Article

Integration and Optimization Strategy of Spatial Video Technology in Virtual Reality Platform

Da Xu [1,*]

[1] Video Infra, Meta, Menlo Park, CA, 94025, USA
* Correspondence: Da Xu, Video Infra, Meta, Menlo Park, CA, 94025, USA

Abstract: With the rapid development of virtual reality technology, spatial videos have gradually become an important carrier for content presentation on VR platforms due to their immersive, multi perspective, and realistic features. This article mainly conducts preliminary theoretical research on the integrated application of spatial video in virtual reality platforms. Summarized the theoretical basis and technological development of spatial video in virtual reality, and constructed a comprehensive system model for the collection, encoding, rendering, and interaction of spatial video content. Further, aiming at the key problems in data processing pressure, rendering delay and terminal compatibility existing in the current space video application process, this paper proposes optimization solutions in edge computing collaboration, intelligent coding compression and standard interface design. I hope that this research can provide theoretical guidance and specific methods for the integration of spatial video and virtual reality systems.

Keywords: spatial video; virtual reality; system integration

Received: 15 November 2025
Revised: 29 November 2025
Accepted: 02 December 2025
Published: 08 December 2025

1. Introduction

With the development of informatization, virtual reality (VR) has become the foundation of digital interactive applications, covering multiple fields such as education and training, culture and art, travel and transportation, industrial simulation, etc. In addition to its immersive and highly interactive features, spatial video technology has gradually become the main means of virtual modeling. However, in the application process, there are certain key technical difficulties in integrating spatial video into virtual reality platforms, such as the real-time processing requirements of large-scale data, the decrease in experience caused by rendering delays, and the instability of the system caused by insufficient compatibility between different platforms, which greatly restrict the practical application and experience of spatial video technology. Therefore, this paper aims to build the framework of the spatial video integration application system, analyze the current mainstream technical difficulties, and propose optimization strategies such as edge computing collaboration, intelligent compression coding, standard interface adaptation, hoping to help the application practice of the spatial video integration virtual reality platform, and provide a feasible technical route in theory and practice.

2. Theoretical Overview of the Integration of Spatial Video Technology and Virtual Reality

In recent years, Volumetric Video technology has become increasingly popular as a new multidimensional digital imaging technology that achieves stereoscopic recording and immersive presentation of real dynamic scenes through video capture and 3D modeling reconstruction from multiple directions. Compared to regular videos, this type of spatial video can not only capture 2D image data, but also record more information such as the 3D shape and depth changes of objects, giving viewers a "free perspective" and "moving position" in virtual space, and providing a more physically perceptible reality foundation for virtual reality environments [1].

Virtual reality platforms are committed to achieving immersive interactive experiences through computer-generated 3D environments. Its working principle relies on the collaborative work of rendering engines, motion capture, head mounted devices, and input controllers. The introduction of spatial videos greatly enriches the information content of VR platforms, allowing the display content of VR platforms to exceed preset scenes and enable real-time interaction with dynamically reconstructed objects in reality. This not only increases immersion, but also expands the use of VR technology in areas such as education and training, cultural tourism, remote healthcare, virtual social networks, and more [2].

From a theoretical perspective, the integration of spatial video and VR platforms is a complex engineering project that covers various stages such as video shooting, 3D modeling, encoding and decoding, network transmission, graphics rendering, and human-computer interaction [3]. The stability and efficiency in these links will directly affect the coherence and authenticity of the end-user experience. Therefore, it is necessary to build a technology integration model that covers all aspects to achieve standardized integration structures for different device platforms and content formats. This theoretical system is not only the foundation for promoting technology implementation, but also the key support for the future virtual reality industry to move towards the "super immersive content era".

3. Construction of an Integrated Framework for Spatial Videos in Virtual Reality Platforms

3.1. Framework of Spatial Video Capture and Encoding Technology

Space video capture uses various fusion methods such as multi view synchronized camera systems, depth cameras, structured light and light field technologies to capture dynamic images and 3D information of target environments or people. The acquisition process mainly involves fusing RGB images with depth data to achieve real-time reconstruction of point clouds or grid models, resulting in a dynamic 3D video data stream with spatial structure and time series characteristics. Considering the high requirements of VR applications for data volume and processing efficiency, it is necessary to compress and encode the collected spatial data in order to convert it into a content format with high conversion efficiency and transportability [4].

The current mainstream encoding schemes include video based point cloud compression (V-PCC), extended coding AVC/HEVC, and the MPEG-I spatial video standard. These algorithms generally use motion compensation, geometric layer prediction, residual compression, and other methods to encode point clouds or voxels layer by layer, thereby reducing redundancy and ensuring visual quality. At the same time, local data priority loading is required for user perspective prediction to accelerate transmission rate and rendering quality [5]. Because the front-end is mainly responsible for data collection and encoding, it has a direct impact on the adaptation and rendering quality of the back-end virtual reality devices. Rendering performance and smooth interaction are key factors in achieving high-quality integration of spatial videos (As shown in Figure 1).

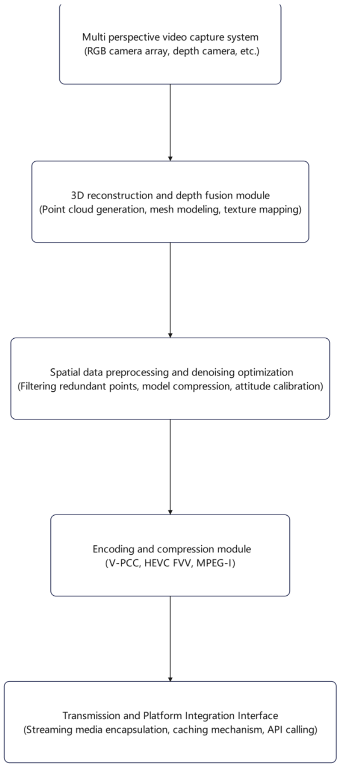

Figure 1. Framework diagram of spatial video capture and encoding technology.

3.2. Virtual Reality Platform Supports Structural Design

To achieve efficient support for spatial video content on virtual reality platforms, a comprehensive support structure covering hardware capabilities, system architecture, and interaction mechanisms needs to be constructed. Firstly, the platform should be equipped with high-performance central processing units (CPUs) and graphics processing units (GPUs) that support multi-threaded computing, and integrated with low latency display modules and high-precision motion tracking devices to achieve support for online rendering and real-time interactive spatial videos. Secondly, the system should establish a decoding module that can read various types of spatial videos, support various 3D data formats such as point clouds, voxels, and grids, and introduce buffer mechanisms to improve data processing efficiency [6].

At the rendering level, parallel graphics pipelines, multi-threaded workload allocation, and partitioned rendering methods should be used to improve user experience, and a "focus first" approach should be adopted to process user perspectives in order to reduce overall computational burden. In terms of interaction support, the platform should have 6DoF spatial positioning capability, gesture recognition capability, and multimodal input support to ensure consistency between user operations and virtual environment response. In addition, the platform also provides standardized interfaces and cross engine adaptation plugins to enhance compatibility with mainstream rendering engines such as Unity and Unreal. The design of these architectures aims to ensure smooth loading, stable operation, and immersive interaction of spatial video content in VR systems, and is also an essential solid foundation for the practical application of technology.

3.3. Integrated System Collaboration Mechanism

The smooth operation of spatial videos in virtual reality platforms relies on the efficient collaboration of five subsystems: capture, encoding, transmission, rendering, and interaction. Each module needs to effectively implement data channels and task scheduling to ensure that multi-dimensional video data can respond to user operations in real time. From the perspective of system architecture, preprocessing work (such as denoising, compression transcoding, etc.) can be carried out at distributed edge nodes, and the data stream can be split and transmitted to the central body for unified customized rendering and scheduling. In addition, a dynamic scheduling mechanism based on QoS (Quality of Service) dynamic policies is used in the transmission process to detect network conditions in real time and adjust transmission rates in a timely manner, ensuring that latency is controlled within an acceptable threshold range for VR interaction.

In terms of interaction, user head movements, gesture operations, and other operations rely on high-speed feedback mechanisms to drive the rendering engine to make corresponding image updates. The system requires a parallel allocation structure with multiple processes to maintain consistency in the processing of each module. To prevent information isolation in the field, a universal control interface and protocol bridging layer are deployed to achieve seamless connection and content synchronization between multiple terminals.

4. Issues Faced by Spatial Video Technology in the Integration of Virtual Reality Platforms

4.1. Sudden Increase in Data Processing and Storage Pressure

Space video data has the characteristics of high dimensionality, high resolution, and multi-channel. The amount of information generated during the collection and rendering process is much larger than that of ordinary videos. The standard recording based on 60 frames per second and multiple viewpoints per frame is still extremely large even after data compression, causing huge pressure on the backend system in storage, scheduling, and processing. In a comprehensive virtual reality environment, users frequently switch perspectives, requiring constant adjustment of data information from different viewpoints, which further increases the I/O burden on the entire system. Many existing platforms do not provide dedicated storage optimization strategies for spatial videos, often leading to system processing congestion and loading delays (see Table 1).

Table 1. Comparison of Data Processing and Storage Requirements between Spatial Video and Traditional Video.

Indicator project	Space video	Traditional video
data dimension	Multi perspective + depth + time	Single perspective + time
Raw data rate	≥ 1.5 Gbps	10-50 Mbps
Average data volume after encoding	Approximately 3-5 GB per minute	Approximately 50-150 MB per minute
Processing delay tolerance	≤ 50 ms (high requirement for real-time interaction)	≤ 200 ms (can be cached and played)
Storage system architecture requirements	Combination of distributed + cache + local storage	Centralized or CDN caching is sufficient

4.2. Rendering Latency and Poor Interactive Experience

Real-time rendering of spatial videos in virtual reality environments requires rapid system responses to user head movements, visual transitions, and interactive actions. This places extremely high demands on rendering speed and system responsiveness. When system performance falls below the required threshold, issues such as frame stuttering, synchronization discrepancies, and motion-induced dizziness may occur. These problems mainly stem from complex data decoding, high rendering resource consumption, and imperfect synchronization mechanisms—challenges that become more pronounced under conditions such as multi-user concurrent access or limited performance on mobile devices.

In contrast, traditional videos exhibit much higher fault tolerance because their rendering targets are fixed, and no viewpoint transformations are involved (see Table 2).

Table 2. Comparison of rendering and interactive response between spatial video and traditional video.

Indicator project	Space video	Traditional video
Render Target	Multi perspective + 3D dynamic scene	Single screen 2D sequence
Real time interactive requirements	High: Need to dynamically respond to user behavior	Low: basically passive viewing
Delay tolerance threshold	≤ 20 ms (to prevent motion sickness)	≤150 ms
Processing mechanism	Parallel rendering + GPU acceleration + prediction algorithm	Sequential rendering + buffering playback
User experience dependency	Extremely high: significant delay affects immersion	Medium: Short term delay has little impact

4.3. Poor Terminal Compatibility and Platform Adaptation Barriers

For virtual reality terminal devices, there are strict requirements regarding processing speed, computing power, and latency. Currently, mainstream VR headsets on the market differ significantly in processor performance, storage capacity, and image rendering capabilities. As a result, some devices cannot smoothly decode the high-dimensional and high-bandwidth data required for spatial videos. In addition, due to the absence of unified standards for video formats, communication protocols, rendering engines, and control interfaces, it is difficult to ensure that spatial video content can deliver a consistent experience across multiple platforms. These compatibility differences limit the dissemination of spatial video content and increase the development cost associated with adapting to diverse devices (see Table 3).

Table 3. Comparison of Spatial Video Adaptation Capability of Different VR Terminals.

Device Type	Graphics rendering capability	Support spatial video formats	Decoding efficiency	Platform compatibility	Does it support 6DoF
High end PC VR devices	High	full support	High	good	support
Integrated VR headset	medium	partially supported	medium	general	Support some functions
Mobile VR glasses	low	Basically not supported	low	poor	Not Supported

5. Optimization Strategy of Spatial Video Technology in Virtual Reality Platform Integration

5.1. Edge Computing Collaboration and Cloud Rendering Architecture Optimization

In order to meet the challenges of high computing and low latency brought by space video in virtual reality platform, building edge computing collaboration and cloud rendering architecture has become one of the key optimization directions of virtual reality and space video. Edge nodes are deployed close to user terminals and can locally complete preliminary decoding, viewpoint prediction, and cache pre allocation of spatial videos, reducing the amount and latency of data transmission back to the cloud. In addition, the cloud can handle large-scale point cloud rendering and content synchronization processing, achieving unified resource management and multi-party device linkage.

Taking a 5G+VR museum exhibition project as an example, edge nodes were deployed in the spatial video prediction model to predict the next frame's FOV (Field of

View) area based on user perspective movement. The rendering resources were scheduled according to the following formula:

$$R_i = \alpha \cdot D_i + \beta \cdot P_i \quad (1)$$

Among them, R_i represents the resource allocation priority of the current frame, D_i is the depth complexity of the user's gaze area, P_i is the access probability of the predicted viewpoint area, and α, β are scheduling weight coefficients. This model achieves dynamic optimization allocation of resources, reducing the average user latency to 24ms, significantly improving the immersive experience and system stability.

5.2. Intelligent Encoding Compression and Low Latency Rendering Mechanism

The smooth operation of spatial videos in virtual reality platforms highly relies on efficient video compression and low latency image rendering mechanisms. Traditional compression encoding cannot simultaneously meet the goals of spatial structure and real-time performance, which may lead to a decrease in image quality or decoding delay. Therefore, in recent years, many studies have focused on content-based intelligent coding, such as V-PCC (Video based Point Cloud Compression) and AV1's multi-layer prediction based compression coding. This type of algorithm combines the complexity of video content and user interaction behavior to dynamically select compression parameters, minimizing the encoding rate while protecting spatial information. By utilizing Tile based region loading technology and variable rate rendering (VRS) mode in graphic rendering, only the details of the user's current area of interest are rendered, reducing system pressure. For example, in a certain educational VR platform, by prioritizing the rendering of panoramic videos based on user gaze hotspots, the average frame rendering time can be reduced by about 36%. The application of intelligent video compression encoding combined with graphics rendering in virtual reality systems enables high system efficiency and good interactivity, which is the key to spatial video optimization technology.

5.3. Standard Interface Development and Platform Compatibility Framework Construction

To address the challenge of limited interoperability of spatial videos across multiple terminals and platforms, industry consensus emphasizes the establishment of a unified interface protocol and a compatibility framework. At present, the mapping standards among spatial video encoding formats (e.g., MPEG-I, OMAF), interaction protocols (e.g., WebXR), and rendering engines (e.g., Unity, Unreal) remain unclear. Consequently, developers are often required to repeatedly encode content or reconstruct interfaces for different terminal platforms, which significantly increases system integration costs. By implementing a platform-neutral API layer, standardizing spatial video data exchange, and integrating multi-format adapters to enable automatic format conversion and resource-loading optimization, mainstream XR platforms can achieve automatic recognition and adaptation of prevalent encoding and interaction protocols. This approach greatly facilitates content development, deployment, and cross-platform distribution **(see** Table 4**)**.

Table 4. Cross platform compatibility comparison and interface requirement analysis of spatial videos.

platform type	Support mainstream encoding formats	Standardization of interface protocols	Rendering engine compatibility	Does it support spatial video middleware
High end PC VR	Support (MPEG-I, etc.)	Basic standardization	High (Unity/UE)	Support mainstream middleware

Mobile VR all-in-one machine	Support some formats	Low standardization level	medium	Customized adaptation plugins are required
WebVR / WebXR	Need transcoding adaptation	Relatively standardized interface	basic support	Deploying lightweight middleware

6. Conclusion

As an important medium for building immersive virtual reality experiences, spatial video technology is actively breaking through various technical challenges such as data collection, encoding compression, and real-time rendering. This paper mainly studies the integration path of space video in the virtual reality platform, systemically combs the key technical framework, makes special technical discussions on the current technical problems of space video (such as pressure processing, interaction delay, compatibility), and gives a series of corresponding improvement methods, including putting forward edge computing collaboration, intelligent coding compression technology to reduce delay, and building standard interface specification strategies. Research has shown that only by building an effective collaborative work system and a standardized integration mechanism can the applicability and stable application of spatial video on VR platforms be achieved. In the future, emphasis should be placed on the integration of technical standards and the development of collaborative innovation between software and hardware, in order to achieve deeper penetration of this technology into fields such as education, tourism culture, and healthcare, and to generate stronger radiation on more platforms.

References

1. D. Zhao, M. Wang, K. Huang, W. Zhong, P. V. Arun, Y. Li, and H. Zhou, "OCSCNet-Tracker: Hyperspectral Video Tracker Based on Octave Convolution and Spatial-Spectral Capsule Network," *Remote Sensing*, vol. 17, no. 4, p. 693, 2025. doi: 10.3390/rs17040693
2. N. E. Aissaoui, M. S. Azzaz, R. Kaibou, and C. Tanougast, "Efficient FPGA implementation of chaos-based real-time video watermarking system in spatial and DWT domain using QIM technique," *Journal of Real-Time Image Processing*, vol. 22, no. 1, p. 40, 2025. doi: 10.1007/s11554-025-01620-2
3. J. S. Kim, K. W. Kim, H. J. Kim, and S. Y. Moon, "A novel virtual reality-based simulator for maxillofacial reconstruction surgery: development and validation study," *Applied Sciences*, vol. 15, no. 1, p. 332, 2024.
4. P. Sabalette, N. Dubé, P. Ménard, M. Labelle, M. T. Laramée, J. Higgins, and C. Duclos, "Immediate effect of alone and combined virtual reality, gait-like muscle vibration and transcranial direct current stimulation on neuropathic pain after spinal cord injury: a pilot study," *Spinal Cord Series and Cases*, vol. 10, no. 1, p. 83, 2024.
5. K. M. Bhargavi, and N. S. Jayalakshmi, "Kalman filter-based DC bus voltage control for autonomous DC microgrid system with PV, wind and EVs integration," *IETE Journal of Research*, vol. 69, no. 12, pp. 9065-9075, 2023.
6. R. Sahara, S. Abdullah, and M. I. Saputra, "Integration model of academic information systems and learning management systems with REST web services using external databases," *Journal of Information and Organizational Sciences*, vol. 47, no. 2, pp. 375-386, 2023.