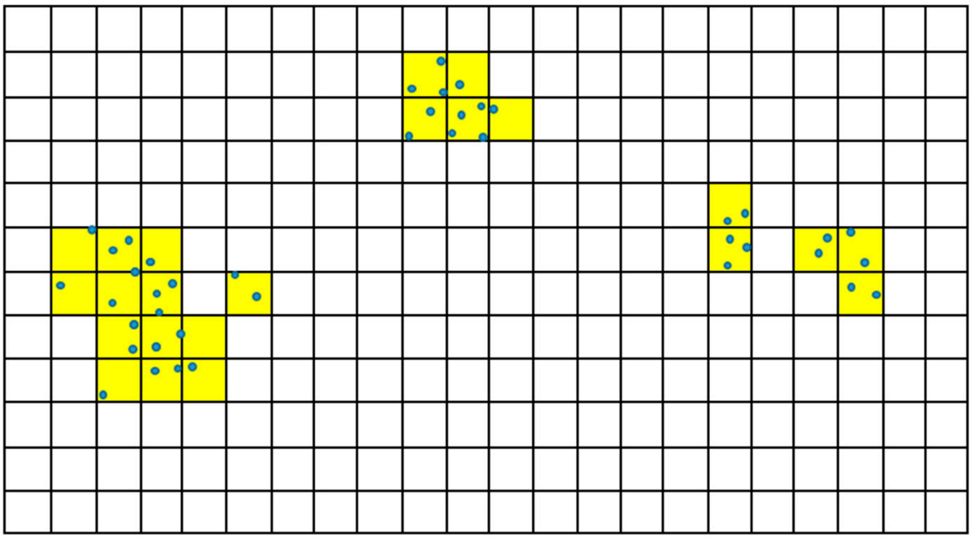

Figure 4. Uniformly segmented areas in the content divided into detection units. The blue circles represent point cloud data detected from LiDAR sensors, while the yellow box indicates the areas where touch detection is inferred.

In the proposed method, for efficient compression, the edge computer assesses the areas that detect touches based on the received input points. This approach efficiently removes redundant points detected at the same location. Furthermore, it allows the data to be structured in an image format with screen resolution, distinct from irregularly scanned LiDAR data. This structured data can be applied with various compression techniques like Huffman encoding [26] or H.264 [27], enhancing the compression efficiency.

The proposed method introduces a compression technique that utilizes previously scanned data to reduce data transmission. This compression approach sends information about touch detection for segmented areas to the content server. After the initial data transmission, subsequent data transmissions only send the coordinates of the altered data to make it easier for the server to determine touch detection. A transmission control protocol (TCP) must be used because it is crucial not to lose data. The system avoids transmitting data without user input by sending only the detected coordinates. In addition, it can respond immediately with minimal data transmission when changes are detected. Therefore, this approach offers advantages over the traditional method of directly transmitting scan values from sensors to the cloud.

Upon receiving such input, the edge computer responsible for the output must process the received data promptly in real time. On the other hand, cloud-based systems often experience delays in data transmission depending on network conditions. Hence, our proposed method leverages a fog computing system to establish a dedicated content server, distinct from the cloud, which operates as a dedicated server [28]. This approach involves the fog server assessing touch-related data from the edge, processing the output accordingly, and relaying this information to the cloud server for synchronization with other XR users.

Several crucial constraints must be considered when incorporating edge and fog computing into XR environments. XR environments require real-time interactions for quality user experiences, emphasizing the importance of processing inputs and outputs as locally as possible to mitigate the latency issues caused by data transmission in cloud computing. Hence, local computers, such as edge computing, can process input and output data rapidly, minimizing latency and providing users with a seamless interaction.

XR environments that merge virtual and real-world settings were constructed. In this XR environment, virtual worlds are rendered on all the walls of a small room, providing users with the sensation of being immersed in a virtual world within the physical one. As shown in Figure 1, multiple laser projectors were used to project images onto various walls, and user inputs are detected using LiDAR sensors [29] when users interact with the walls. The proposed system employs edge computers to handle extensive point cloud data generated from LiDAR sensors, render screens and sound, and ensure swift processing [30].

First, the edge computer responsible for processing input data received all the data from the LiDAR sensor. On the other hand, the data size was too large to be transmitted as a complete point cloud. Therefore, this study used a method of extracting and compressing only the data that differs from the previous data. This compression helped mitigate the problem of input data that will cause bottlenecks. Once the compressed data were transmitted to the central server, the original data were restored, and the touch events were determined. Subsequently, the input results were displayed on the screen to facilitate interactions with the user, which was the fundamental processing method.

On the other hand, synchronizing input and output can be challenging because input and rendering are processed separately on each edge computer. Therefore, the proposed approach established a fog system using a local private cloud. This allowed for faster data exchange than with a typical cloud environment. Within this system, synchronization was designed based on the operational time of edge computers in the form of dedicated servers. Figure 5 shows the synchronization method in the proposed system.

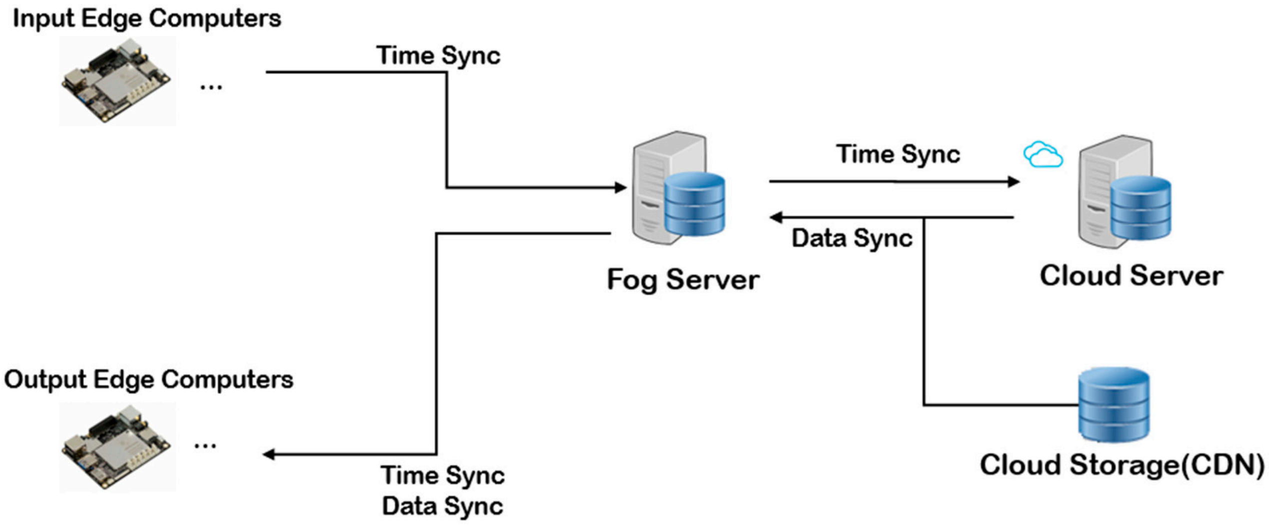

Figure 5. Synchronization method in the proposed system.

Time synchronization is a crucial element in real-time systems. In the proposed method, the fog system synchronized the time of all edge computers through time alignment during the initial connection. In addition, it periodically checks and adjusts time discrepancies to maintain synchronization. Through this process, the fog system's server can deliver data or events to the output edge computer accurately and enable real-time rendering.

Nevertheless, addressing response-delay situations due to network conditions is essential. The proposed approach used the same clock to make all content progress identical between the fog and edge computers, except for rendering tasks. The server was consistently prepared to render data identical to that of the output edge computer. It handled operations responding to input data, ensuring synchronization among all edge computers via the fog. In this manner, the proposed system minimized delays and provided interactive content precisely on time.

Consequently, if a specific output edge computer experiences delays, the system utilizes pre-processed data from the fog to synchronize with the data flow, enabling rendering that is identical to the other edge computers precisely and on time. This approach leveraged the edge computer performance using local computing power to render and pre-compute the content flow of the fog. The system provided efficient results with fewer devices. Moreover, compared to conventional RTP-based methods, which may encounter buffering issues due to network latency and synchronization errors when using multiple displays, the direct real-time rendering performed by edge computers offers superior responsiveness and synchronization speed.

Data processed in cloud systems are typically categorized into static resource data used in content and user data. Static resource data are generally large and known to many users. Such data are updated by storing in central cloud storage and provided via a CDN approach, allowing updates to be made at each fog or edge location. User data, however, require real-time interaction and are processed in the fog. The data are transformed from input data into touch events to reduce the size and are transmitted to minimize delays. This approach enables central data collection for various users in the cloud while minimizing device and edge computer usage adapting to individual XR system environments.

4. Experimental Results and Discussion

This section reports the efficiency of the proposed method by comparing the compression rate and interaction performance when using the edge computer-based compression method in a testbed environment. The experimental setup used a Mini PC (Intel 10th Gen i5, 16 GB RAM) for the edge computer, while the cloud environment consisted of a self-built personal cloud with 6th Gen Intel i3 processors and OpenStack [31]. Four touchscreen walls were used, each equipped with two LiDAR sensors, totaling eight sensors, and four projectors were employed. For content delivery, a dedicated server format was used in the fog system, and a rendering-specific client was formed and executed on the output edge. The LiDAR sensor rotated 30 times per second, transmitting 10,000 points per rotation, while the projector had a resolution of 1600 × 900, as used in the experiments.

The data compression rate experiments indicated an average data compression rate of approximately 70–80%, as listed in Table 1. This compression allowed for the data to be reduced to sizes of 1 KB or less during restoration and transformation into event messages.

Table 1. Average compression rates are based on the number of touches.

Area	User Touches	Compression Ratio
Front	5	83
Back	6	81
Left	5	77
Right	6	77

Hence, in recent network environments, where the amount of data transferred can contribute to network latency in addition to pure response delay, this approach resulted in no network latency between the cloud and the fog because the data size was tiny. As the data compression rate increases, the required computing resources in the fog can be reduced, solidifying the role of edge computers.

The second experiment analyzed the interaction latency by monitoring the delay count over five minutes through logs. Table 2 lists the following scenarios: the proposed method was used; only the edge and cloud were used; the edge, fog, and cloud were used; only the cloud was used. When using only the cloud or fog, rendering was replaced with streaming (RTP), allowing for an evaluation of the impact of the rendering method on latency. Unreal

Engine 4 [32] was used for the real-time streaming approaches [25] without edge computers. In addition, the Unreal Engine was used as the dedicated server [28].

Table 2. Number of occurrences of latency that affects the screen output for five minutes.

Area	Our Method	Use Edge + Cloud	Use Fog + Cloud	Use Cloud Only
Front	1	7	37	92
Back	0	4	46	108
Left	0	12	45	112
Right	1	8	31	99

Minimal latency was observed when utilizing edge computers. This result was attributed to the effective compression of most data through edge computing. On the other hand, in scenarios without a fog system, rendering delays and issues related to time synchronization through data updates occurred multiple times, and there were instances of network response latency in the cloud. Therefore, latency issues were minimized when the fog system played a significant role in user interaction, and the edge focused on data compression.

The touch recognition speed for the output from the moment of input was measured. This measurement was conducted to provide a clearer comparison of the responsiveness with the existing approaches from previous experiments. Figure 6 illustrates the measurement conducted in the same environment. The delay times were compared based on logs created when rendering in the time-synchronized input/output edge computers. Measurements were taken over time, and the results are based on a one-minute measurement in a 15 fps environment. Figure 6 records the data at 30 frame (two second) intervals for visual clarity and to reduce the amount of visualized data.

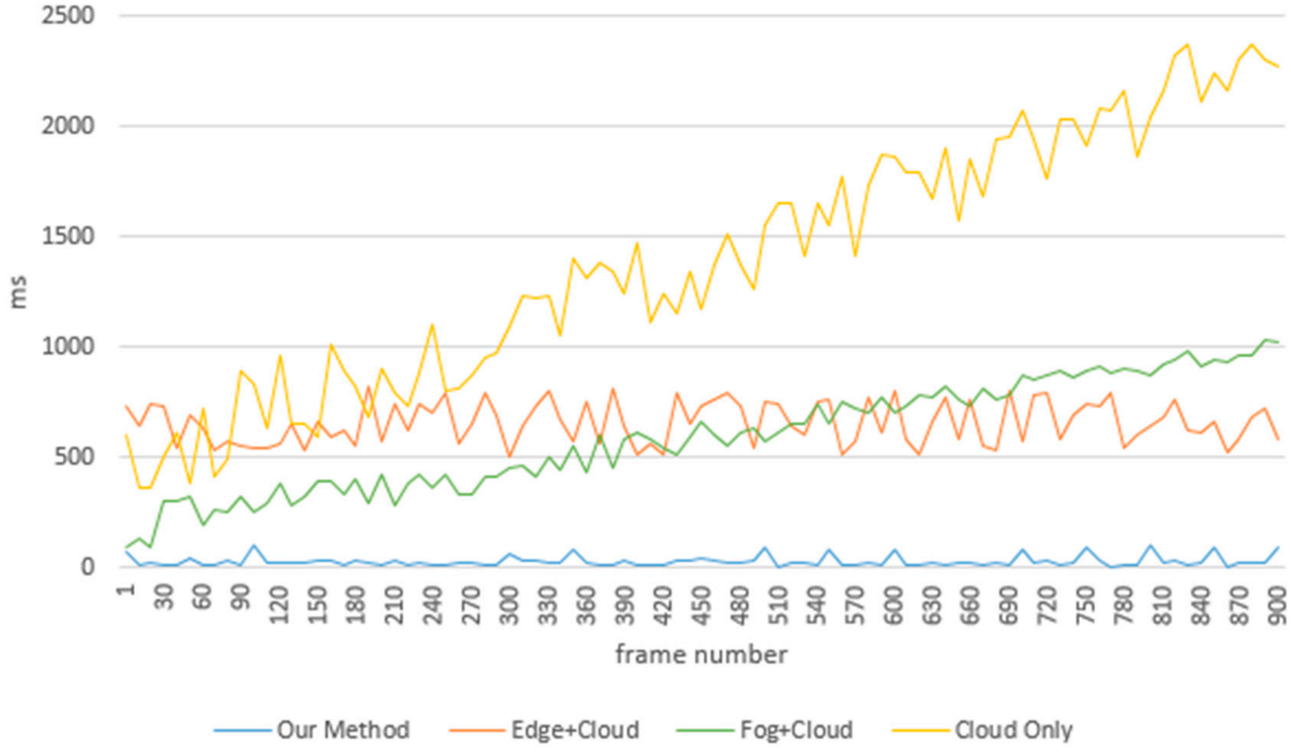

Figure 6. Comparison of the delay time of various approaches.

The proposed method exhibited a very stable pace of user interaction time. With an average delay of less than 100 ms for touch events, users could comfortably engage with the content at 15 fps contents. When an edge computer was involved, it reduced the amount of data transmitted to the cloud significantly, resulting in a consistent response time. The method utilizing both edge and cloud exhibited a slight delay in video playback because it used real-time streaming for rendering. For touch detection, however, it received compressed data, maintaining a consistent speed. On the other hand, when edge computing was not used, the data had to be received and processed from multiple LiDAR sensors over the network. As a result, the transmitted data accumulated over time, leading to progressively slower responsiveness. The same phenomenon was observed when using the cloud only, and it slowed down even faster than with the fog due to slower network latency.

In this section, we evaluated the performance of the proposed method in a testbed environment. We compared our system with various system architectures and performances in existing systems that utilize the functionality of commercial game engines. Our system processes a substantial amount of point cloud data received from LiDAR sensors located on the six walls. Handling such large-scale data processing is typically challenging in cloud environments. However, we efficiently addressed this issue by employing edge computing, which involves data compression. Our experiments demonstrated that our system achieved an average data compression rate of approximately 70–80%, minimizing network latency between cloud and fog and allowing for the transmission of smaller data sizes. This reduction in data transfer delays significantly improves real-time interaction, enhancing the user experience in XR environments.

Furthermore, we conducted a comparative analysis between fog servers and central servers. In our proposal, fog servers share the same logic as central servers but do not retain data. Instead, they fetch data from the central server when needed. This approach minimizes network latency and effectively resolves time synchronization issues among users. As a result, this configuration provides a clear advantage in response time compared to traditional cloud-based systems, optimizing user interactions, and delivering an improved user experience.

Additionally, our system is designed to perform GPU-intensive tasks, particularly screen rendering, exclusively on edge computers. This design ensures that graphics-related tasks, which demand GPU capabilities, are offloaded to the edge, while CPU-based operations, which do not require graphics processing, are carried out within the fog and cloud servers. This efficient division of labor reduces the cost of cloud infrastructure setup and maintenance and effectively distributes the load in online XR environments. As a result, our system offers a more immersive and realistic virtual reality experience.

The proposed method used edge computing for the efficient local control of sensors and displays and used fog computing for the rapid processing of user interactions with feedback, synchronizing with the cloud later, enabling faster and more realistic interactions.

5. Conclusions

This paper proposed and experimentally validated a novel approach to enhance real-time user interactions in XR environments. The proposed system distributes data processing and rendering tasks from the cloud to local environments, leveraging edge and fog computing. The experimental results showed that data compression and processing using edge computers mitigated the rendering delays significantly, highlighting the crucial role of fog systems in time synchronization and data synchronization for user interaction. These findings represent significant advances in ensuring excellent performance and real-time interactions in XR environments [6].

Furthermore, data processing in cloud environments and network response delays can lead to increased latency, emphasizing the that leveraging edge and fog computing to address these issues is essential. Consequently, the proposed system has great potential for enhancing user experiences in XR environments, and its application and development in various systems will lead to further progress and expansion in future research and development endeavors.

Author Contributions: Conceptualization, E.-S.L. and B.-S.S.; methodology, E.-S.L.; software, E.-S.L.; validation, E.-S.L. and B.-S.S.; formal analysis, B.-S.S.; investigation, E.-S.L.; resources, E.-S.L.; data curation, B.-S.S.; writing—original draft preparation, E.-S.L.; writing—review and editing, B.-S.S.; visualization, E.-S.L.; supervision, B.-S.S.; project administration, E.-S.L.; funding acquisition, B.-S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the National Research Foundation of Korea (NRF) grants funded by the Korean government (No. NRF-2022R1A2B5B01001553 and No. NRF-2022R1A4A1033549).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the license of contents.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yin, Y.; Zheng, P.; Li, C.; Wang, L. A State-of-the-Art Survey on Augmented Reality-Assisted Digital Twin for Futuristic Human-Centric Industry Transformation. *Robot. Comput.-Integr. Manuf.* **2023**, *81*, 102515. [CrossRef]
2. Heo, M.-H.; Kim, D. Effect of Augmented Reality Affordance on Motor Performance: In the Sport Climbing. *Hum.-Centric Comput. Inf. Sci.* **2021**, *11*, 40. [CrossRef]
3. Zhang, F.; Wu, T.-Y.; Pan, J.-S.; Ding, G.; Li, Z. Human Motion Recognition Based on SVM in VR Art Media Interaction Environment. *Hum.-Centric Comput. Inf. Sci.* **2019**, *9*, 40. [CrossRef]
4. Bang, G.; Yang, J.; Oh, K.; Ko, I. Interactive Experience Room Using Infrared Sensors and User's Poses. *J. Inf. Process. Syst.* **2017**, *13*, 876–892. [CrossRef]
5. Lee, Y.; Yoo, B.; Lee, S.-H. Sharing Ambient Objects Using Real-Time Point Cloud Streaming in Web-Based XR Remote Collaboration. In Proceedings of the 26th International Conference on 3D Web Technology, Pisa, Italy, 8–12 November; Association for Computing Machinery: New York, NY, USA, 2021; pp. 1–9.
6. Theodoropoulos, T.; Makris, A.; Boudi, A.; Taleb, T.; Herzog, U.; Rosa, L.; Cordeiro, L.; Tserpes, K.; Spatafora, E.; Romussi, A.; et al. Theodoros Theodoropoulos Cloud-Based XR Services: A Survey on Relevant Challenges and Enabling Technologies. *J. Netw. Netw. Appl.* **2022**, *2*, 1–22. [CrossRef]
7. Liubogoshchev, M.; Ragimova, K.; Lyakhov, A.; Tang, S.; Khorov, E. Adaptive Cloud-Based Extended Reality: Modeling and Optimization. *IEEE Access* **2021**, *9*, 35287–35299. [CrossRef]
8. Wu, H.; Zhang, H.; Cheng, J.; Guo, J.; Chen, W. Perspectives on Point Cloud-Based 3D Scene Modeling and XR Presentation within the Cloud-Edge-Client Architecture. *Vis. Inform.* **2023**, *7*, 59–64. [CrossRef]
9. Huang, Y.; Song, X.; Ye, F.; Yang, Y.; Li, X. Fair and Efficient Caching Algorithms and Strategies for Peer Data Sharing in Pervasive Edge Computing Environments. *IEEE Trans. Mob. Comput.* **2020**, *19*, 852–864. [CrossRef]
10. Boletsis, C.; Cedergren, J.E. VR Locomotion in the New Era of Virtual Reality: An Empirical Comparison of Prevalent Techniques. *Adv. Hum.-Comput. Interact.* **2019**, *2019*, e7420781. [CrossRef]
11. AR/VR Light Engines: Perspectives and Challenges. Available online: https://opg.optica.org/aop/abstract.cfm?uri=aop-14-4-783 (accessed on 1 October 2023).
12. Chiang, F.-K.; Shang, X.; Qiao, L. Augmented Reality in Vocational Training: A Systematic Review of Research and Applications. *Comput. Hum. Behav.* **2022**, *129*, 107125. [CrossRef]
13. Çöltekin, A.; Lochhead, I.; Madden, M.; Christophe, S.; Devaux, A.; Pettit, C.; Lock, O.; Shukla, S.; Herman, L.; Stachoň, Z.; et al. Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 439. [CrossRef]
14. Erol-Kantarci, M.; Sukhmani, S. Caching and Computing at the Edge for Mobile Augmented Reality and Virtual Reality (AR/VR) in 5G. In *Ad Hoc Networks, Proceedings of the 9th International Conference, AdHocNets 2017, Niagara Falls, ON, Canada, 28–29 September 2017*; Zhou, Y., Kunz, T., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 169–177.
15. Pan, J.; McElhannon, J. Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet Things J.* **2018**, *5*, 439–449. [CrossRef]
16. He, S.; Ren, J.; Wang, J.; Huang, Y.; Zhang, Y.; Zhuang, W.; Shen, S. Cloud-Edge Coordinated Processing: Low-Latency Multicasting Transmission. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1144–1158. [CrossRef]
17. Elbamby, M.S.; Perfecto, C.; Bennis, M.; Doppler, K. Toward Low-Latency and Ultra-Reliable Virtual Reality. *IEEE Netw.* **2018**, *32*, 78–84. [CrossRef]
18. Zhou, Y.; Pan, C.; Yeoh, P.L.; Wang, K.; Elkashlan, M.; Vucetic, B.; Li, Y. Communication-and-Computing Latency Minimization for UAV-Enabled Virtual Reality Delivery Systems. *IEEE Trans. Commun.* **2021**, *69*, 1723–1735. [CrossRef]
19. Fu, C.; Li, G.; Song, R.; Gao, W.; Liu, S. OctAttention: Octree-Based Large-Scale Contexts Model for Point Cloud Compression. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 625–633. [CrossRef]

20. Kammerl, J.; Blodow, N.; Rusu, R.B.; Gedikli, S.; Beetz, M.; Steinbach, E. Real-Time Compression of Point Cloud Streams. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 778–785.
21. Tu, C.; Takeuchi, E.; Miyajima, C.; Takeda, K. Compressing Continuous Point Cloud Data Using Image Compression Methods. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Janeiro, Brazil, 1–4 November 2016; pp. 1712–1719.
22. Banfi, F.; Previtali, M. Human–Computer Interaction Based on Scan-to-BIM Models, Digital Photogrammetry, Visual Programming Language and eXtended Reality (XR). *Appl. Sci.* **2021**, *11*, 6109. [CrossRef]
23. Bamodu, O.; Ye, X.M. Virtual Reality and Virtual Reality System Components. *Adv. Mater. Res.* **2013**, *765–767*, 1169–1172. [CrossRef]
24. Kuhlen, T.; Dohle, C. Virtual Reality for Physically Disabled People. *Comput. Biol. Med.* **1995**, *25*, 205–211. [CrossRef]
25. Yew, H.T.; Ng, M.F.; Ping, S.Z.; Chung, S.K.; Chekima, A.; Dargham, J.A. IoT Based Real-Time Remote Patient Monitoring System. In Proceedings of the 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), Langkawi, Malaysia, 28–29 February 2020; pp. 176–179.
26. Srikanth, S.; Meher, S. Compression Efficiency for Combining Different Embedded Image Compression Techniques with Huffman Encoding. In Proceedings of the 2013 International Conference on Communication and Signal Processing, Melmaruvathur, India, 3–5 April 2013; pp. 816–820.
27. Mansri, I.; Doghmane, N.; Kouadria, N.; Harize, S.; Bekhouch, A. Comparative Evaluation of VVC, HEVC, H.264, AV1, and VP9 Encoders for Low-Delay Video Applications. In Proceedings of the 2020 Fourth International Conference on Multimedia Computing, Networking and Applications (MCNA), Valencia, Spain, 19–22 October 2020; pp. 38–43.
28. Li, Y.; Zhao, C.; Tang, X.; Cai, W.; Liu, X.; Wang, G.; Gong, X. Towards Minimizing Resource Usage with QoS Guarantee in Cloud Gaming. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 426–440. [CrossRef]
29. Lee, J.-H.; Lee, G.-M.; Park, S.-Y. Calibration of VLP-16 Lidar Sensor and Vision Cameras Using the Center Coordinates of a Spherical Object. *KIPS Trans. Softw. Data Eng.* **2019**, *8*, 89–96.
30. Song, W.; Zhang, L.; Tian, Y.; Fong, S.; Liu, J.; Gozho, A. CNN-Based 3D Object Classification Using Hough Space of LiDAR Point Clouds. *Hum.-Centric Comput. Inf. Sci.* **2020**, *10*, 19. [CrossRef]
31. Heuchert, S.; Rimal, B.P.; Reisslein, M.; Wang, Y. Design of a Small-Scale and Failure-Resistant IaaS Cloud Using OpenStack. *Appl. Comput. Inform.* 2021; *ahead-of-print*. [CrossRef]
32. Unreal Engine. Available online: https://www.unrealengine.com/ko/unreal-engine-5?gclid=Cj0KCQjw1aOpBhCOARIsACXYv-drJT2jlSeepwKq23R3R0Z_ytbHwFsbQ48UeEdRqxZCGn-ertYdOs0aAiOeEALw_wcB (accessed on 14 October 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.