Application for Advanced Video Conferencing in 3D Space

Maksym Alieksieienko, Radoslav Vargic
Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology, Bratislava, Slovakia
E-mail address: xalieksieienko@stuba.sk

Abstract - **This paper presents a novel approach to video conferencing by introducing immersive 3D environments that extend communication beyond traditional 2D interfaces. In response to the limitations of traditional platforms, we propose a browser-based solution that combines spatial visualisation with real-time video streaming. The system uses Gaussian Splatting and Three.js to render realistic 3D environments, and integrates the Agora SDK for efficient video communication. Our architecture uses React.js for the frontend, Quarkus for the backend, and a MySQL database for user management. Through iterative prototyping and validation, we have demonstrated that this approach significantly improves user engagement and presence without requiring specialised hardware. The solution combines the accessibility of a web-based application with the immersive qualities of a 3D environment, addressing a critical gap in today's video conferencing technologies.**

Keywords - Video conferencing; 3D environments; Gaussian Splatting; Three.js; Real-time communication; React.js; Quarkus

I. INTRODUCTION

The rapid evolution of digital communication has fundamentally transformed the way people work, study, and socialize. The outbreak of the COVID-19 pandemic accelerated the global dependence on videoconferencing technologies, revealing both their strengths and limitations. While platforms such as Zoom, Microsoft Teams, and Google Meet have provided accessible and practical solutions for online meetings, they predominantly rely on two-dimensional interfaces. These 2D environments, although sufficient for basic interactions, fail to deliver a sense of spatial immersion, physical presence, or dynamic interactivity.

In professional, educational, and entertainment contexts, users are increasingly seeking more engaging and realistic virtual experiences. As such, the limitations of traditional video conferencing—lack of shared spatial awareness, flat representations, and non-interactive elements—have created an opportunity for rethinking digital meetings through the lens of 3D environments. This work introduces a novel approach to video conferencing using modern 3D rendering techniques and real-time communication frameworks, aiming to bridge the gap between virtual interaction and physical presence.

II. GOAL OF THE WORK

The primary objective of this work is to develop an innovative, browser-based video conferencing system that leverages immersive 3D environments to enhance the communication experience. The proposed platform aims to combine spatial visualization with real-time video streaming, allowing users to interact in virtual rooms as if physically co-located. Current videoconferencing solutions offer limited realism and often ignore the spatial context of communication. This research proposes a system that addresses this deficiency by implementing 3D virtual spaces rendered using Gaussian Splatting and Three.js, with video feeds embedded as live textures within the environment. The ultimate goal is to deliver a practical, efficient, and accessible solution that operates on standard consumer hardware and browsers, removing the dependency on VR headsets or expensive equipment. In achieving this, the work defines the following sub-goals:

- Analyze the strengths and weaknesses of existing 2D and 3D video conferencing platforms.
- Design a scalable architecture with clear separation of concerns.
- Implement video streaming onto 3D surfaces in real-time using Agora SDK.
- Ensure responsive performance across various devices, minimizing computational demands.
- Validate usability through performance tests and iterative prototype development.

III. WORK METHODOLOGY AND RESEARCH METHODS

The development process combined analysis, prototyping, and experimental validation. The methodology involved:

A. Reviewing existing videoconferencing technologies (2D and 3D) to identify current limitations.

The first phase of the project involved a comprehensive review of existing videoconferencing platforms, both 2D and 3D, to identify their strengths and limitations.

2D platforms such as Zoom, Microsoft Teams, Google Meet, and Cisco Webex offer simple, widely accessible video communication tools with features like screen sharing and recording. However, they lack spatial context and immersive interaction, which limits the overall user experience, especially in collaborative or educational environments.

3D platforms such as Mozilla Hubs, Spatial, and Engage VR offer enhanced realism and support for virtual reality (VR). These systems allow users to interact in shared virtual spaces, manipulate objects, and simulate physical environments. Nevertheless, they often require specialized hardware (e.g., VR headsets), are computationally intensive, and present challenges in accessibility and ease of use.

B. Selecting suitable tools and frameworks: React.js for frontend, Quarkus for backend, Agora SDK for real-time video communication, and Gaussian Splatting with Three.js for 3D rendering.

To build an innovative and scalable videoconferencing solution, a modern technology stack was selected:

- **React.js** was chosen for frontend development due to its component-based architecture, strong community support, and suitability for building interactive user interfaces.
- **Quarkus** was selected for the backend layer for its lightweight runtime, native compilation capabilities via GraalVM, and fast startup times—crucial for real-time applications.
- **Agora SDK** was integrated to manage real-time video and audio streaming. It offloads heavy computation to Agora's servers and provides a stable and scalable video infrastructure.
- **Three.js combined with Gaussian Splatting** was used to render realistic 3D environments in the browser. Gaussian Splatting allows for the creation of high-fidelity, volumetric 3D scenes from image data, while Three.js enables dynamic rendering and interaction with 3D elements.

C. Designing system architecture using a layered approach with clear separation between database, backend, and frontend.

The system was designed using a layered architecture to ensure modularity, scalability, and ease of maintenance. The architecture consists of the following layers:

- **Frontend** (React.js) – Manages the user interface and client-side logic. This includes rendering the 3D environment, handling user interactions, and integrating video textures onto 3D objects. As shown in Fig. 1, the virtual room provides an immersive 3D environment where participants can interact during video conferences, shown here from a distant perspective.

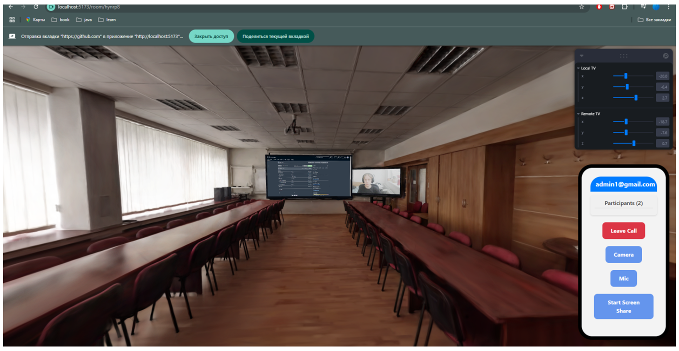

Figure 1. Virtual Room (distant view)

- **Backend** (Quarkus) – Responsible for authentication, API routing, business logic, and communication with the database. JSON Web Tokens (JWT) are used for secure user sessions.
- **Database** (MySQL) – Stores user data, authentication credentials, and metadata for video sessions.
- **Real-Time Video Layer** (Agora SDK) – Handles video stream management and user connectivity in real time.

This architecture ensures separation of concerns, enabling independent development and scaling of different system components. Fig. 2 illustrates the complete system architecture with clearly defined layers and communication patterns.

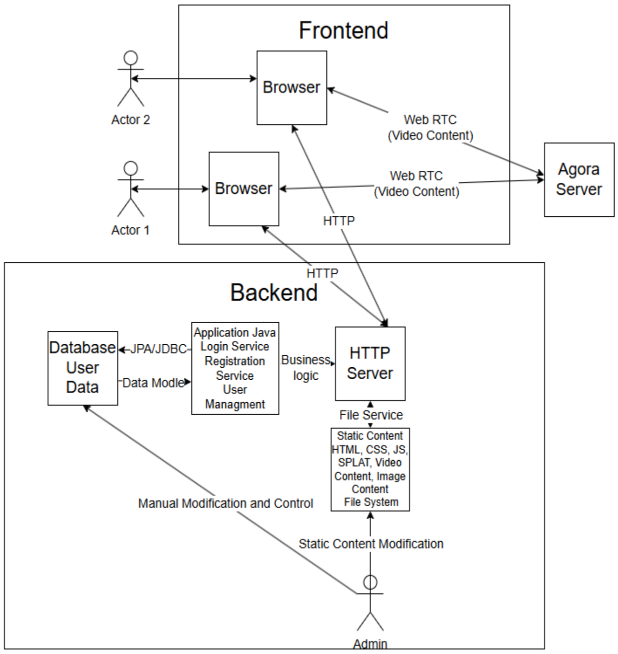

Figure 2. System architecture

D. Creating prototypes of user registration, authentication flows, video streaming, and 3D environment rendering.

Multiple prototypes were developed and iteratively tested to validate core system components:

- **User Registration and Authentication**: A registration flow was implemented using a MySQL database for user data storage and JWT for secure login and session management. The login interface, depicted in Fig. 3, provides secure authentication while maintaining a user-friendly design.

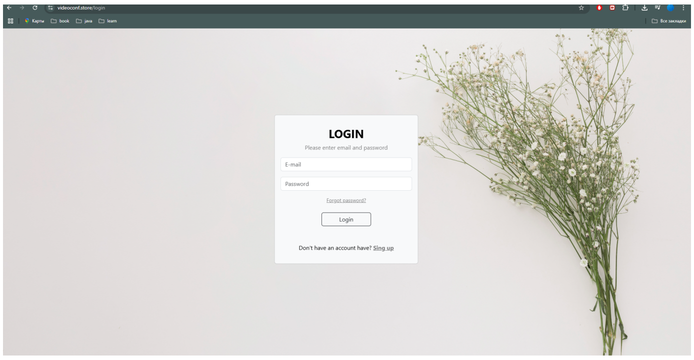

Figure 3. Login page

- **Video Streaming**: A working prototype was created using Agora SDK, allowing users to join a virtual meeting and stream video. These video feeds were mapped as textures onto 3D objects like monitors and screens inside the virtual environment.
- **3D Rendering and Interaction**: A virtual 3D space was designed using Three.js and Gaussian Splatting techniques. Several libraries were evaluated (Luma AI, Zappar, mkkellogg), with mkkellogg's library ultimately chosen for its stability and compatibility. Fig. 4 presents a close-up view of the virtual room, demonstrating the detailed rendering of the conferencing environment and how video feeds are integrated onto surfaces within the 3D space.

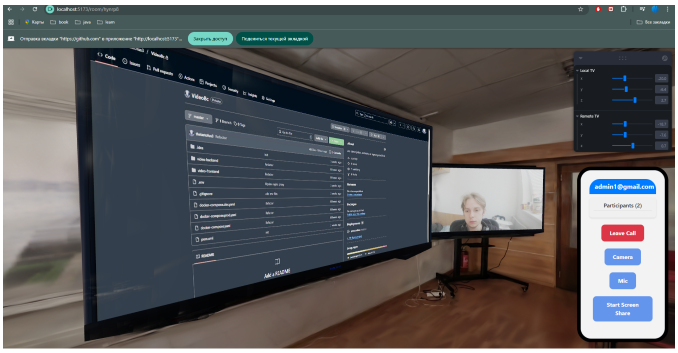

Figure 4. Virtual room (close-up view)

IV. PROPOSED SOLUTION

The proposed system features a three-layered architecture:

- **Database Layer:** A MySQL database ensures secure user management with unique constraints and hashed passwords.
- **Backend Layer:** A Java-based Quarkus backend provides RESTful APIs, JWT-based authentication, and secure data transactions.
- **Frontend Layer:** A React.js frontend implements user interfaces, authentication screens, 3D space rendering, and Agora-based video streaming. The registration page shown in Fig. 5 collects necessary user credentials while maintaining simplicity and clarity.

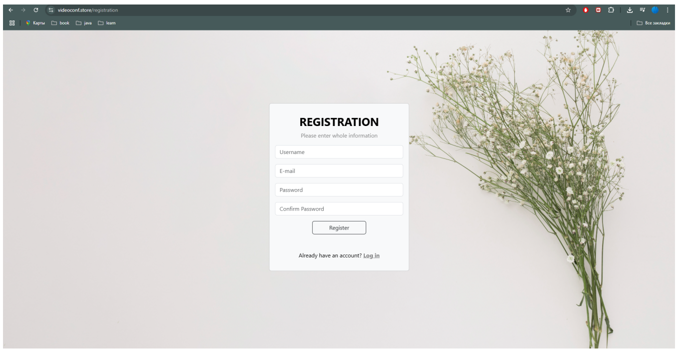

Figure 5. Registration Page

The system supports both predefined virtual rooms (using nanoid identifiers) and custom room creation, enabling flexible and private conferencing options. As demonstrated in Fig. 6, users can select from available virtual rooms or create custom spaces for their specific conferencing needs. The 3D environments are generated using Gaussian Splatting techniques integrated with Three.js, allowing realistic scene creation with manageable computational loads. Real-time communication is handled by Agora's cloud services, optimizing latency and scalability.

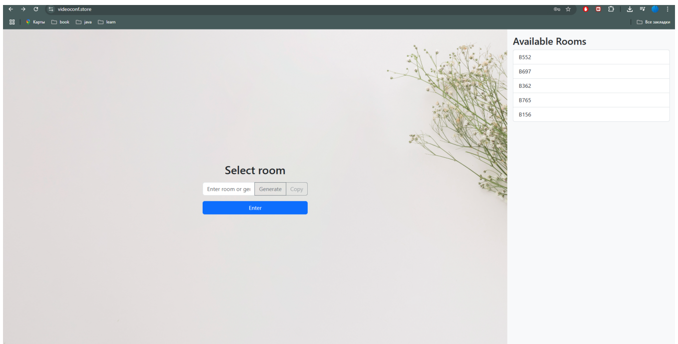

Figure 6. Room selection

V. Benefits

The developed solution offers numerous advantages:

- **Enhanced User Engagement:** Realistic 3D environments foster a greater sense of presence and collaboration.
- **Accessibility:** The system is optimized for browser use without requiring VR headsets or high-end hardware.
- **Scalability:** Agora's infrastructure ensures smooth performance even with large numbers of participants**.**
- **Security:** JWT authentication and role-based access control guarantee user data safety.
- **Flexibility:** nanoid-based room management simplifies creation of private or public sessions without server-side complexity.
- **Future Expansion:** Modular design allows easy integration of additional features like AI support, interactive objects, and multimedia sharing.

Conclusion

This work explores the potential of immersive 3D environments in video conferencing applications, demonstrating how modern web technologies can overcome the limitations of 2D interfaces. By combining Gaussian Splatting for realistic environment rendering, Three.js for 3D interaction, React for responsive UIs, and Agora SDK for real-time media transmission, a functional prototype was developed and tested. The proposed solution significantly improves user engagement, interactivity, and realism without relying on expensive or specialized equipment. The architecture ensures modularity and scalability, setting a foundation for continuous development. *Key contributions of this work include:*

- Integration of 3D scene rendering with live video feeds.
- Cross-platform, browser-based implementation.
- Real-world usability tested through prototype iterations**.**

The findings suggest that the use of volumetric rendering techniques and real-time video streaming can be efficiently combined to create next-generation conferencing systems. Further work may focus on optimizing system performance, adding more user interaction features, and refining UI/UX based on user feedback.

Acknowledgement

This paper was supported by DISIC (09I05-03-V2), NEXT (ERASMUS-EDU-2023-CBHE-STRAND-2), and InteRViR (VEGA 1/0605/23).

References

[1] Oracle Corporation. *Java SE Documentation*. [Online]. Available: https://docs.oracle.com/javase/

[2] Mozilla Foundation. *MDN Web Docs: JavaScript*. [Online]. Available: https://developer.mozilla.org/en/docs/Web/JavaScript

[3] Quarkus Framework. *Supersonic Subatomic Java*. [Online]. Available: https://quarkus.io

[4] Meta Platforms, Inc. *React: A JavaScript Library for Building User Interfaces*. [Online]. Available: https://react.dev/

[5] Three.js Authors. *Three.js JavaScript 3D Library Documentation*. [Online]. Available: https://threejs.org

[6] M. Kellogg. *Gaussian Splats 3D*. [Online]. Available: https://github.com/mkkellogg/GaussianSplats3D

[7] Agora.io. *Agora SDK for Real-Time Video and Audio*. [Online]. Available: https://www.agora.io/en

[8] WebRTC Project. *WebRTC Protocol Overview*. [Online]. Available: https://webrtc.org