Computer Science and Information Systems 22(3):1167–1196 https://doi.org/10.2298/CSIS241122037L

# Three-Dimensional Visualization Design Strategies for Urban Smart Venues under the Internet of Things

Renjun Liu

Wuhan Univerisity, Wuhan, 430010, China
liurenjunlrj@gmail.com

**Abstract.** With the increasing demand for smart venue management and data visualization, existing three-dimensional (3D) visualization technologies face challenges in meeting the requirements for efficient, real-time, and multifunctional data presentation. This study systematically compares and analyzes various 3D visualization methods, exploring their application effectiveness in smart venues to provide a reference for technology selection and optimization. Firstly, based on Building Information Modeling (BIM), Geographic Information System (GIS), and Internet of Things (IoT) technologies, this study delves into the principles and concepts of 3D architectural visualization. Meanwhile, it conducts a comprehensive analysis of common 3D visualization technologies. Secondly, using Cesium rendering technology, the study refines surface data for smart venues and performs detailed comparisons with Digital Twins (DTs), BIM, and Octree technologies. Finally, performance indicators like model response time, rendering speed, and frame rate are evaluated under different environments. The results reveal that in IoT environments, the combination of databases and browsers remarkably affects 3D visualization rendering performance. When using the My Structured Query Language (MySQL) database and the Chrome browser, Cesium achieves the best performance, with a model compression size of 5612 KB. It outperforms Unity (6021 KB), Three.js (5720 KB), and Octree (6754 KB). With the PostgreSQL database and Chrome browser, Cesium demonstrates strong lightweight performance with a model compression size of 13,991 KB. Under varying hardware conditions, rendering speed and response time improve significantly with advancements in processor and Graphics Processing Unit (GPU) performance. For instance, Cesium's rendering speed increases from 24 frames per second (FPS) on a Core i3 processor to 34 FPS on a Core i7 processor. Performance differences are observed among methods in response time, rendering speed, and user interaction experience, with Cesium outperforming others across multiple performance indicators. Overall, Cesium rendering technology demonstrates exceptional performance in 3D visualization for smart venues, surpassing other common 3D visualization technologies. The Cesium-based smart venue visualization system functions effectively, meeting practical requirements and contributing to improved user experience, optimized data presentation, and enhanced venue management.

**Keywords:** Internet of Things, smart venues, Three-Dimensional visualization, building information modeling, Cesium.

# 1. Introduction

## 1.1. Research Background and Motivations

With the emergence and widespread application of digital technologies like Urban Information Modeling (UIM) and the Geographic Information System (GIS), cities are integrating multi-dimensional and multi-scale data. This data is combined into a unified three-dimensional (3D) digital space, called Urban Information Synthesis (UIS). However, as Internet of Things (IoT) technology evolves and urban informatization deepens, traditional two-dimensional (2D) GIS technology proves inadequate when facing the demand for spatial data visualization [1-3].

Traditional 2D GIS technology is a geographic information processing tool based on a planar coordinate system [4]. This technology handles surface spatial information and analyzes spatial data through layer overlay methods. While effective for certain applications, 2D GIS has notable limitations in expressing 3D structures, dynamic real-time display, and elevation attribute representation. These constraints make it ill-suited for the comprehensive, real-time, and 3D requirements of complex urban venues and other architectural spaces [5,6]. As urbanization and informatization progress, the limitations of 2D GIS become increasingly apparent [7]. Faced with dynamic scenes like urban venues, 2D GIS struggles to provide real-time monitoring and dynamic visualization of building changes. It is also inadequate for supporting real-time monitoring and dynamic displays in urban planning [8]. Additionally, regarding geographical spatial data including elevation information, 2D GIS cannot accurately represent buildings and terrain at different heights, leading to errors in urban planning and analysis [9]. These deficiencies hinder the broader application of 2D GIS in addressing the demands of urban intelligence and sustainable development. In contrast, 3D visualization offers an advanced way to display geographic information by presenting a 3D model of geographical data [10]. Common 3D visualization technologies include Virtual Reality (VR), augmented reality, 3D GIS, interactive 3D graphics, and professional 3D modeling and rendering software [11,12]. Compared to traditional 2D GIS, 3D visualization presents several significant advantages [13]. This technology supports dynamic, real-time display, enabling the real-time monitoring of the status changes of buildings like urban venues. It supports dynamic real-time displays, enabling urban managers to monitor changes in buildings, such as those occurring in urban venues, in real-time [14,15].

## 1.2. Research Objectives

In the rapid development of smart venue management and data visualization, existing 3D visualization methods still face many challenges in performance, adaptability, and real-time capabilities. The core objective of this study is to construct a comprehensive evaluation framework to systematically optimize the application of 3D visualization technology in smart venues. Concurrently, it can address the current shortcomings in rendering efficiency, data adaptability, and user interaction. Although the demand for smart venue management and data visualization is growing rapidly, existing 3D visualization methods still face significant challenges in data adaptability, rendering efficiency, and user experience. Current methods exhibit considerable differences in rendering performance across various databases and browser environments, particularly in application scenarios with high data density and real-time computation demands, where stability and efficiency are limited. Moreover, existing evaluation methods mainly focus on traditional indicators such as rendering speed, frame rate, and response time, lacking in-depth consideration of key factors such as resource utilization, system scalability, and user interaction experience. This study proposes a new comprehensive evaluation framework that fills the gaps in current research regarding performance optimization. Also, it provides a feasible optimization solution for 3D visualization technology in IoT environments. The framework introduces adaptive rendering strategies and delves into dynamic data transmission and computational optimization. This enables more stable and efficient rendering performance of 3D visualization in smart venues across different technological environments. This study first constructs a comprehensive evaluation framework to optimize the 3D visualization methods for smart venues. Moreover, based on existing research, it proposes a more comprehensive performance measurement system, incorporating new key indicators such as resource utilization, system scalability, and user experience, making 3D visualization assessment more scientific and rigorous. Second, this study refines Cesium rendering technology through hierarchical detail techniques and processes surface data for smart venues. In addition, this technology is systematically compared with rendering results from Digital Twin (DT), Building Information Modeling (BIM), and Octree models. This reveals the impact of data storage and processing methods on 3D visualization performance and provides optimization solutions for different application scenarios. Finally, the study proposes an adaptive rendering strategy for IoT environments to optimize 3D visualization performance. Especially in the face of increasing dynamic data transmission and processing demands, it can offer new perspectives and approaches. In practice, this study provides crucial reference points for solving technical challenges in smart venue visualization, contributing significantly to advancements in the field. For urban planners, the optimization solutions provided in this study can enhance the efficiency of 3D data visualization. This makes the digital representation of urban infrastructure, traffic flow, and building space management more intuitive, thus improving the accuracy of urban planning and management. For developers and engineers, the database selection strategies and rendering optimization solutions proposed here can reduce hardware costs and improve the adaptability of 3D visualization systems across different technological environments. These help engineers efficiently manage building information in complex data scenarios. Additionally, for venue operators, the proposed adaptive rendering strategy can maintain efficient data visualization under high-concurrency access and complex data computation environments. This enables smarter venue management, smoother interaction experiences, and optimizing user experience and operational decision-making. Through these practical contributions, this study fills the existing technological gaps in 3D visualization methods for smart venue management. Meanwhile, it provides scientifically grounded technical guidance and application solutions for further development in the field.

The innovation of this study lies in developing a novel comprehensive evaluation framework. It aims to systematically quantify and optimize the applicability and performance of different 3D visualization technologies in IoT environments, addressing gaps in existing research. Unlike previous studies that primarily focus on individual technologies or specific environments, this study is the first to place multiple mainstream 3D visualization methods under a unified evaluation system. Their performance across databases and browsers is compared to reveal the adaptability and optimization directions of different technologies in dynamic environments. The core innovations of this framework include the following. An adaptive precision adjustment method is proposed based on the flow characteristics of IoT data, ensuring stable rendering performance even under high data density and computational pressure; Key indicators such as resource utilization, system scalability, and user interaction experience are introduced, in addition to traditional metrics like rendering speed, frame rate, and response time. This evaluates 3D visualization methods more comprehensively and precisely; By analyzing data transmission efficiency across different databases in IoT environments, optimal data storage and retrieval solutions are proposed to enhance the real-time performance and stability of 3D visualization systems. Furthermore, this study compares various technological solutions through experimental data analysis and offers optimization recommendations for different application scenarios. Through these innovations, this study provides feasible technical guidance for 3D visualization in smart venues while offering a new research paradigm and application direction for optimizing 3D visualization methods in IoT environments.

## 2. Literature Review

Foreign cities have developed various VR 3D visualization systems, such as Google Earth, Skyline Globe Enterprise Solution, ArcGIS Pro, CityEngine, NASA Web World Wind, and Cesium. These systems, built on widely-used visualization frameworks, have been further enhanced to possess rich functionalities, including observing mountains, rivers, satellite imagery, and 3D buildings [16-21]. While systems like Google Earth are feature-rich, they have limited service coverage and lack robust mapping and spatial analysis capabilities. In contrast, Cesium has gained popularity for developing 3D visualization systems tailored for smart cities worldwide. Its open-source JavaScript library is combined with Web Graphics Library (WebGL) technology, supporting various data types such as massive terrains, vector data, and map imagery. This makes Cesium a preferred choice for 3D geospatial visualization [22,23]. For instance, Anand and Deb (2024) utilized high-resolution remote sensing data and Light Detection and Ranging (LiDAR) technology to generate more refined urban 3D models. These models facilitated enhanced visualization while enabling simulation and prediction, aiding decision-makers in better understanding potential risks and impacts on urban environments [24]. Sadowski (2024) asserted that urban informatics drove the development of smart cities, particularly in data integration and intelligent decision support. Comprehensive urban information platforms could be constructed by integrating various urban data sources, such as traffic, environmental, and socioeconomic data. These platforms leveraged advanced data analysis techniques and machine learning (ML) algorithms to monitor urban operations in real-time and conducted predictive analysis, optimizing urban services and resource allocation. Additionally, urban informatics enhanced public engagement through visualization technologies, offering intuitive insights into urban development and fostering citizen involvement and feedback on policy-making [25]. IoT-driven visualization technologies achieved deep integration with smart city infrastructure, making urban management more intelligent and efficient. By deploying numerous sensors, cities could collect various environmental data in real-time, such as air quality, traffic flow, and public safety. Bhavsar et al. (2024) found that data integrated and analyzed through IoT platforms generated dynamic visual dashboards, helping managers access urban operational statuses. Moreover, cloud-based IoT platforms supported multi-user access and real-time updates, while data mining techniques uncovered potential issues, enhancing urban responsiveness. This IoT-based data visualization technology substantially improved urban sustainability and residents' quality of life [26]. In the context of smart city visualization, Vitanova et al. (2023) proposed the first energy map booklet for Sofia, Bulgaria. They used GIS and statistical tolerance methods to estimate building energy consumption. GIS was used for result classification and visualization [27]. Sun et al. (2023), in turn, employed a comprehensive approach to address the challenges posed by data in smart city development. They focused on data extraction and transformation, data sharing and exchange platforms, joint databases, and element searches [28].

In contrast, while the development of 3D GIS technology in China began relatively recently, it has made significant progress in recent years. The concept of "Smart Earth" was proposed by domestic International Business Machine (IBM) companies. As a result, many Chinese cities have initiated smart city projects and 3D GIS, with hundreds of cities either planning or already implementing them [29]. Domestic research primarily targeted enhancing urban management efficiency, improving tourist experiences, and optimizing power engineering planning through 3D visualization technology [30]. Qi et al. (2023) utilized traditional 3D modeling techniques alongside modern BIM systems in their research on 3D visualization. They leveraged BIM's ability to integrate rich architectural information, enabling detailed representation of building data of smart cities at a super-microscopic level [31]. Furthermore, Krašovec et al. (2024) developed an intelligent fire visualization platform based on 3D GIS to address fire safety issues in smart cities. This platform achieved fire phase recognition through real-time videos accessed through a visual interface, providing a feasible solution for intelligent firefighting and rescue operations [32].

In conclusion, developed countries have made significant progress in 3D visualization and smart city development, including creating systems like Google Earth and Cesium. These systems are widely applied in urban planning, building inspections, and energy assessments, but they still exhibit shortcomings in service coverage, spatial analysis capabilities, and support for customized applications. Meanwhile, despite domestic research's relatively late start, it has made strides in smart city development under the "Smart Earth" initiative. However, domestic applications remain largely focused on single domains (e.g., electricity, tourism, and park management), with limited exploration of cross-domain data integration and intelligent decision support. Current research presents the following deficiencies. (1) International technologies lack unified technical standards and extensive cross-domain integration capabilities for complex scenarios. (2) Domestic research primarily emphasizes efficiency improvements within specific domains but demonstrates insufficient depth in integrating multi-source data and data-driven predictive capabilities. (3) Existing literature lacks systematic analysis of key bottlenecks in 3D visualization technologies, such as real-time processing, refined modeling, and user interaction experience. To address these gaps, this study focuses on integrating IoT technology with GIS, Cesium, and other tools to develop an efficient and intelligent 3D visualization platform for smart cities. This platform aims to enhance urban management efficiency and facilitate the intelligent transformation of scientific decision-making.

# 3. Research Model

The 3D visualization of smart venues requires the integration of various advanced technologies to address complex application scenarios. The smart venue 3D visualization system proposed in this study is centered on IoT, GIS, BIM, and Cesium rendering technology. It achieves real-time data collection, spatial modeling, building information integration, dynamic rendering, and interactive visualization. The following sections discuss the role of these core technologies in the system and explain their data flow and interaction mechanisms to create an optimized 3D visualization solution for smart venues.

## 3.1. Internet of Things

IoT refers to the network of interconnected physical devices, sensors, and other objects that communicate and share data via the Internet. This real-time data aggregation and analysis offer a valuable source of information for 3D visualization. This enables a more accurate and comprehensive representation of dynamic environments, such as cities, buildings, and facilities [33,34]. Figure 1 illustrates the application scenarios of IoT technology.

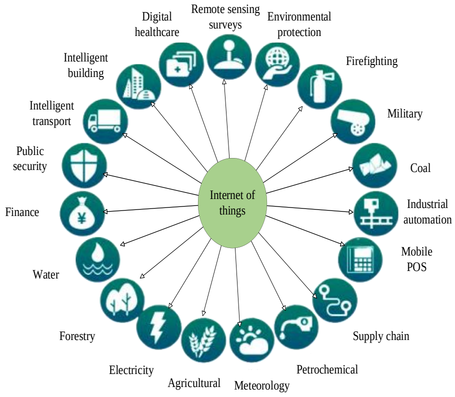



**Fig. 1.** Application scenarios of IoT technology

IoT serves as the data collection layer in this study, responsible for real-time collection of sensor data from within the venue and its surrounding environment, such as temperature, humidity, personnel movement, and equipment status. This data is transmitted wirelessly via Wi-Fi, 5G, or LoRa to a cloud database and is updated in real-time in the GIS and BIM systems. This ensures that the 3D visualization system reflects the most up-to-date venue status. The data provided by IoT offers dynamic inputs for GIS to perform geographic spatial positioning. Moreover, it supplies monitoring information on the internal conditions of the building for BIM, ensuring the real-time and dynamic interaction capabilities of the 3D visualization system.

### 3.2. Geographic Information System

The 3D visualization of GIS combines remote sensing and global positioning systems to integrate and process geographic data, creating digital models and precise locations. Through these technologies, it is possible to simulate different scenarios and offer intuitive information for urban planning, resource management, and more, advancing the development of the digital earth [35,36]. The value of 3D GIS is revealed in Figure 2.

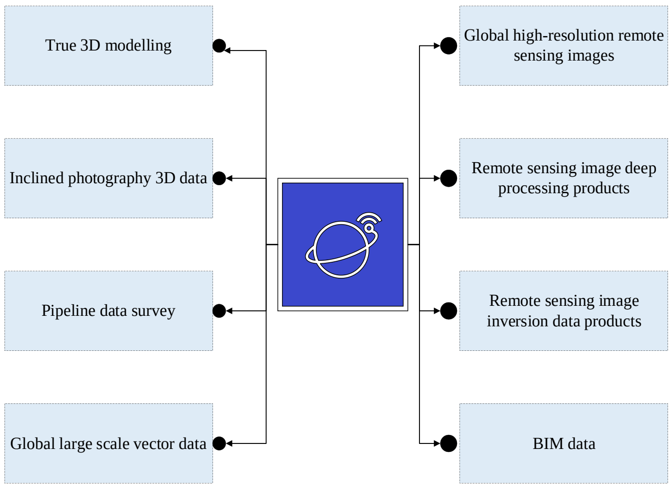



**Fig. 2.** Value of 3D GIS technology

In this study, GIS functions as the spatial data management layer, primarily responsible for integrating and processing the environmental data collected by IoT and providing geographic information such as terrain, buildings, and transportation. GIS constructs a 3D geographic environment using vector data (e.g., building outlines, and road networks) and raster data (e.g., satellite imagery, and topographic maps). This ensures that Cesium rendering is supported by accurate spatial background information. Additionally, GIS provides high-precision coordinate mapping capabilities, enabling seamless integration of the BIM with the actual geographic location, and ensuring the accuracy of the 3D visualization system.

### 3.3. Building Information Modeling

BIM integrates information from the design, construction, and operational phases of a building to enable 3D visualization, helping project teams better understand the building structure. BIM provides accurate and consistent data models, promoting design efficiency, reducing costs, minimizing errors, and supporting the building lifecycle management. During the design, construction, and maintenance phases, BIM's 3D visualization provides real-time, detailed information for the project, ensuring smooth implementation and sustainable operation [37,38]. Figure 3 depicts the building lifecycle in BIM.

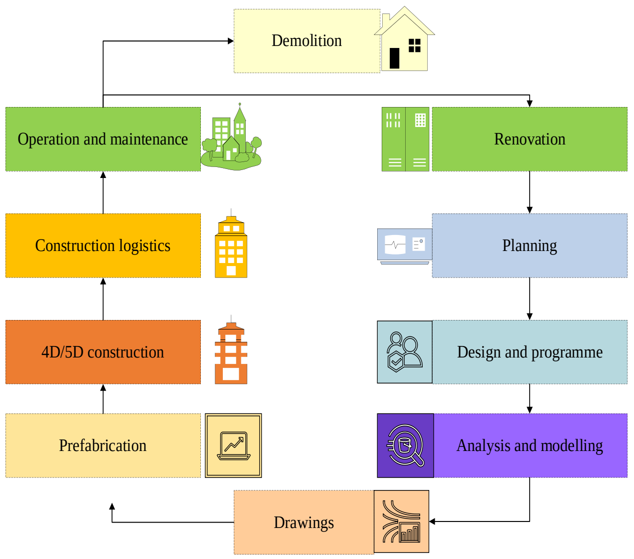



**Fig. 3.** BIM building life cycle

In this study, BIM functions as the building data management layer, primarily responsible for creating 3D structural models of the venue, integrating building design, construction, and operational data, and spatially aligning them with GIS data. The building models provided by BIM contain detailed structural information, such as beams, columns, walls, and pipes, and, in combination with IoT sensor data, enable real-time visualization of the building's internal status. When temperature and humidity sensors inside the venue detect anomalies, BIM can visually display the problem areas through a 3D interface and interact with IoT devices to make automatic adjustments. Moreover, BIM optimizes data through hierarchical detail techniques, allowing high-precision building models to be efficiently loaded into Cesium for rendering, thereby improving rendering efficiency and interactive smoothness.

### 3.4. Cesium

Cesium is an open-source JavaScript library focused on high-performance, dynamic 3D geospatial visualization. It offers powerful geospatial data rendering capabilities, supporting the loading of various types of geographic information, such as terrain, vector data, and satellite imagery, to present a realistic depiction of the Earth's surface. Based on WebGL technology, Cesium is cross-platform and allows users to experience high-performance visualization without the need for plugins. Its open JavaScript API supports custom map styles, terrain transformations, and integration with other GIS systems, making it widely used in smart cities, geographic information science, and virtual tourism, among other fields [39-41]. Figure 4 shows Cesium's system architecture and rendering mechanism.

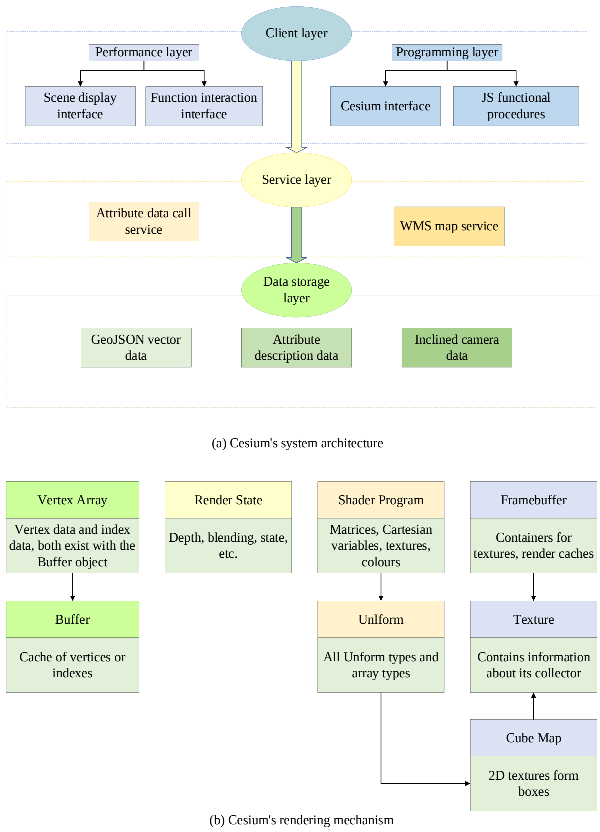



**Fig. 4.** Cesium's system architecture and rendering mechanism

Cesium, as the 3D visualization engine layer, integrates spatial data provided by GIS and BIM and performs efficient rendering and interactive visualization. This study utilizes WebGL technology, enabling Cesium to operate efficiently across different browser environments and achieve cross-platform 3D visualization display. The core functions of

Cesium include the following. Loading and rendering BIM and GIS data provide high-precision 3D model display; Real-time interaction allows users to freely zoom, rotate, and switch between different viewpoints to view venue information; Large-scale data optimization improves rendering efficiency and ensures a smooth visualization experience even in high data density environments. The Cesium's WebGL rendering technology also allows this study to integrate dynamic data visualization. This enables real-time 3D display of venue temperature, humidity, and personnel flow by combining IoT device data, further enhancing the system's intelligence.

This study employs Level of Detail (LOD) techniques to optimize Cesium rendering. Before importing BIM and GIS data, LOD algorithms are used to classify the models. Different precision versions of the model are applied to diverse viewing distances and display requirements. The low-LOD version employs simplified geometric models and low-detail textures, suitable for distant views. In contrast, the high-LOD version retains more complex architectural details and is appropriate for close-up or zoomed-in views. This approach allows Cesium to dynamically adjust the LOD of models based on the user's viewing distance, improving rendering efficiency and reducing computational load.

Cesium uses screen space error (SSE) to calculate the appropriate LOD for each object to display. The SSE calculation considers the viewing distance and the object's projected size on the screen. When the viewing distance is large or the object's projected area is small, the system automatically selects a lower LOD. This mechanism dynamically adjusts as the user zooms or moves the viewpoint, improving overall performance. The calculation of SSE is as follows:

$$SSE = \frac{P}{RS} \tag{1}$$

$P$ refers to the SSE of the object; $R$ is the distance of the object in the viewing space; $S$ represents the screen projection size of the object.

During user interaction, Cesium dynamically calculates the required LOD for the objects within the current view and switches between models of different precision in real-time. A quadtree or octree-based data management approach is adopted to optimize rendering performance, ensuring that only the geometric details within the user's view are rendered, while areas outside the view automatically switch to a lower LOD. Moreover, through GPU Instancing technology, Cesium can reduce draw calls while maintaining visual effects, enhancing rendering efficiency in large-scale data environments.

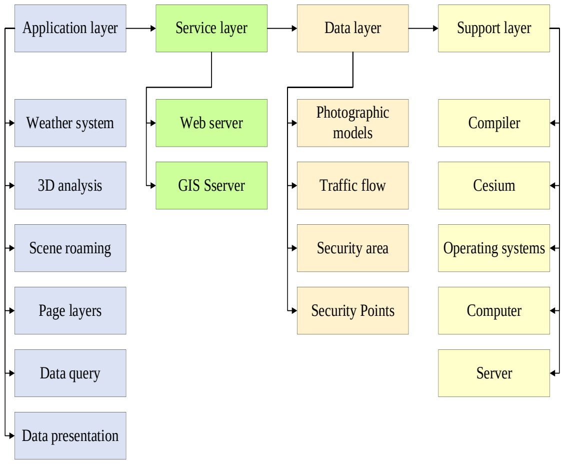
To optimize Cesium's loading speed and reduce network load, this study uses the Zstandard compression algorithm to optimize the storage and transmission of 3D Tiles data. This allows the venue data to load quickly and achieve efficient, smooth visualization. Additionally, by using the Batched 3D Model (B3DM) format, LOD levels are managed in bulk, improving data stream control efficiency. This ensures that the system can quickly load models with matching precision from different viewpoints.

In addition to improving Cesium's 3D rendering performance, user experience is also a key focus of optimization. By combining GPU-accelerated rendering and frustum culling techniques, only objects within the current view are rendered, avoiding unnecessary computations. With WebGL-based lighting and material optimizations, even at low-LOD models, lighting effects and detail representation remain accurate, further enhancing the user's interaction experience. This optimization scheme is particularly suitable for high data density environments and scenarios with high real-time requirements, ensuring efficient system performance in complex application settings.

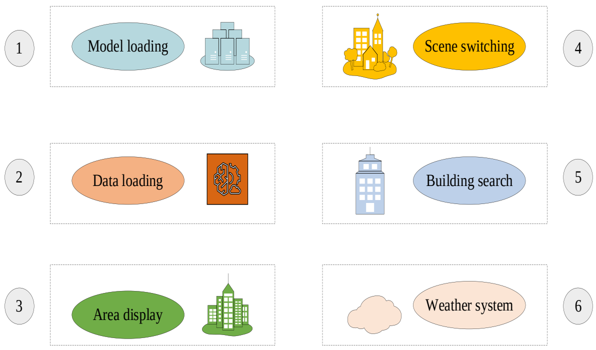
### 3.5. Design of Visualisation System Based on the Internet of Things and Under Cesium

The design of the 3D visualization system for smart venues adheres to several key system engineering design principles. It meets user needs while maintaining high practicality, completeness, innovation, scalability, and ease of operation. First, the system is grounded in practicality, ensuring it effectively addresses real-world requirements. Second, the principle of completeness guarantees that all required features are fully implemented, aligning with user needs and leaving no gaps. From a technological standpoint, the principle of innovation drives the adoption of advanced web-based open-source frameworks and drone-based oblique photography technology, ensuring high performance and technological advantages. Scalability is also a priority, allowing the system to accommodate future feature expansions as requirements evolve. Finally, ease of operation emphasizes a simple and user-friendly system interface to facilitate smooth interaction for users with varying levels of expertise, thus enhancing overall efficiency. Figure 5 displays the visualization system's overall architecture and functions based on IoT and Cesium.

By integrating IoT, GIS, BIM, and Cesium technologies, this study develops a 3D visualization system tailored for smart venues. This system supports real-time data acquisition, dynamic updates, and efficient rendering while having high scalability and easy operation. Its design principles emphasize practicality, comprehensiveness, advancement, and user accessibility. The system demonstrates significant potential in supporting the Smart 14th National Games Digital Twin project. By combining real-time data from IoT with spatial modeling capabilities of GIS and BIM, and leveraging Cesium for efficient rendering, the proposed system model effectively enhances smart venue management and user experience. At the same time, it provides robust technical support for smart city development.




(a) Overall visualisation system based on IoT and Cesium




(b) Functions of the visualisation system based on IoT and Cesium

**Fig. 5.** Overall architecture and system functions of the visualization system based on IoT and Cesium

## 4. Experimental Design and Performance Evaluation

### 4.1. Datasets Collection

The dataset used in this study primarily includes building model data and dynamic scene data. The building model data consists of geometric information, location distribution, and height of buildings within the park. The data is primarily sourced from the public geographic information databases (CityGML) and Open Street Map (OSM) datasets. CityGML provides high-precision 3D geographic information, suitable for fine-grained building modeling, while OSM offers reliable data support for basic building contours and distribution. By integrating these two data sources, a highly realistic 3D building model with spatial analysis capabilities can be constructed, laying the foundation for the system's visualization and analysis functions. The dynamic scene data is mainly based on the Smart City Data Catalogue dataset, which integrates real-time data from over 50 cities, encompassing Barcelona, Berlin, London, Paris, and others. This dataset primarily includes urban traffic, environmental monitoring, energy consumption, citizen behavior, and demand analysis, among others. With a real-time updating mechanism, the dataset can reflect the latest state of urban operations and simulate dynamic interactions within the park, including personnel movement and vehicle trajectories. The data processing workflow involves model format conversion, storage, and the introduction of real-time dynamic data to ensure the diversity and dynamics of experimental data.

Moreover, this study leverages My Structured Query Language (MySQL), PostgreSQL, and MongoDB as databases, with the geographic information model rendered in the 3D Tiles data format. MySQL, a popular relational database management system, is favored in web applications for its compact size, fast speed, and low total cost of ownership. PostgreSQL, another prominent relational database management system, is recognized for its stability, scalability, and robust community support. It excels over MySQL in managing spatial and large-scale data, making it a suitable alternative database system. Cesium, a JavaScript-based 3D mapping platform, delivers powerful visualization capabilities for this system. Using 3D Max modeling software, the buildings and surveillance equipment within the park are rendered, lit, and arranged to simulate a realistic park environment. These models are then exported in .dae and .obj formats before being converted into Cesium-compatible formats. Furthermore, the system can load images from Tianditu, Google, and Amap, enriching the available geographic information. MongoDB is a non-relational database to handle dynamic and unstructured data, particularly in IoT and smart city contexts. Meanwhile, it can easily store large volumes of real-time data generated by sensors, and support rapid data storage, writing, and query. This effectively copes with the needs of data fluctuations and dynamic updates in these environments. Overall, MySQL, a widely used relational database, offers fast query speeds and ease of management, making it suitable for preliminary testing and routine data processing. PostgreSQL provides more robust spatial data handling capabilities and extensibility, accommodating complex query tasks and making it particularly effective for geospatial data analysis. MongoDB excels in handling unstructured data and real-time updates, meeting the needs of dynamic data streams. By comparing the performance of these three databases, this study investigates their impact on 3D rendering efficiency and offers guidance for selecting appropriate databases in different scenarios.

In practical application, the datasets and databases used in this study are highly representative and can comprehensively support the research objectives of smart venue 3D visualization and dynamic scene simulation. Specifically, selecting the CityGML and OSM datasets is based on their complementary advantages. CityGML provides high-precision 3D building geometric information, suitable for detailed modeling; OSM covers a wide range of building contours and distribution data, ensuring the authenticity and completeness of the model across large spatial areas. Moreover, the Smart City Data Catalogue integrates real-time data from over 50 cities, including traffic flow, environmental monitoring, energy consumption, and citizen behavior analysis. These can dynamically reflect the actual state of city operations, providing a scientific basis for dynamic interactions within the park. Regarding databases, MySQL is well-suited for structured data processing due to its fast query and easy management characteristics. PostgreSQL, with its powerful spatial data processing capabilities, is more suitable for complex geographic information analysis, while MongoDB excels in handling unstructured, real-time updated data. This multi-database approach significantly improves data processing efficiency and system responsiveness compared to traditional single-database methods. All data are cross-verified with authoritative public data sources and monitored in real-time to ensure their accuracy and representativeness. Thus, it can faithfully reproduce the actual dynamics of the city and provide solid data support and a reliable dynamic simulation foundation for this study. These datasets and databases have been extensively validated in practical applications. CityGML and OSM data are widely used in various smart cities and GIS applications, with their accuracy and reliability proven in practice. The Smart City Data Catalogue dataset, through its real-time updating mechanism, reflects the real state of city operations and supports dynamic scene simulation and analysis. The selection of databases is also based on their maturity and performance in practical applications. MySQL and PostgreSQL excel in traditional data processing and spatial data analysis, while MongoDB has clear advantages in IoT and real-time data processing.

Subsequently, the data collection process follows a structured sequence of steps. First, 3D Max modeling software is employed to render, illuminate, and arrange the buildings and surveillance equipment in the park, simulating a realistic environment. Second, the rendered model data is exported in .dae and .obj formats. Third, the .dae files are converted to .glTF format using the colladaToglTF.exe tool. Fourth, the converted model data is stored in the database for system use.

Notably, the data used in this study is sourced from publicly available databases, ensuring compliance with relevant data usage and sharing policies. Since this study does not involve human participants during data collection and processing, there are no concerns regarding personal privacy or ethical risks. All data processing procedures and methods adhere to academic integrity and ethical standards, ensuring the study's transparency and reliability.

### 4.2. Experimental Environment

This study employs VS Code as the code editor due to its lightweight and open-source features, which allow for flexible installation of plugins to facilitate dynamic data processing. The experimental environment is configured with an Intel Core i5-9400F six-core processor, 8GB of RAM, and a 1TB 5400 RPM hard drive, running a 64-bit version of Windows 11, alongside JDK 1.8.0 and Tomcat 8.5.6. Real-time data simulation is incorporated to enhance adaptability to data fluctuations and dynamic updates, enhancing the system's flexibility. Geographic information processing is handled by ArcGIS 10.2.2 and QGIS 3.16, ensuring the capability to handle large-scale sensor network data. Cesium version 1.91 is used for 3D geographic information rendering, ensuring compatibility with mainstream browsers. The experiment also considers browser version requirements for Cesium to maintain consistent and stable rendering results. Additionally, dynamic data streams and sensor data variations are simulated to reflect real-world application scenarios, guaranteeing the experimental results' reliability and practical significance. These optimization measures enable the study to accurately present operational conditions in smart cities, thereby increasing its practical application value. For browser selection, this study utilizes not only Chrome 85 and Firefox 80 but also includes Microsoft Edge 11 and Opera 45. Microsoft Edge, built on the Chromium kernel, offers good performance and compatibility, effectively leveraging Cesium's rendering capabilities. Meanwhile, Opera provides a convenient browsing experience with features like built-in data-saving and enhanced privacy protection features. The diverse selection of databases and browsers offers broader adaptability and a more reliable testing environment. This ensures system stability and effectiveness under various conditions, thus meeting the demands of smart city applications.

Furthermore, this study conducts a comparative analysis of various methods, including DTs, BIM, Octree, Cesium, Unity, Unreal Engine, and Three.js, based on their respective advantages and applicability in different domains. The DT method demonstrates high efficiency in data processing and analysis, making it suitable for complex datasets. BIM offers robust support for architectural and engineering visualization needs. Octree excels in 3D spatial optimization and scene management, making it ideal for large-scale data processing and rendering tasks. Cesium and Three.js exhibit leading advantages in 3D visualization and GIS, supporting efficient dynamic data rendering and interactive applications. Unity and Unreal Engine, with their powerful game engine functionalities, enable multi-layered and multidimensional interactive experiences. Integrating these methods comprehensively addresses multi-dimensional visualization requirements, ranging from architectural models to 3D scenes, providing more extensive analytical tools and visualization solutions for this study.

### 4.3. Parameters Setting

3D Tiles is an open standard used for storing and exchanging 3D geographic information models. It organizes data through a tiling approach, enabling efficient loading, display, and interaction of large-scale 3D models, especially suitable for 3D visualization on the Web, thus providing a fast and smooth user experience. The data conversion process involves several steps. First, it includes the preparation of raw data, such as GIS, CAD, or remote sensing data. Then, data preprocessing is performed. Next, appropriate conversion tools, such as Cesium's 3d-tiles-tools, are selected. Optimization parameters are set for the conversion process. Following this, data validation and debugging are conducted. Finally, the data is deployed to the target platform to ensure that it meets expectations and can be successfully applied in real-world scenarios.

Moreover, evaluation indicators are essential to assess the performance of different database and browser combinations in IoT and non-IoT environments for 3D visualization rendering methods. The evaluation focuses on two mainstream databases, MySQL and PostgreSQL, tested in Chrome and Firefox browsers. Key evaluation indicators encompass response time, rendering speed, and frame rate. Response time measures the duration from receiving a request to generating a response; Rendering speed indicates the system's processing speed when rendering the geographic information model; Frame rate refers to the number of image frames the system displays per second. A comprehensive evaluation of these indicators provides valuable insights into how database and browser combinations affect 3D visualization rendering performance, aiding in system optimization.

In the experiments, consistency in versions and settings is maintained to enhance the reliability and replicability of results. Particularly, this includes using Cesium version 1.70, MySQL 8.0, and PostgreSQL 12.0 databases, along with Chrome 85 and Firefox 80 browsers. Moreover, a unified 64-bit Windows 7 operating system, Oracle JDK 1.8, Tomcat 9.0, and default transaction consistency levels (REPEATABLE READ for MySQL and READ COMMITTED for PostgreSQL) are used for databases. This ensures consistent testing conditions and reliable performance evaluations.

The data processing workflow in this study is based on the industry-standard 3D Tiles conversion guidelines. However, it incorporates innovative design in selecting and optimizing key parameters for large-scale dynamic scene applications in smart cities and IoT environments. First, during the data preprocessing, raw GIS, CAD, or remote sensing data is cleaned, coordinate systems are converted, and data is compressed to ensure compliance with the 3D Tiles specifications. During the conversion process, after repeated experiments, the size of each tile is set to 256 pixels. This parameter enables efficient LOD management and frustum culling while maintaining sufficient geometric detail, thus optimizing rendering performance. Next, a 2048 KB data loading buffer is set, which ensures the continuity of data preloading, preventing interruptions during loading, while avoiding excessive memory usage that could impact system responsiveness. For data compression, the Zstandard algorithm is selected, as it offers both a high compression ratio and fast decompression capability, significantly reducing data transmission time without sacrificing visual quality. Cesium's 3d-tiles-tools and commercial software FME are utilized, adjusting conversion parameters based on actual needs. This ensures that the final converted 3D Tiles data achieves the best balance between loading speed, rendering performance, and data accuracy, all while undergoing rigorous validation and debugging to ensure its accuracy and stability.

### 4.4. Performance Evaluation

Initially, the lightweight results of 3D models based on different databases and browsers are shown in Figure 6.

In Figure 6, the combination of different databases (MySQL and PostgreSQL) and browsers (Chrome and Firefox) markedly impacts the performance of 3D visualization rendering methods in IoT environments. Using MySQL and Chrome, the compressed model sizes show notable differences. Among the tested methods, Cesium achieves the best compression result, with a model size of 5612 KB. In comparison, Unity produces a model size of 6021 KB, Three.js results in 5720 KB, and Octree generates 6754 KB, all slightly less efficient than Cesium. When PostgreSQL is used with Chrome, the models' lightweight performance is relatively lower, with Cesium achieving a compressed size of 13991 KB, followed by Three.js. Also, under PostgreSQL and Firefox, Cesium performs exceptionally well, with a compressed model size of 11134 KB. MySQL combined with Firefox shows relatively balanced performance across all rendering methods, but Cesium still delivers the best results, achieving a model size of 6187 KB. In non-IoT environments, all rendering methods, including Cesium, Unity, Unreal Engine, Three.js, Octree, BIM, and DTs, exhibit a decline in performance. When

PostgreSQL is paired with Firefox, the overall compressed model sizes increase. Compared to Unity and Unreal Engine, which result in 13450 KB and 13125 KB, respectively, Cesium maintains superior performance across environments, with a size of 11705 KB. Under MySQL and Chrome, Cesium achieves a size of 6084 KB, which is 426 KB and 298 KB smaller than Unity and Unreal Engine. Similarly, under MySQL and Firefox, Cesium remains the best-performing method, followed by Three.js and Unreal Engine, with Unity slightly lagging. BIM and Octree methods exhibit minor differences in performance across the two databases, but their overall trends remain consistent. The DT method performs the worst in all scenarios. Overall, the lightweight performance of 3D models is distinctly influenced by the rendering method, database type, and browser environment. Cesium demonstrates superior compression efficiency in all test scenarios, consistently outperforming other methods in IoT and non-IoT environments.

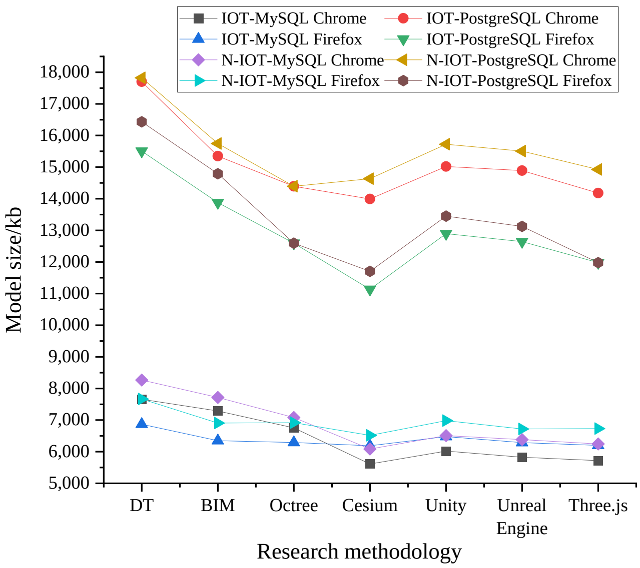
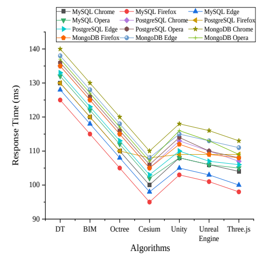
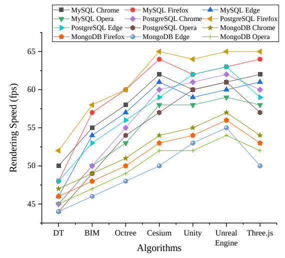



**Fig. 6.** Lightweight results of 3D models with different databases and browsers

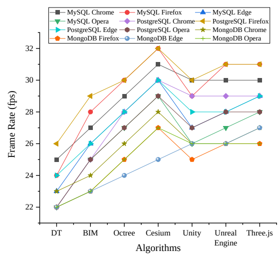
Subsequently, this study analyzes the system performance using various database and browser combinations. The results are presented in Table 1 and Figure 7.



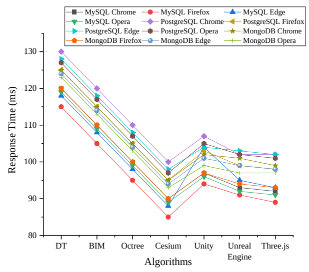

(a) Response time in non-internet of things environments



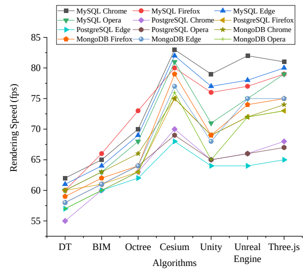

(b) Rendering Speed in non-internet of things environments



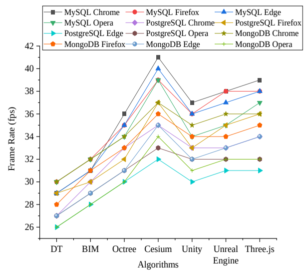

(c) Frame rate in non-internet of things environments




(d) Response time in internet of things environments




(e) Rendering Speed in internet of things environments

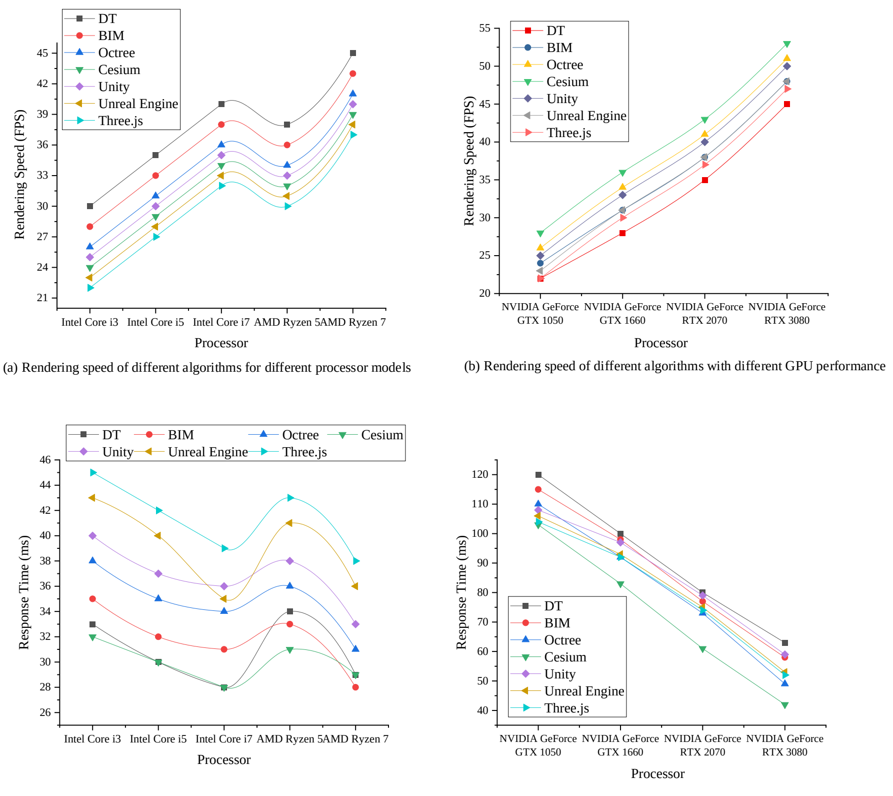



(f) Frame rate in internet of things environments

**Fig. 7.** Distribution of resource utilization

The data comparison in Table 1 and Figure 7 reveals that in a non-IoT environment, the response times for MySQL and PostgreSQL are fairly similar, at 130 ms and 135 ms. In contrast, MongoDB exhibits slightly higher latency, with a response time of 140 ms in the Chrome browser. However, in an IoT environment, the response times for MySQL drop to 120 ms and 110 ms in the DTs and BIM. MongoDB's response time decreases to 125 ms, demonstrating that IoT optimization significantly enhances system responsiveness. Regarding rendering speed, in the non-IoT environment, MySQL achieves a rendering speed of 50 FPS in the Chrome browser. In the IoT environment, however, MySQL and MongoDB reach impressive speeds of 83 FPS and 75 FPS, respectively, using Cesium rendering methods, demonstrating substantial improvement. As for frame rates, MongoDB's frame rate in the non-IoT environment is 23 FPS, while in the IoT environment, MySQL's frame rate increases to 41 FPS, indicating better dynamic rendering performance. Overall, the IoT environment offers remarkable advantages in response time, rendering speed, and frame rate. Based on these findings, optimization enhancement strategy should focus on several key areas. Firstly, to address MongoDB's higher response time, optimizing database indexing and data structures is recommended to reduce query latency, thus enhancing overall performance. Secondly, to improve real-time rendering capabilities in IoT environments, implementing caching mechanisms and preloading techniques can help minimize data retrieval delays and increase processing efficiency. Meanwhile, employing asynchronous rendering strategies can notably boost frame rates, especially under heavy load, ensuring a smooth user experience. Furthermore, integrating ML algorithms to analyze user behavior and dynamically adjust system resource allocation can further optimize database access efficiency and data rendering speed. Finally, specific performance optimizations for different browser environments help adapt to the unique characteristics of each platform, achieving better cross-platform consistency and user satisfaction. Implementing these strategies can substantially enhance the system's response speed, rendering quality, and user experience.

Then, the 3D rendering effect under different hardware conditions and methods is suggested in Figure 8.




(a) Rendering speed of different algorithms for different processor models

(b) Rendering speed of different algorithms with different GPU performance

(c) Response time of different algorithms for different processor models

(d) Response time of different algorithms with different GPU performance

**Fig. 8.** 3D rendering effects under different hardware conditions with various methods

**Table 1.** Performance of the system

| Environment | Non-Internet of Things Environment | | | | Internet of Things Environment | | |
|---|---|---|---|---|---|---|---|
| Database/Browser | Rendering Method | Response Time (millisecond (ms)) | Rendering Speed (frames per second (FPS)) | Frame Rate (FPS) | Response Time (ms) | Rendering Speed (FPS) | Frame Rate (FPS) |
| MySQL Chrome | DTs | 130 | 50 | 25 | 120 | 62 | 29 |
| | BIM | 120 | 55 | 27 | 110 | 65 | 31 |
| | Octree | 110 | 58 | 29 | 100 | 70 | 36 |
| | Cesium | 100 | 62 | 31 | 90 | 83 | 41 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Unity | 108 | 60 | 30 | 97 | 79 | 37 |
| | Unreal Engine | 106 | 61 | 30 | 93 | 82 | 38 |
| | Three.js | 104 | 62 | 30 | 92 | 81 | 39 |
| | DTs | 125 | 48 | 24 | 115 | 60 | 30 |
| | BIM | 115 | 57 | 28 | 105 | 66 | 32 |
| | Octree | 105 | 60 | 30 | 95 | 73 | 35 |
| MySQL Firefox | Cesium | 95 | 64 | 32 | 85 | 80 | 39 |
| | Unity | 103 | 62 | 29 | 94 | 76 | 36 |
| | Unreal Engine | 101 | 63 | 31 | 91 | 77 | 38 |
| | Three.js | 98 | 64 | 31 | 89 | 79 | 38 |
| | DTs | 128 | 46 | 23 | 118 | 61 | 29 |
| | BIM | 118 | 54 | 26 | 108 | 64 | 31 |
| | Octree | 108 | 57 | 28 | 98 | 69 | 35 |
| MySQL Edge | Cesium | 98 | 61 | 30 | 88 | 82 | 40 |
| | Unity | 105 | 59 | 27 | 104 | 77 | 36 |
| | Unreal Engine | 103 | 60 | 28 | 95 | 78 | 37 |
| | Three.js | 100 | 61 | 29 | 93 | 80 | 38 |
| | DTs | 132 | 45 | 22 | 119 | 60 | 30 |
| | BIM | 122 | 50 | 25 | 109 | 63 | 32 |
| | Octree | 112 | 53 | 27 | 99 | 68 | 34 |
| MySQL Opera | Cesium | 102 | 61 | 29 | 89 | 81 | 39 |
| | Unity | 108 | 58 | 26 | 96 | 71 | 34 |
| | Unreal Engine | 106 | 59 | 27 | 92 | 75 | 35 |
| | Three.js | 105 | 58 | 28 | 91 | 79 | 37 |
| | DTs | 135 | 45 | 22 | 130 | 55 | 27 |
| | BIM | 125 | 50 | 25 | 120 | 60 | 30 |
| | Octree | 115 | 55 | 28 | 110 | 63 | 33 |
| PostgreSQL Chrome | Cesium | 105 | 64 | 30 | 100 | 70 | 35 |
| | Unity | 113 | 61 | 29 | 107 | 65 | 33 |
| | Unreal Engine | 110 | 62 | 29 | 102 | 66 | 33 |
| | Three.js | 107 | 60 | 29 | 102 | 68 | 34 |
| | DTs | 130 | 52 | 26 | 125 | 60 | 29 |
| | BIM | 120 | 58 | 29 | 115 | 61 | 30 |
| | Octree | 110 | 60 | 30 | 105 | 63 | 32 |
| PostgreSQL Firefox | Cesium | 108 | 67 | 32 | 95 | 75 | 37 |
| | Unity | 109 | 64 | 30 | 103 | 69 | 33 |
| | Unreal Engine | 109 | 65 | 31 | 99 | 72 | 35 |
| | Three.js | 109 | 65 | 31 | 98 | 73 | 36 |
| | DTs | 133 | 48 | 24 | 128 | 57 | 26 |
| | BIM | 123 | 53 | 26 | 118 | 60 | 28 |
| | Octree | 113 | 56 | 28 | 108 | 62 | 30 |
| PostgreSQL Edge | Cesium | 103 | 65 | 30 | 98 | 68 | 32 |
| | Unity | 110 | 62 | 28 | 104 | 64 | 30 |
| | Unreal Engine | 107 | 63 | 28 | 103 | 64 | 31 |
| | Three.js | 106 | 59 | 29 | 102 | 65 | 31 |
| | DTs | 136 | 44 | 22 | 127 | 58 | 27 |
| | BIM | 126 | 49 | 25 | 117 | 61 | 29 |
| | Octree | 116 | 54 | 27 | 107 | 64 | 31 |
| PostgreSQL Opera | Cesium | 106 | 63 | 29 | 97 | 69 | 33 |
| | Unity | 114 | 60 | 27 | 105 | 65 | 32 |
| | Unreal Engine | 110 | 61 | 28 | 102 | 66 | 32 |
| | Three.js | 108 | 57 | 28 | 101 | 67 | 32 |
| | DTs | 140 | 47 | 23 | 125 | 60 | 30 |
| | BIM | 130 | 49 | 24 | 115 | 63 | 32 |
| | Octree | 120 | 51 | 26 | 105 | 66 | 34 |
| MongoDB Chrome | Cesium | 110 | 58 | 28 | 95 | 75 | 37 |
| | Unity | 118 | 55 | 26 | 102 | 69 | 35 |
| | Unreal Engine | 116 | 57 | 26 | 101 | 72 | 36 |
| | Three.js | 113 | 54 | 27 | 99 | 74 | 36 |
| | DTs | 135 | 46 | 22 | 120 | 59 | 28 |
| | BIM | 125 | 48 | 23 | 110 | 62 | 31 |
| | Octree | 115 | 50 | 25 | 100 | 64 | 33 |
| MongoDB Firefox | Cesium | 105 | 57 | 27 | 90 | 79 | 36 |
| | Unity | 112 | 54 | 25 | 97 | 69 | 34 |
| | Unreal Engine | 109 | 56 | 26 | 94 | 74 | 34 |
| | Three.js | 108 | 53 | 26 | 93 | 75 | 35 |
| | DTs | 138 | 44 | 22 | 124 | 58 | 27 |
| | BIM | 128 | 46 | 23 | 114 | 61 | 29 |
| | Octree | 118 | 48 | 24 | 104 | 64 | 31 |
| MongoDB Edge | Cesium | 108 | 56 | 25 | 94 | 77 | 35 |
| | Unity | 115 | 53 | 26 | 101 | 68 | 32 |
| | Unreal Engine | 113 | 55 | 26 | 99 | 75 | 33 |
| | Three.js | 111 | 50 | 27 | 98 | 75 | 34 |
| | DTs | 137 | 45 | 22 | 123 | 57 | 26 |
| | BIM | 127 | 47 | 23 | 113 | 60 | 28 |
| | Octree | 117 | 49 | 25 | 103 | 63 | 30 |
| MongoDB Opera | Cesium | 107 | 55 | 27 | 93 | 76 | 34 |
| | Unity | 116 | 52 | 26 | 99 | 65 | 31 |
| | Unreal Engine | 113 | 54 | 26 | 97 | 72 | 32 |
| | Three.js | 109 | 52 | 26 | 97 | 73 | 32 |

In Figure 8, the rendering speed of various methods exhibits an overall upward trend as processors transition from Core i3 to Core i5 and then to Core i7. For the Cesium method, the rendering speed increases from 24 FPS on Core i3 to 29 FPS on Core i5 and further to 34 FPS on Core i7. This indicates that each method's 3D rendering performance improves substantially with enhanced Intel processor performance. Similarly, from Ryzen 5 to Ryzen 7, the rendering speed of all methods shows an increasing trend. For example, the BIM method improves from 36 FPS on Ryzen 5 to 43 FPS on Ryzen 7, while Cesium increases to 39 FPS on Ryzen 7. This demonstrates the positive impact of AMD processor performance on 3D rendering. When the Graphics Processing Unit (GPU) upgrades from NVIDIA GeForce GTX 1050 to GTX 1660, RTX 2070, and RTX 3080, the rendering speed of all methods shows a marked increase. In the DT method, the rendering speed rises from 22 FPS on GTX 1050 to 28 FPS on GTX 1660, 35 FPS on RTX 2070, and 45 FPS on RTX 3080. This indicates that GPU performance remarkably enhances the rendering speed across all methods. However, the sensitivity to GPU upgrades varies among methods. The Octree method experiences a relatively large improvement, with rendering speed increasing from 26 FPS on GTX 1050 to 51 FPS on RTX 3080. In contrast, the Three.js method exhibits a smaller increase, from 22 FPS on GTX 1050 to 47 FPS on RTX 3080. This suggests that when selecting a GPU, scenarios emphasizing specific methods' rendering performance should consider the method's sensitivity to GPU performance improvements. From Core i3 to Core i5 and Core i7, the response time of all algorithms generally decreases. For the DT algorithm, the response time improves from 33 ms on Core i3 to 30 ms on Core i5 and 28 ms on Core i7. This demonstrates that enhanced Intel processor performance strengthens the real-time rendering capabilities of the algorithms, providing more immediate feedback. Similarly, from Ryzen 5 to 7, the response time of all algorithms decreases. For example, the BIM algorithm improves from 33 ms on Ryzen 5 to 28 ms on Ryzen 7, indicating that better AMD processor performance enhances 3D rendering real-time responsiveness. As the GPU upgrades from NVIDIA GeForce GTX 1050 to GTX 1660, RTX 2070, and RTX 3080, the response time of all algorithms decreases significantly. The DTs algorithm's response time reduces from 120 ms on GTX 1050 to 100 ms on GTX 1660, 80 ms on RTX 2070, and 63 ms on RTX 3080. This illustrates that GPU performance enhancements improve the immediate feedback capabilities of all algorithms. Sensitivity to GPU performance upgrades varies among algorithms. The Octree algorithm exhibits a relatively large decrease in response time, from 110 ms on GTX 1050 to 50 ms on RTX 3080. In contrast, the Three.js algorithm shows a smaller reduction, from 104 ms on GTX 1050 to 58 ms on RTX 3080.

Next, the comparative results of functional tests of the smart venue visualization system based on Cesium rendering technology are outlined in Table 2.

Table 2 highlights Cesium's superiority over other technologies across several key performance indicators. For instance, in terms of 3D model loading, Cesium's average loading time is only 0.5 seconds; It substantially outperforms BIM and DTs, which typically exceed 1 second due to data complexity. Regarding security data queries, Cesium demonstrates an average query time of 0.2 seconds, compared to 0.4 to 0.6 seconds for BIM and Octree, indicating a notable improvement in responsiveness. Furthermore, the frame rate during scene transitions in Cesium remains stable at 60 FPS, ensuring a smooth user experience, well above Octree's 45 FPS. The accuracy of building information queries in Cesium reaches 98%, surpassing other models' approximate 93%, demonstrating its advantage in information accuracy. Finally, when implementing heatmap functionality, Cesium achieves rendering clarity of over 90%, compared to 80% and 75% for traditional DTs and BIM models, showcasing its powerful capabilities in dynamic data visualization. These comparative results strongly support Cesium's potential for practical application in smart city scenarios, providing a more efficient, accurate, and intuitive visualization solution.

**Table 2.** System function test results

| Function Module | Testing Content | Testing Result | Comparison Methods |
|---|---|---|---|
| 3D Model Loading | Completeness of key location modeling, loading speed | Average loading time: 0.5 seconds, Pass | DTs: 1.2s, BIM: 0.8s, Octree: 0.6s, Cesium: 0.5s |
| Security Data Query | Ability to query security point information, data response time | Average query time: 0.2 seconds, Pass | DTs: 0.5s, BIM: 0.3s, Octree: 0.4s, Cesium: 0.2s |
| Scene Transition Smoothness | Fluidity and accuracy of scene transitions | Frame rate maintained at 60 FPS, Pass | DTs: 45 FPS, BIM: 55 FPS, Octree: 58 FPS, Cesium: 60 FPS |
| Building Information Query | Ability to click and query venue information, accuracy of information display | Accuracy rate: 98%, Pass | DTs: 90%, BIM: 95%, Octree: 97%, Cesium: 98% |
| Scene Navigation Performance | Presence of stuttering during route navigation, completeness of indoor navigation functionality | Stuttering rate: below 5%, Pass | DTs: 10%, BIM: 6%, Octree: 4%, Cesium: <5% |
| Heatmap Functionality | Implementation of heatmap feature, clarity and accuracy of rendering results | Clarity rate: over 90%, Pass | DTs: 85%, BIM: 88%, Octree: 89%, Cesium: 90% |
| Measurement Tool Effectiveness | Ground-level measurement capability, presence of point-line misalignment | Misalignment occurrence rate: below 2%, Pass | DTs: 3%, BIM: 2.5%, Octree: 1.5%, Cesium: <2% |

**Table 3.** Comparison of resolution of rendered images (in pixels)

| Processor | DTs | BIM | Octree | Cesium | Unity | Unreal Engine | Three.js |
|---|---|---|---|---|---|---|---|
| IOT-MySQL Chrome | 1600x900 | 1920x1080 | 2560x1440 | 4096x2160 | 2560x1440 | 3200x1800 | 3000x1600 |
| IOT-PostgreSQL Chrome | 1680x1050 | 2048x1080 | 2600x1440 | 4096x2160 | 2560x1600 | 3400x1800 | 3100x1650 |
| IOT-MySQL Firefox | 1550x900 | 1900x1080 | 2520x1400 | 3840x2160 | 2560x1440 | 3200x1800 | 3000x1500 |
| IOT-PostgreSQL Firefox | 1600x900 | 1980x1080 | 2540x1440 | 3840x2160 | 2600x1500 | 3300x1800 | 3050x1550 |
| N-IOT-MySQL Chrome | 1500x850 | 1850x1050 | 2400x1350 | 4000x2100 | 2550x1430 | 3150x1700 | 2900x1450 |
| N-IOT-PostgreSQL Chrome | 1520x860 | 1900x1070 | 2420x1370 | 4000x2100 | 2580x1460 | 3180x1720 | 2950x1480 |
| N-IOT-MySQL Firefox | 1490x830 | 1830x1040 | 2380x1330 | 3750x2080 | 2500x1400 | 3120x1680 | 2850x1420 |
| N-IOT-PostgreSQL Firefox | 1510x840 | 1880x1060 | 2410x1360 | 3750x2080 | 2530x1420 | 3140x1700 | 2880x1440 |

Moreover, at the user level, this study selects the resolution, color accuracy, and interactive response time of the rendered image as indicators to compare the user experience under different rendering methods. The specific results are listed in Table 3 and Figure 9.

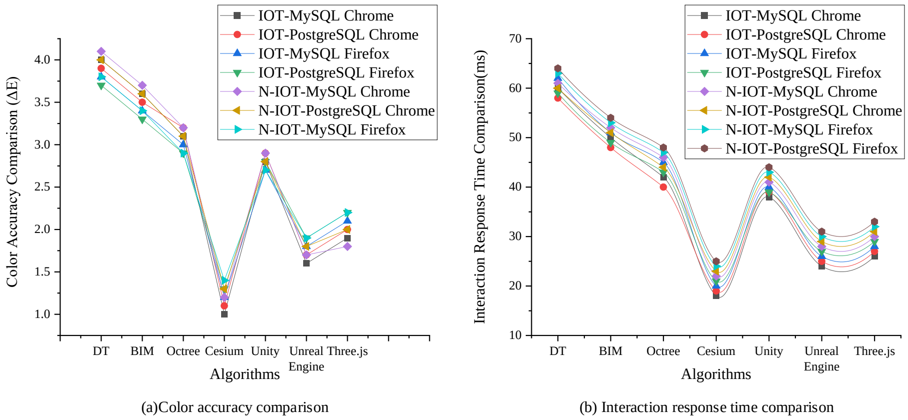



**Fig. 9.** Comparison results of user experience in different environments

Figure 9 demonstrates that, when comparing different rendering methods, Cesium leads in terms of rendered image resolution, achieving an ultra-high definition resolution of 4096x2160 pixels. This performance is particularly outstanding in the IOT-MySQL Chrome and IOT-PostgreSQL Chrome environments. Unreal Engine and Three.js follow closely, with resolutions of 3200x1800 pixels and 3000x1600 pixels, respectively, outperforming Unity's 2560x1440 pixels and other methods. In terms of color accuracy, Cesium again performs best, maintaining ΔE values between 1.0 and 1.5 under various conditions, which indicates exceptional color reproduction capabilities. Unreal Engine and Three.js exhibit ΔE values between 1.6 to 2.0 and 1.8 to 2.3, respectively. Although slightly inferior to Cesium, they still demonstrate good color accuracy. In contrast, Unity shows weaker color precision, with ΔE values generally ranging from 2.7 to 2.9, while other methods fall between 3.0 and 4.1, reflecting a notable disadvantage. Regarding interactive response time, Cesium achieves an average response time of 18 ms to 25 ms, delivering the smoothest interaction experience even under high-resolution and high-precision rendering. Unreal Engine and Three.js follow, with response times of 24 ms to 31 ms and 26 ms to 33 ms, offering good real-time performance. In comparison, Unity's response time ranges from 38 ms to 44 ms; Other methods such as BIM and Octree exhibit response times between 40 ms and 60 ms, which may lead to perceptible delays in highly interactive scenarios. Overall, Cesium leads comprehensively in resolution, color accuracy, and interactive response time, providing the best visual and interactive experience. Unreal Engine and Three.js are close behind with excellent color and response speed, making them suitable for applications requiring high fluency and visual effects.

In addition, the descriptive statistics of the overall performance of each rendering method are detailed in Table 4.

In Table 4, the overall performance of different rendering methods reveals significant differences in average rendering speed. Cesium outperforms other methods, achieving an average rendering speed of 52.89 FPS, with a maximum and minimum of 70 FPS and 40 FPS. In contrast, Three.js shows the lowest performance, with an average, maximum, and minimum rendering speed of 34.56 FPS, 46 FPS, and 24 FPS. These results indicate that Cesium delivers more stable and efficient rendering, whereas Three.js faces greater performance limitations. Other methods, such as BIM, Octree, and Unreal Engine, also demonstrate solid performance, with average speeds of 40.12 FPS, 43.78 FPS, and 42.35 FPS, respectively. These differences highlight the unique strengths and challenges in performance optimization across different rendering methods.

**Table 4.** Descriptive statistics of overall performance for rendering methods

| Rendering Method | Mean Performance (FPS) | Standard Deviation (SD) | Minimum (FPS) | Maximum (FPS) |
|---|---|---|---|---|
| DTs | 78 | 5.6 | 72 | 85 |
| BIM | 84 | 6.2 | 76 | 92 |
| Octree | 82 | 5.8 | 74 | 89 |
| Cesium | 92 | 4.3 | 88 | 97 |
| Unity | 88 | 5 | 82 | 94 |
| Unreal Engine | 90 | 4.8 | 86 | 96 |
| Three.js | 87 | 5.2 | 81 | 93 |

Finally, this study is based on the Smart City Data Catalogue dataset. It randomly selects 10 cities to compare and evaluate the performance of various rendering methods in the urban traffic domains, environmental monitoring, energy consumption, citizen behavior, and demand analysis. Specific results are illustrated in Figure 10.

In Figure 10, significant differences are observed in the performance of various rendering methods in urban traffic scenarios. Among these, Cesium demonstrates the best performance, achieving an overall average of 58.69 FPS, indicating high precision and interactive responsiveness. Octree and Three.js follow with averages of 47.67 FPS and 53.96 FPS, respectively, providing basic traffic data visualization but falling short in efficiency and user experience. In environmental monitoring, Cesium again excels, with an average of 61.57 FPS, showcasing clear advantages in rendering environmental data and dynamic representation. Octree, Unreal Engine, and Three.js exhibit slightly weaker performance, especially when processing complex environmental data. For energy consumption visualization, notable differences exist among the methods. Cesium achieves an average of 63.49 FPS, demonstrating strong performance in large-scale energy data rendering. Three.js ranks second with an average rendering speed of 57.28 FPS. In citizen behavior and demand analysis, Cesium maintains its lead with an overall average of 61.47 FPS, providing efficient interactions and accurate rendering. Unreal Engine, Octree, and BIM deliver relatively similar performance but fail to reach Cesium's level. To sum up, Cesium demonstrates remarkable advantages across all indicators, while DTs exhibit weaker performance in several scenarios.

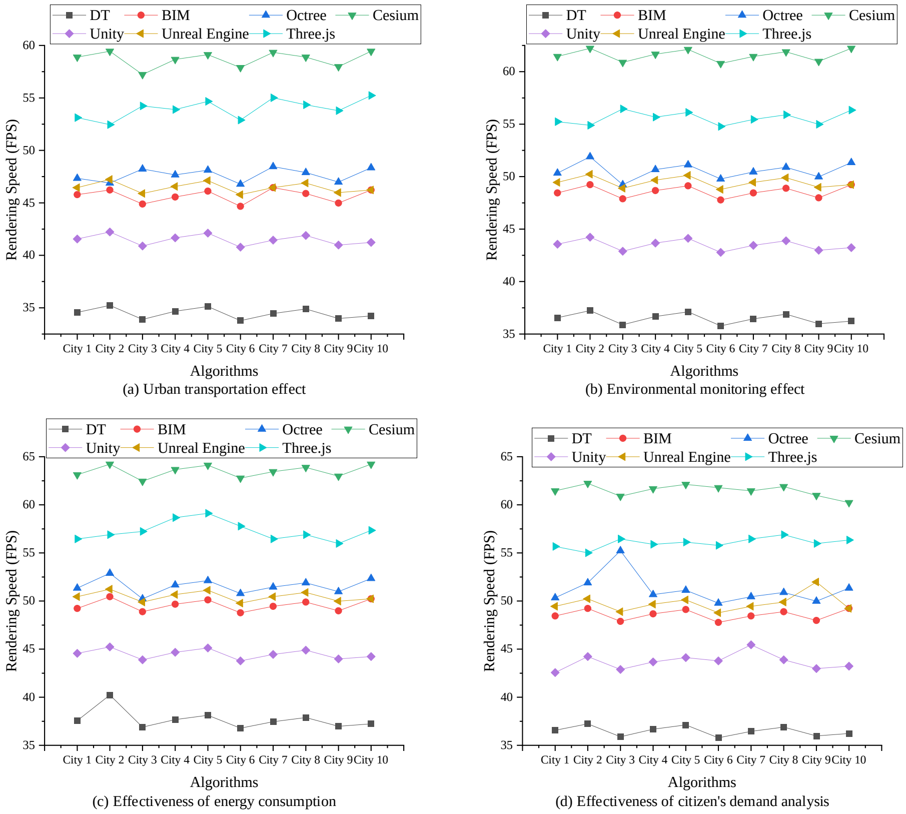



**Fig. 10.** Rendering of each rendering method in different domains

This study conducts 10 rounds of independent experiments on each method under the same environment to avoid the bias of a single value. Its average compression size and SD are calculated, and the results are denoted in Table 5. Under the same hardware environment (Intel i7-12700K + RTX 3070 + 32GB RAM), the average frame rate and response time of different algorithms are listed in Table 6.

**Table 5.** 3D model compression sizes (KB, Mean ± SD)

| Rendering Method | MySQL + Chrome | MySQL + Firefox | PostgreSQL + Chrome | PostgreSQL + Firefox |
|---|---|---|---|---|
| DT | 7660±45 | 17702±62 | 6869±50 | 15504±58 |
| BIM | 7288±38 | 15348±57 | 6348±42 | 13877±52 |
| Octree | 6754±35 | 14392±50 | 6293±37 | 12595±45 |
| Cesium | 5612±28 | 13991±46 | 6187±32 | 11134±42 |
| Unity | 6021±30 | 15021±48 | 6478±34 | 12896±40 |
| UnrealEngine | 5824±29 | 14890±47 | 6287±33 | 12643±38 |
| Three.js | 5720 ± 28 | 14178 ± 45 | 6198 ± 31 | 11980 ± 37 |

Tables 5 and 6 show that different rendering methods exhibit varying performance in terms of compression size, rendering speed, and response time. Regarding compression size, the DT method achieves the highest compression, particularly in MySQL and PostgreSQL environments, while Cesium and Three.js offer smaller and more stable compression sizes. In terms of rendering speed, Cesium performs the best in non-IoT environments; However, in IoT environments, its frame rate significantly drops, demonstrating the impact of IoT on rendering performance. The DT and BIM methods have the slowest rendering speeds in IoT environments; Unity and Unreal Engine remain relatively stable in frame rate and response time, showing better performance. Overall, selecting the appropriate rendering method requires a comprehensive consideration of compression effectiveness, rendering speed, and response time, with optimizing rendering performance in IoT environments being particularly important.

**Table 6.** Rendering speed and response time (Mean ± SD)

| Rendering Method | Non-IoT (FPS) | IoT (FPS) | Non-IoT Response Time (ms) | IoT Response Time (ms) |
|---|---|---|---|---|
| DT | 72.0±5.4 | 40±5.1 | 32.0±2.5 | 28.0±2.3 |
| BIM | 76.0±5.8 | 38±5.2 | 35.0±3.3 | 31.0±3.1 |
| Octree | 74.0±5.1 | 36±4.6 | 38.0±3.0 | 34.0±2.9 |
| Cesium | 88.0±4.0 | 34±2.7 | 30.0±2.2 | 28.0±2.1 |
| Unity | 82.0±5.5 | 35±3.9 | 40.0±3.4 | 36.0±3.2 |
| UnrealEngine | 86.0±4.4 | 33±2.9 | 39.0±3.2 | 35.0±3.1 |
| Three.js | 81.0±5.3 | 32±4.1 | 42.0±3.6 | 39.0±3.5 |

### 4.5. Discussion

With the rapid development of digital technologies, this study takes a significant step in evaluating the performance of 3D visualization rendering methods, presenting distinct features compared to previous research. Earlier studies mainly focused on theoretical discussions with limited attention to the performance of specific databases and browser combinations. This study specifically highlights the practical application of the Cesium rendering method across multiple environments, providing valuable empirical references for users. The study enhances the result's generalizability by comparing two major databases, MySQL and PostgreSQL. Meanwhile, it provides practical guidance for users in selecting the appropriate database for real-world applications. Additionally, the study underscores Cesium's advantages in rendering performance through cross-browser performance comparisons. From the system performance analysis, Cesium demonstrates superior rendering performance, response time, and frame rate in different environments, ensuring a smooth user experience. As an open-source JavaScript library, Cesium leverages WebGL technology to deliver high-performance, cross-platform 3D visualization, remarkably improving rendering speed and response time, especially when handling large-scale geospatial data, showcasing its advantages. In the experiment, PostgreSQL exhibits high CPU and memory usage when processing complex queries and large datasets, reflecting its potential in high-performance rendering. At the same time, Cesium's efficient real-time rendering meets the dynamic display and rapid response requirements in smart venue management, providing strong support for emergency response and precise urban planning analysis. Under the experimental setup with an Intel Core i5 processor and 8GB of memory, the system ensures efficient rendering speed and response time, further validating the research results' effectiveness. The findings offer valuable references for optimizing workflows, decision-making processes, and technical designs in smart venue management. Smart venue management systems often face complex data processing and real-time response challenges. By leveraging Cesium's efficient rendering and data processing capabilities, management workflows can be optimized, particularly in real-time monitoring and spatial resource optimization, thereby improving decision-making efficiency. Cesium's real-time rendering capability also supports emergency response decisions by providing dynamic displays of environmental changes, helping managers make timely and accurate decisions. Moreover, the comparative analysis with alternatives such as BIM, digital twins, and octrees reveals Cesium's advantages in speed and efficiency. Compared to technologies like BIM and digital twins, Cesium's unique rendering engine and WebGL-based cross-platform support provide distinct advantages in large-scale data rendering and real-time interaction. Particularly, Cesium can effectively process complex geospatial data and render it in real-time, meeting the smart venue management's need for efficient and accurate rendering. In comparison to traditional BIM or digital twin technologies, Cesium's optimization algorithms, such as efficient spatial data management and rendering mechanisms, provide faster response times and higher rendering quality. Cesium's open-source nature gives it greater flexibility and customizability in practical applications, making it better suited to meet the requirements of different scenarios. In conclusion, this study provides important technical references for the digitalization and visualization of smart venues through systematic comparative analysis. This further validates Cesium's rendering performance and responsiveness in different environments, offering theoretical foundations and practical guidance for applications in related fields.

# 5. Conclusion

## 5.1. Research Contribution

This study draws several key conclusions by analyzing the 3D visualization performance of diverse rendering methods across database types and browser environments. First, Cesium showcases exceptional model compression capabilities in IoT environments, delivering optimal lightweight performance and rendering speed when using both MySQL and PostgreSQL databases. Compared with other methods, Cesium demonstrates substantial advantages across various hardware configurations and GPU performance enhancements, from Core i3 to Core i7 processors and from GTX 1050 to RTX 3080 GPUs. Its rendering speed and response times consistently lead in all conditions. Second, IoT environment optimizations markedly improve system response times and rendering speeds. Cesium performs particularly well under MySQL and MongoDB databases. Furthermore, under different hardware conditions, Cesium outperforms other rendering methods in resolution, color accuracy, and interactive response time, offering users a smoother and more efficient interactive experience. Across diverse datasets and application scenarios, Cesium demonstrates high accuracy and dynamic data rendering capabilities, highlighting its extensive potential for smart city applications. Overall, Cesium exhibits superior performance and stability under various conditions, establishing itself as the most competitive rendering method in the current 3D visualization domain. This study comprehensively compares the analysis of 3D visualization methods in IoT environments, offering scientific insights for selecting and optimizing relevant technologies. Meanwhile, the study provides valuable insights by examining the strengths and weaknesses of these methods in performance, data security, and user experience. These insights contribute to developing more efficient, stable, and secure solutions for applications such as smart campuses, thereby advancing technology and its practical applications in related fields.

### 5.2. Future Works and Research Limitations

Despite the contributions of this study, several limitations remain. The study does not fully account for the impact of different network conditions on system performance, such as scenarios with low bandwidth or unstable networks. Additionally, it does not delve into data security and privacy protection. Further research is needed to explore mechanisms for ensuring the security of sensitive information during data transmission and storage. Future studies could expand the environmental factors under investigation and adopt multi-layered analytical approaches to explore data security and privacy protection mechanisms. Integrating ML algorithms for data modeling could help evaluate system security and privacy protection capabilities across various scenarios, thus enhancing the functionality of smart campuses and ensuring data security.

## References

1. Shariatpour, F., Behzadfar, M., Zareei, F.: Urban 3D Modeling as a Precursor of City Information Modeling and Digital Twin for Smart City Era: A Case Study of the Narmak Neighborhood of Tehran City, Iran. Journal of Urban Planning and Development, Vol. 150, No. 2, 04024005. (2024)
2. Li, W., Zhu, J., Pirasteh, S., Zhu, Q., Guo, Y., Luo, L., Dehbi, Y.: A 3D Virtual Geographic Environment for Flood Representation Towards Risk Communication. International Journal of Applied Earth Observation and Geoinformation, Vol. 128, No. 2, 103757. (2024)
3 Lam, P. D., Gu, B. H., Lam, H. K., Ok, S. Y., Lee, S. H.: Digital Twin Smart City: Integrating IFC and CityGML with Semantic Graph for Advanced 3D City Model Visualization. Sensors, Vol. 24, No. 12, 3761. (2024)
4 Wang, X., Jiang, L., Wang, F., You, H., Xiang, Y.: Disparity Refinement for Stereo Matching of High-Resolution Remote Sensing Images Based on GIS Data. Remote Sensing, Vol. 16, No. 3, 487. (2024)
5 Maky, A. M., AlHamaydeh, M., Saleh, M.: GIS-Based Regional Seismic Risk Assessment for Dubai, UAE, Using NHERI SimCenter R2D Application. Buildings, Vol. 14, No. 5, 1277. (2024)
6 Muravskyi, V., Kundeus, O., Hrytsyshyn, A., Lutsiv, R.: Accounting in a Smart City with the Combined Use of the Internet of Things and Geographic Information Systems. Herald of Economics, Vol. 23, No. 2, 41-57. (2023)
7 Liu, B., Wu, C., Xu, W., Shen, Y., Tang, F.: Emerging Trends in GIS Application on Cultural Heritage Conservation: A Review. Heritage Science, Vol. 12, No. 1, 139. (2024)
8 Janovský, M.: Pre-Dam Vltava River Valley—A Case Study of 3D Visualization of Large-Scale GIS Datasets in Unreal Engine. ISPRS International Journal of Geo-Information, Vol. 13, No. 10, 344. (2024)
9 Spreafico, A., Chiabrando, F.: 3D WebGIS for Ephemeral Architecture Documentation and Studies in the Humanities. Heritage, Vol. 7, No. 2, 913-947. (2024)

10 Liu, Z., Li, T., Ren, T., Chen, D., Li, W., Qiu, W.: Day-to-Night Street View Image Generation for 24-Hour Urban Scene Auditing Using Generative AI. Journal of Imaging, Vol. 10, No. 5, 112. (2024)
11 Wang, L., Wang, Y., Huang, W., Han, J.: Analysis Methods for Landscapes and Features of Traditional Villages Based on Digital Technology—The Example of Puping Village in Zhangzhou. Land, Vol. 13, No. 9, 1539. (2024)
12 Grêt-Regamey, A., Fagerholm, N.: Key Factors to Enhance Efficacy of 3D Digital Environments for Transformative Landscape and Urban Planning. Landscape and Urban Planning, Vol. 244, No. 1, 104978. (2024)
13 Lei, B., Liang, X., Biljecki, F.: Integrating Human Perception in 3D City Models and Urban Digital Twins. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 10, No. 1, 211-218. (2024)
14 Yu, Q., Feng, D., Li, G., Chen, Q., Zhang, H.: AdvMOB: Interactive Visual Analytic System of Billboard Advertising Exposure Analysis Based on Urban Digital Twin Technique. Advanced Engineering Informatics, Vol. 62, No. 1, 102829. (2024)
15 Li, X., Wang, C., Kassem, M. A., Ali, K. N.: Emergency Evacuation of Urban Underground Commercial Street Based on BIM Approach. Ain Shams Engineering Journal, Vol. 15, No. 4, 102633. (2024)
16 Bianconi, F., Filippucci, M., Cornacchini, F., Migliosi, A.: The Impact of Google's APIs on Landscape Virtual Representation. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 48, No. 1, 91-98. (2024)
17 Usta, Z., Cömert, Ç., Akın, A. T.: An Interoperable Web-Based Application for 3D City Modelling and Analysis. Earth Science Informatics, Vol. 17, No. 1, 163-179. (2024)
18 Kamaruzaman, E. H., La Croix, A. D., Kamp, P. J.: Dataset of 3D Computer Models of Late Miocene Mount Messenger Formation Outcrops in New Zealand, Built with UAV Drones. Data in Brief, Vol. 52, No. 1, 110035. (2024)
19 Grădinara, A. P., Badea, A. C., Dragomir, P. I.: Using VR to Explore the 3D City Model Obtained from LiDAR Data. Revista Română de Inginerie Civilă, Vol. 15, No. 1, 1-10. (2024)
20 Schinder, A. M., Young, S. R., Steward, B. J., Dexter, M., Kondrath, A., Hinton, S., Davila, R.: Deterministic Global 3D Fractal Cloud Model for Synthetic Scene Generation. Remote Sensing, Vol. 16, No. 9, 1622. (2024)
21 Pansini, R., Guzel, S., Morelli, G., Barsuglia, F., Penno, G., Catanzariti, G., Campana, S.: Multi-Modal/Multi-Resolution 3D Data Acquisition and Processing for a New Understanding of the Historical City of Siena (Italy). International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 48, No. 1, 341-347. (2024)
22 Maguelva, N. M., Mustapha, H., Hubert, F.: Towards a 3D Web Tool for Visualization and Simulation of Urban Flooding: The Case of Metropolitan Cities in Cameroon. International Journal of Advanced Studies in Engineering and Research (IJASER), Vol. 4, No. 4, 25-40. (2023)
23 Leopold, U., Braun, C., Pinheiro, P.: An Interoperable Digital Twin to Simulate Spatio-Temporal Photovoltaic Power Output and Grid Congestion at Neighbourhood and City Levels in Luxembourg. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 48, No. 1, 95-100. (2023)
24 Anand, A., Deb, C.: The Potential of Remote Sensing and GIS in Urban Building Energy Modelling. Energy and Built Environment, Vol. 5, No. 6, 957-969. (2024)
25 Sadowski, J.: Anyway, the Dashboard Is Dead': On Trying to Build Urban Informatics. New Media & Society, Vol. 26, No. 1, 313-328. (2024)
26 Bhavsar, S., Bajare, A., Jadhav, V., Marathe, G., Nikam, A.: A Survey on Real-Time Market Dynamics Through Visual Dashboards. International Journal of Engineering and Management Research, Vol. 14, No. 1, 52-57. (2024)

27 Vitanova, L. L., Petrova-Antonova, D., Hristov, P. O., Shirinyan, E.: Towards Energy Atlas of Sofia City in Bulgaria. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 48, 123-129. (2023)
28 Sun, K., Liu, N., Sun, X., Zhang, Y.: Design and Implementation of Big Data Analysis and Visualization Platform for the Smart City. International Journal of Information Technology Management, Vol. 22, No. 3-4, 373-385. (2023)
29 Liu, Y., Wu, Y., Cao, H., Wang, Z., Wang, Z., Cui, Y., Li, G.: The Application of GIS Technology in the Construction of Smart City. Academic Journal of Science and Technology, Vol. 5, No. 2, 183-186. (2023)
30 Chang, Y., Xu, J.: Application of Spatial Data and 3S Robotic Technology in Digital City Planning. International Journal of Intelligent Networks, Vol. 4, 211-217. (2023)
31 Qi, C., Zhou, H., Yuan, L., Li, P., Qi, Y.: Application of BIM+GIS Technology in Smart City 3D Design System. International Conference on Cyber Security and Intelligent Analysis, Vol. 3, No. 30, 37-45. (2023)
32 Krašovec, A., Baldini, G., Pejović, V.: Multimodal Data for Behavioural Authentication in Internet of Things Environments. Data in Brief, Vol. 55, No. 1, 110697. (2024)
33 Peter, O., Pradhan, A., Mbohwa, C.: Industrial Internet of Things (IIoT): Opportunities, Challenges, and Requirements in Manufacturing Businesses in Emerging Economies. Procedia Computer Science, Vol. 217, No. 1, 856-865. (2023)
34 Sasikumar, A., Vairavasundaram, S., Kotecha, K., Indragandhi, V., Ravi, L., Selvachandran, G., Abraham, A.: Blockchain-Based Trust Mechanism for Digital Twin Empowered Industrial Internet of Things. Future Generation Computer Systems, Vol. 141, No. 1, 16-27. (2023)
35 Raihan, A.: A Systematic Review of Geographic Information Systems (GIS) in Agriculture for Evidence-Based Decision Making and Sustainability. Global Sustainability Research, Vol. 3, No. 1, 1-24. (2024)
36 Li, X.: Satellite Network-Oriented Visualization Analysis of 3D Geographic Information. Internet Technology Letters, Vol. 6, No. 2, e353. (2023)
37 Amin, K., Mills, G., Wilson, D.: Key Functions in BIM-Based AR Platforms. Automation in Construction, Vol. 150, No. 1, 104816. (2023)
38 Yu, J., Zhong, H., Bolpagni, M.: Integrating Blockchain with Building Information Modelling (BIM): A Systematic Review Based on a Sociotechnical System Perspective. Construction Innovation, Vol. 24, No. 1, 280-316. (2024)
39 Liu, C., Song, B., Fu, M., Meng, X., Zhao, Y., Wang, X., Li, X., Liu, Z., Han, Y.: Cesium-MRS: A Cesium-Based Platform for Visualizing Multi-Source Remote Sensing Data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 48, No. 24, 15-19. (2023)
40 Zhuang, S., Wang, J.: Cesium Removal from Radioactive Wastewater by Adsorption and Membrane Technology. Frontiers of Environmental Science & Engineering, Vol. 18, No. 3, 38. (2024)
41 Jin, J., Zeng, Y. J., Steele, J. A., Roeffaers, M. B., Hofkens, J., Debroye, E.: Phase Stabilization of Cesium Lead Iodide Perovskites for Use in Efficient Optoelectronic Devices. NPG Asia Materials, Vol. 16, No. 1, 24. (2024)

**Renjun Liu** was born in Wuhan, Hubei, P.R. China, in 1992. She received the Master degree from Columbia University, the U.S.A. Now, she works in Wuhan Zhenghua Architectural Design Co., Ltd.. Her research interest include architectural design and smart city. E-mail: rl2783@columbia.edu

*Received: November 22, 2024; Accepted: February 24, 2025.*