

Article

Construction of a Real-Scene 3D Digital Campus Using a Multi-Source Data Fusion: A Case Study of Lanzhou Jiaotong University

Rui Gao ¹, Guanghui Yan ^{1,*}, Yingzhi Wang ¹, Tianfeng Yan ¹, Ruiting Niu ¹ and Chunyang Tang ²

¹ School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China; gaorui@lztu.edu.cn (R.G.)

² Silk Road Fantian (Gansu) Communication Technology Co., Ltd., Lanzhou 730070, China

* Correspondence: ghyan@mail.lztu.cn

Abstract: Real-scene 3D digital campuses are essential for improving the accuracy and effectiveness of spatial data representation, facilitating informed decision-making for university administrators, optimizing resource management, and enriching user engagement for students and faculty. However, current approaches to constructing these digital environments face several challenges. They often rely on costly commercial platforms, struggle with integrating heterogeneous datasets, and require complex workflows to achieve both high precision and comprehensive campus coverage. This paper addresses these issues by proposing a systematic multi-source data fusion approach that employs open-source technologies to generate a real-scene 3D digital campus. A case study of Lanzhou Jiaotong University is presented to demonstrate the feasibility of this approach. Firstly, oblique photography based on unmanned aerial vehicles (UAVs) is used to capture large-scale, high-resolution images of the campus area, which are then processed using open-source software to generate an initial 3D model. Afterward, a high-resolution model of the campus buildings is then created by integrating the UAV data, while 3D Digital Elevation Model (DEM) and OpenStreetMap (OSM) building data provide a 3D overview of the surrounding campus area, resulting in a comprehensive 3D model for a real-scene digital campus. Finally, the 3D model is visualized on the web using Cesium, which enables functionalities such as real-time data loading, perspective switching, and spatial data querying. Results indicate that the proposed approach can effectively get rid of reliance on expensive proprietary systems, while rapidly and accurately reconstructing a real-scene digital campus. This framework not only streamlines data harmonization but also offers an open-source, practical, cost-effective solution for real-scene 3D digital campus construction, promoting further research and applications in twin city, Virtual Reality (VR), and Geographic Information Systems (GIS).

Keywords: 3D digital campus; UAV-based oblique photography; multi-source data fusion; Cesium; 3D modeling



Academic Editors: Wolfgang Kainz, Junxing Zheng and Peng Cao

Received: 11 November 2024

Revised: 30 December 2024

Accepted: 30 December 2024

Published: 3 January 2025

Citation: Gao, R.; Yan, G.; Wang, Y.; Yan, T.; Niu, R.; Tang, C. Construction of a Real-Scene 3D Digital Campus Using a Multi-Source Data Fusion: A Case Study of Lanzhou Jiaotong University. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 19. <https://doi.org/10.3390/ijgi14010019>

Copyright: © 2025 by the authors.

Published by MDPI on behalf of the

International Society for

Photogrammetry and Remote Sensing.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license

([https://creativecommons.org/](https://creativecommons.org/licenses/by/4.0/)

[licenses/by/4.0/](https://creativecommons.org/licenses/by/4.0/)).

1. Introduction

With the rapid advancement of scientific research and technology, three-dimensional (3D) visualization technologies have become increasingly important. Studies have shown that photorealistic 3D models are essential components of geospatial infrastructure for a variety of fields, including urban planning, green space management, urban mapping, and urban monitoring [1–3]. In urban planning, 3D visualization technologies provide

intuitive and detailed geographic information, which facilitates the more scientific design of urban spaces and assists decision-makers in resource allocation [4]. Furthermore, these technologies play a critical role in enhancing public engagement, improving data transparency, and fostering social collaboration [5]. The integration of 3D visualization technologies with geographic information systems (GIS) has led to the development of novel WebGIS-based real-scene 3D visualization frameworks. This advancement offers researchers and practitioners a more accessible alternative to traditional PC client-based GIS and 3D visualization software, which often face limitations in sharing capabilities and collaborative geospatial analysis. Additionally, these WebGIS-based 3D techniques enable users to interact with geospatial data in a virtual 3D environment, recognized as an effective approach for advanced 3D visualization and analysis [6,7]. Among the various application domains, university campuses represent a critical area where such technologies have the potential to address unique management and planning challenges, though they have not been as extensively explored as other fields.

Contemporary university management and planning face several challenges, including the need for precise spatial data representation, efficient resource management, and effective infrastructure planning. Traditional two-dimensional (2D) GIS often fall short in addressing these needs due to their limited spatial detail and lack of interactivity. The construction of high-resolution, real-scene 3D digital campus models addresses these urgent needs by providing detailed spatial information that enhances campus management, facilitates precise planning, and optimizes resource allocation. High-resolution 3D models enable accurate facility maintenance, asset management, and emergency response planning. In campus planning, these models support the design and optimization of new buildings and infrastructure, promoting sustainable development and efficient resource utilization. Additionally, 3D digital campuses contribute to advanced energy management and resource optimization, leading to reduced operational costs and minimized environmental impact. For students and faculty, immersive 3D environments enhance user engagement and interaction with campus spaces, improving overall user experience and fostering a sense of community.

High-resolution 3D digital campus models are indispensable for effective campus management and planning as they provide the necessary detail and accuracy for precise monitoring and optimization of resources. However, developing a real-scene 3D digital campus platform based on WebGIS presents several challenges. One primary challenge is that most existing 3D geographic information platforms are commercial and costly, placing a substantial financial burden on researchers and educators. This financial strain not only hinders the development and dissemination of these platforms but also limits the exchange and sharing of technology. The second challenge is the difficulty and time-consuming nature of creating high-quality 3D real-scene models. Many commercial 3D modeling tools are highly complex, and mastering modeling techniques requires extensive training, making them hard to use efficiently. The third challenge stems from the diversity and heterogeneity of data sources for constructing a real-scene 3D geographic information platform. Data obtained from various software, databases, and storage structures—including oblique photography, 3D models (OBJ, OSGB, FBX format files), 3D elevation data, and Building Information Modeling (BIM) data—exhibit significant differences in format, accuracy, and structure. These differences make data interoperability difficult, complicating the seamless integration of multi-source heterogeneous data.

To address the aforementioned issues, Cesium.js, a widely used open-source 3D visualization framework based on WebGIS, was selected due to its robust capabilities in handling large-scale geospatial data, real-time rendering, and extensive support for 3D Tiles data format. These features make Cesium.js particularly suitable for constructing a

real-scene 3D geographic information platform, as it facilitates efficient data streaming, seamless integration of multi-source data, and interactive visualization, crucial for creating accurate and detailed 3D models in web applications. However, integrating various data layers into such a platform still presents several challenges, as illustrated in Figure 1.

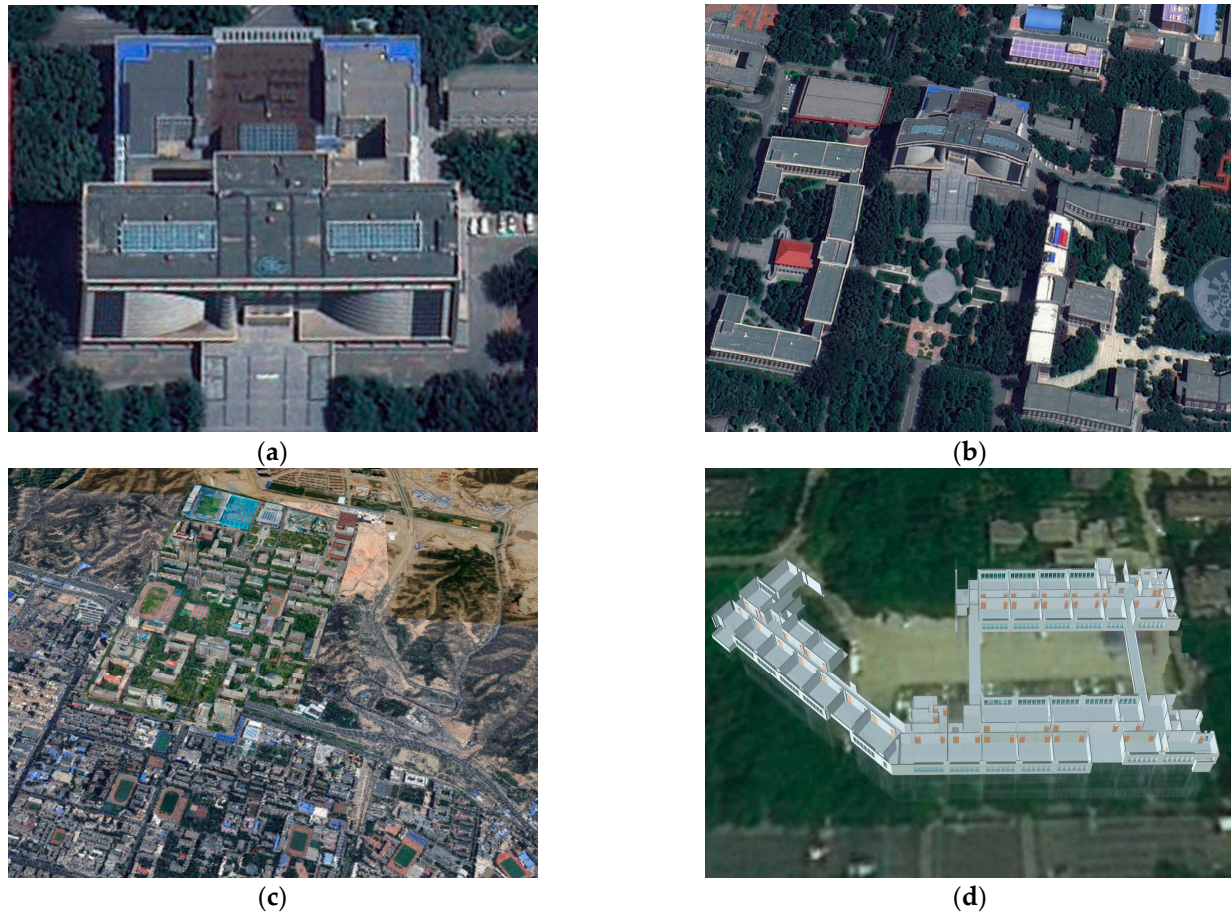


Figure 1. Challenges in Integration of Different Data Layers for 3D Digital Campus: (a) Satellite Imagery Alone; (b) Satellite Imagery Combined with Digital Surface Model (DSM); (c) Satellite Imagery Combined with Oblique Photography; (d) Oblique Photography Data Alone.

In Scenario (a), satellite imagery offers broad coverage but lacks elevation data, which results in insufficient clarity and detail, particularly at higher zoom levels. In Scenario (b), satellite imagery is integrated with a Digital Surface Model (DSM) to provide elevation data. However, the DSM may distort certain building features due to inaccurate protrusions. Scenario (c) demonstrates the combination of satellite imagery with oblique photography, which improves spatial resolution but introduces misalignment due to the lack of elevation data in satellite imagery. Ultimately, scenario (d) depicts oblique photography data in isolation. While this approach enhances resolution and accuracy, it is unable to clearly depict internal building features.

This paper proposes a systematic multi-source data fusion approach to construct a real-scene 3D digital campus by unmanned aerial vehicle (UAV) oblique photography, digital elevation model (DEM) or DSM, OpenStreetMap (OSM) building data, building information modeling (BIM), satellite imagery, and other data sources. The feasibility of this approach is demonstrated through the development of a real-scene 3D digital campus for Lanzhou Jiaotong University. This university represents the typical characteristics of China's higher education institutions, with a medium-sized campus, a balanced array of architectural structures, diverse geographical features, and a representative number of enrolled students.

As a representative example of a broader category of ordinary Chinese universities, it provides a suitable and practical case study for evaluating the proposed multi-source data fusion approach. The process begins with high-resolution image collection using DJI UAVs, followed by processing in open-source software to generate the initial real-scene 3D campus model. Two-dimensional (2D) maps provide basic geographic information and spatial references, while OSM building data and 3D terrain data represent macroscopic building structures and terrain features. The oblique photography model serves as the foundation for detailed 3D construction, which is further refined using BIM models, point cloud data, and OBJ models. This approach enables multi-level data integration, from overall layout to fine details, ensuring the creation of a comprehensive and accurate real-scene 3D digital campus model. As part of the proposed approach, two important methods are employed to achieve effective data integration: the Coordinate Transformation and Position Calibration algorithms, which transform local coordinates of 3D Tiles models (such as those generated by the BIM model) and perform position calibration to address initial positional discrepancies; and the Layered Geospatial Interaction Retrieval Algorithm (LGIRA), which establishes layer priorities for correct visualization order and facilitates the extraction of precise geospatial information from user interactions within the multi-layered 3D environment powered by the Cesium framework. The contributions of this study are as follows:

- (1) **A Novel Multi-Source Data Fusion Method:** We develop an approach that seamlessly integrates heterogeneous geospatial data—including UAV oblique photography, DEM/DSM, BIM, and OSM data—using the 3D Tiles format. This method enhances data consistency and accuracy, effectively resolving compatibility issues among different data formats.
- (2) **Advancement in Web-Based Real-Scene 3D Digital Campus Visualization:** By utilizing Cesium.js, we implement an interactive web application that renders high-precision 3D campus models with features like dynamic loading and perspective switching. This approach offers advantages over traditional visualization methods by offering real-time interaction without the need for specialized client software, thus increasing accessibility and user engagement.
- (3) **Enhanced Decision Support for Campus Management:** The developed platform serves as a practical tool for administrators by providing an intuitive and comprehensive view of campus infrastructure. It supports informed decision-making in facility monitoring, impact assessment of new construction projects, and rapid emergency response, thereby potentially improving management efficiency.

The remainder of this paper is organized as follows: Section 2 reviews related work, summarizing the current state of 3D modeling and visualization technologies, multi-source data fusion technologies, and the application of Cesium in web-based 3D visualization. Section 3 outlines the acquisition and preprocessing methods for DJI UAV oblique photography data, DEM/DSM data, and BIM data. Section 4 introduces the overall process of generating a high-precision real-scene 3D Digital Campus, detailing the core algorithms for multi-source data fusion along with their theoretical derivations. Section 5 presents the developed real-scene 3D Digital Campus within our case study area, providing analysis and discussion of its main functions. Finally, Section 6 summarizes the research findings and suggests directions for future work.

2. Related Work

Three-dimensional modeling and visualization technologies have seen significant advancements across various fields, including architectural design, urban planning, and environmental monitoring. Traditional 3D modeling techniques, such as manual modeling

and conventional photogrammetry, have been widely used but are increasingly being replaced due to their high time costs and operational complexities [8]. The integration of technologies like UAV oblique photography, point cloud data processing, and BIM has greatly enhanced the accuracy and efficiency of 3D modeling [9,10].

UAV oblique photography has emerged as a prominent method for capturing high-resolution imagery from multiple angles, forming a robust foundation for large-scale real-scene 3D models [11]. Studies have demonstrated that UAV oblique photography offers significant advantages in operational flexibility, data acquisition efficiency, and coverage, especially in complex terrains or densely built environments [12–14]. For example, Ref. [15] utilized UAV oblique imagery to generate detailed 3D models of urban areas, achieving higher accuracy compared to traditional aerial photogrammetry. Research on image stitching and orthorectification for UAV imagery has further improved the accuracy and reliability of 3D models, making UAV technology an essential component of modern 3D modeling systems [16]. However, UAV oblique photography often encounters challenges in areas with dense high-rise buildings, where occlusion and excessive camera tilt lead to geometric defects, such as shape inaccuracies, holes, object merging, and texture blurring [1]. Ref. [9] integrates UAV oblique photography with point cloud data to address these limitations, enhancing the accuracy and completeness of 3D models, and providing improved geometric and texture detail, particularly in complex urban environments.

When combined with BIM technology, point cloud data enable precise capture of building structures, increasing the detail and accuracy of 3D models [17]. BIM models provide not only geometric information but also attribute data, supporting high-precision modeling tasks, particularly in building and infrastructure applications [18,19]. Furthermore, the Scan-to-BIM method employs laser scanning 3D technology to capture detailed geometric and attribute information of buildings, providing high accuracy and supporting comprehensive facility management [20]. The integration of UAV oblique photography, point cloud data, and BIM model has thus become a focal point in contemporary 3D construction research, facilitating detailed modeling of complex scenes.

A major challenge in 3D modeling is the effective fusion of multi-source data to ensure accuracy and consistency [21,22]. Integrating different types of data sources—such as UAV oblique photography, point cloud data, BIM model, and OpenStreetMap (OSM) building data—requires sophisticated techniques for data preprocessing, spatial registration, and coordinate transformation [23,24]. For instance, oblique photography imagery provides surface texture information, while point cloud data enhances the geometric structure of buildings, leading to more accurate and realistic models [1]. Studies have demonstrated the success of multi-source data integration in applications like urban planning, building analysis, and complex terrain modeling [25]. However, challenges remain in standardizing data formats and achieving seamless interoperability among heterogeneous datasets.

To facilitate the fusion and visualization of multi-source data, the 3D Tiles format has been introduced as a standardized solution for managing large-scale 3D datasets. The 3D Tiles specification provides an efficient structure for storing and rendering massive amounts of 3D data through tile-based storage and hierarchical level-of-detail (LOD) rendering [26,27]. Recent studies have demonstrated the conversion of BIM models and point cloud data into the 3D Tiles format [28,29], and engineering approaches have been developed for integrating UAV oblique photography imagery [30]. The widespread adoption of 3D Tiles across various fields—including architectural design and smart cities—underscores its effectiveness in handling diverse data types. Moreover, with the increasing application of 3D Tiles in web-based platforms, research has focused on improving the efficient conversion of different data types into 3D Tiles for real-time visualization [22].

Cesium.js serves as a primary platform for rendering and interacting with 3D Tiles, playing a critical role in the real-time visualization of large-scale 3D models [31,32]. It supports real-time data loading, efficient rendering, and interactive functionalities such as viewpoint switching and spatial data querying, making it well-suited for applications involving massive 3D datasets. The deep integration of Cesium.js with the 3D Tiles format enables the efficient display of multi-source data, providing robust scalability and compatibility in 3D modeling applications for smart cities and other complex environments [33]. Studies have shown that real-time dynamic loading, interaction, and querying of 3D models on the web through the Cesium.js platform significantly improve the user experience and functionality of 3D data visualization [34,35]. The growing popularity of Cesium.js positions it as a key tool in the future development of 3D data visualization [36–38].

Progress in the construction of a 3D Digital Campus has been made both domestically and internationally, with several explorations demonstrating the potential of multi-source data integration and advanced visualization techniques. Ref. [39] developed a smart campus system that integrates BIM and GIS to create a 3D visualization campus information system for dynamic monitoring of geographic and spatial entities, though the visualization capabilities are limited by the use of PC-based client software. Ref. [40] implemented a BIM-based digital twin platform at the Universidad Politécnica de Madrid, linking BIM models with IoT and cloud computing for infrastructure management, but the focus is primarily on individual campus facilities rather than comprehensive campus-wide 3D modeling. Ref. [41] proposed a 3D virtual campus navigation system based on VR and GIS technologies, although the need for model compression limited its scalability and realism. These examples demonstrate technical advancements in 3D digital campus modeling and management, but they also reveal limitations in integrating large-scale geographic and building data. Additionally, there is still a lack of comprehensive solutions that fully integrate these multi-source datasets into a unified platform. Most studies focus on combining two data types, such as UAV oblique photography with point clouds [1,9], BIM with point clouds [17], or UAV oblique photography with BIM [19], or BIM and GIS [39,40], but few address integrating all these data types simultaneously. Challenges in data interoperability, format standardization, and efficient rendering remain unresolved. Our research addresses these gaps by proposing a comprehensive multi-source data fusion method, integrating UAV oblique photography, DEM/DSM, BIM, and other data using the 3D Tiles format. Using Cesium.js for web-based visualization, we develop a real-scene 3D digital campus that improves data interoperability and real-time rendering.

3. Study Area and Data Acquisition

3.1. Study Area

Lanzhou Jiaotong University is located in Lanzhou City, Gansu Province, in Northwest China. The main campus, which was selected as the study area for this research, has a perimeter of approximately 3563 m and covers an area of 712,312 square meters, as shown in Figure 2. It features over 119 buildings with different heights and architectural styles, including residential halls, administrative offices, academic buildings, stadiums, green spaces, and an intricate network of roads. Specifically, there are 49 buildings below 30 m in height and 26 buildings exceeding 50 m, with some buildings reaching nearly 100 m. The topography is high in the north and low in the south, with an elevation difference of 13 m. The overall rectangular layout, coupled with the variety of building heights and densities, poses significant challenges for 3D modeling. The complex architectural features and dense distribution of structures necessitate the use of multi-source geospatial data to achieve a detailed and accurate representation of the built environment within a limited timeframe.

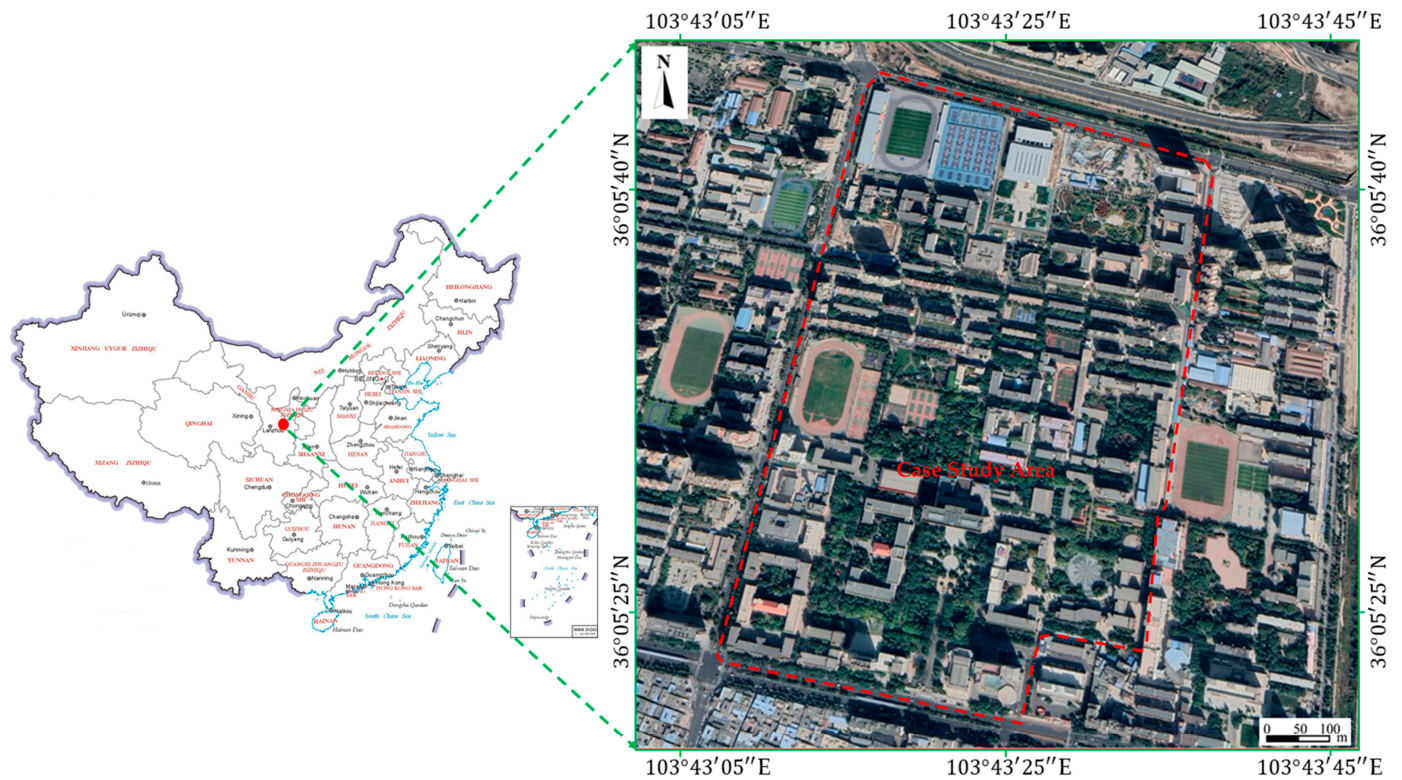


Figure 2. Case study area: Lanzhou Jiaotong University main campus in Lanzhou City (Sources: Google Earth).

3.2. Data Sources

To create a comprehensive and accurate real-scene 3D digital campus model, various datasets from multiple sources were integrated. These datasets differ in data format, spatial reference systems, and content, requiring careful processing to ensure compatibility and seamless integration. Table 1 summarizes the datasets used in this study.

Table 1. Overview of Multi-Source Heterogeneous Data.

Data Type	SRS	File Format	Data Source	Role in 3D Model
UAV Oblique Photography	ECEF	OSGB, OBJ, 3D Tiles	DJI Mavic Air 2 Pro Acquisition	Provides high-resolution 3D models of buildings and the environment, critical for detailed 3D reconstruction and texture-rich visualization.
OSM Building Data	WGS 84 (EPSG:4326)	GeoJSON	OSM Website (https://osmbuildings.org/data/ (accessed on 20 May 2024))	Adds basic geometric and attribute information for buildings not fully captured by UAV photogrammetry, filling data gaps.
DEM/DSM	WGS 84 (EPSG:4326)	Tiff	USGS, JAXA	Provides terrain elevation data and terrain surface data for accurate terrain modeling in the 3D GIS environment.
Point Cloud Data	WGS 84 (EPSG:4326)	LAS, LAZ	DJI Mavic Air 2 Pro Acquisition	Enhances the geometric accuracy of the 3D model, representing spatial structures in great detail.
BIM Model	Local Coordinate System	IFC, OBJ, glTF	Revit (https://www.bimzyw.com (accessed on 25 May 2024))	Delivers detailed architectural models of campus buildings, including geometric and attribute information.
Satellite Imagery	EPSG:3857 (Web Mercator)	TMS, WMTS, WMS	Google Maps online, TianDiTu online	Provides satellite imagery and base maps for geographic context and real-world visualization within the 3D GIS environment.

3.3. Data Acquisition and Processing

3.3.1. UAV Oblique Photography and Route Planning

The primary data source for this study was high-resolution imagery obtained through UAV oblique photography using the DJI Mavic Air 2 Pro. This UAV is equipped with a Hasselblad L1D-20c camera featuring a 1-inch 20-megapixel CMOS sensor, capable of capturing high-quality images essential for detailed 3D modeling. The UAV offers a maximum flight time of up to 31 min and a top speed of 72 km/h, allowing efficient coverage of large areas. Integrated GNSS modules ensure precise geolocation of captured images, which is critical for accurate spatial alignment in 3D reconstruction.

To achieve comprehensive coverage of the study area, precise flight route planning was carried out, as shown in Figure 3a. The campus was divided into 9 distinct sub-region blocks to facilitate effective data collection. Considering the varying building heights—49 structures below 30 m and 26 exceeding 50 m—two sets of oblique photography flights were performed at altitudes of 110 m and 120 m. Each flight plan employs a linear flight mission, illustrated in Figure 3b, consisting of one vertical flight path and four oblique paths from different directions, as depicted in Figure 3d. The overlap of the linear flight mission was set at 70% front-lap and 70% side-lap, with a single set of back-and-forth flight lines. The camera was oriented straight down for nadir photos, with a downward tilt angle of -45° for oblique photography images. This configuration captured both nadir and oblique images to minimize blind spots and ensure detailed facade information. Circular flight paths were also implemented around taller and more complex structures to obtain 360° imagery, enhancing the capture of intricate architectural details as shown in Figure 3c.

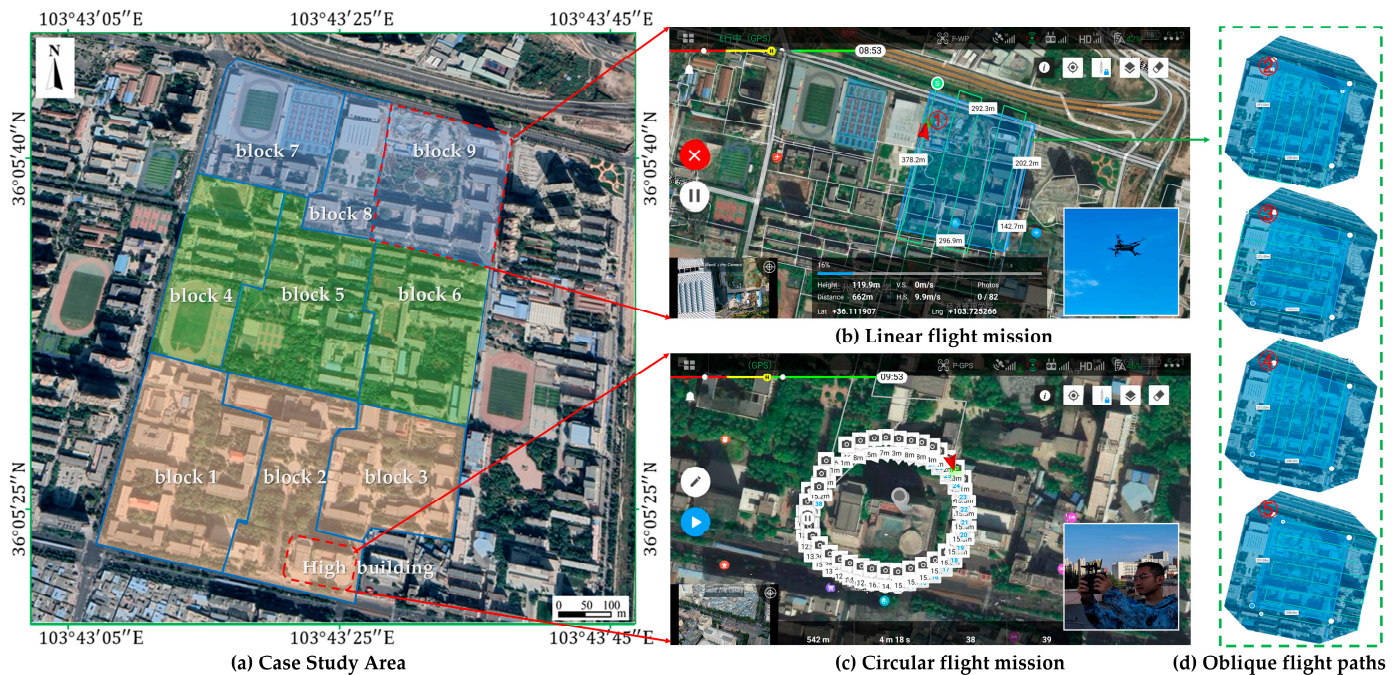


Figure 3. Route planning and design for oblique photography data acquisition.

The UAV data collection resulted in a total of 5117 high-resolution images, amounting to approximately 40.4 GB of data. The images were processed using open-source software WebODM version 2.5.7 to generate initial 3D models in OSGB and OBJ formats. The spatial reference system was initially WGS 84 (World Geodetic System 1984, EPSG:4326) and was later transformed to Earth-Centered Earth-Fixed (ECEF) coordinates to ensure compatibility with the 3D visualization environment. The processed models were then converted into the 3D Tiles format for efficient web-based visualization using the Cesium platform.

3.3.2. Supplementary Data Sources

In addition to the UAV oblique photography imagery, several supplementary datasets (see Table 1) were integrated to enhance the completeness and accuracy of the 3D model, as follows:

- (1) **OSM Building Data:** OSM building data were incorporated to fill gaps not captured by UAV Oblique photography. Extracted using the OSM Web API and downloaded in GeoJSON format, this dataset provided basic geometric shapes and attribute information such as building footprints, heights, and types. Referenced in WGS 84 (EPSG:4326), the OSM data complemented the UAV data by adding missing structures and enhancing the overall building dataset.
- (2) **DEM/DSM:** The 3D terrain tiles were developed using both DEM and DSM to meet diverse user requirements for ground and surface representation. The DEM, obtained from the United States Geological Survey (USGS) SRTM-1 data product at a spatial resolution of 30 m, provided essential baseline elevation data for depicting ground surface variations. In addition, the DSM, sourced from the Japan Aerospace Exploration Agency's (JAXA) ALOS dataset with a finer resolution of 12.5 m, enabled detailed modeling of surface features such as buildings and vegetation. Both DEM and DSM datasets were provided in GeoTIFF format, originally referenced in WGS 84 (EPSG:4326), and were subsequently transformed to Earth-Centered, Earth-Fixed (ECEF) coordinates to ensure precise alignment and seamless integration within the Cesium.js framework.
- (3) **Point Cloud Data:** LiDAR-derived point cloud data contributed to the model's geometric precision. Due to budget constraints, additional point cloud data were collected using a DJI Mavic Air 2 Pro, performing circular flights at various altitudes to capture precise oblique imagery. This imagery was processed to generate dense 3D point representations in LAS format, supplementing ground-based LiDAR data. This imagery was processed to generate a dense 3D point model in LAS format. To enhance accuracy, the oblique photography model was aligned and fused with the LiDAR-derived point cloud data using an Iterative Closest Point (ICP) algorithm, which improved the precision of the oblique photography model. This integration allowed for enhanced detail in building facades and complex architectural elements, resulting in a more accurate representation of the campus's physical structures.
- (4) **BIM Data:** BIM data provided detailed architectural insights not captured by aerial imagery alone. Created using software such as Revit, Bentley, and Civil3D, the BIM data included comprehensive structural and attribute details of campus buildings, including internal layouts and material properties. Initially in Industry Foundation Classes (IFC) format, these models were converted to OBJ and glTF formats for compatibility with other datasets. The BIM data were georeferenced and transformed to align with the spatial reference system used in the 3D model, enriching the representation by adding a higher level of architectural detail.
- (5) **Satellite Imagery:** Satellite imagery from Google Maps or TianDiTu provided contextual background for the 3D model. The satellite imagery was retrieved using web mapping services, including Tile Map Service (TMS), Web Map Tile Service (WMTS), and Web Map Service (WMS). These services employed the EPSG:3857 (Web Mercator) projection, which offers a real-scene representation of the geographic context. The imagery enhanced visualization and aided in orientation within the 3D GIS environment by serving as base maps upon which the 3D models were rendered, thereby providing a familiar visual reference.

4. Research Approaches

This section details the construction process of the proposed real-scene 3D digital campus based on multi-source data fusion. Figure 4 presents the overall workflow from data acquisition to platform integration, covering several key steps including oblique photography, BIM data transformation, OSM building integration, 3D terrain tiles processing, and final 3D environmental setup using the Cesium framework. The following subsection describes the methods used for each component of the proposed approach.

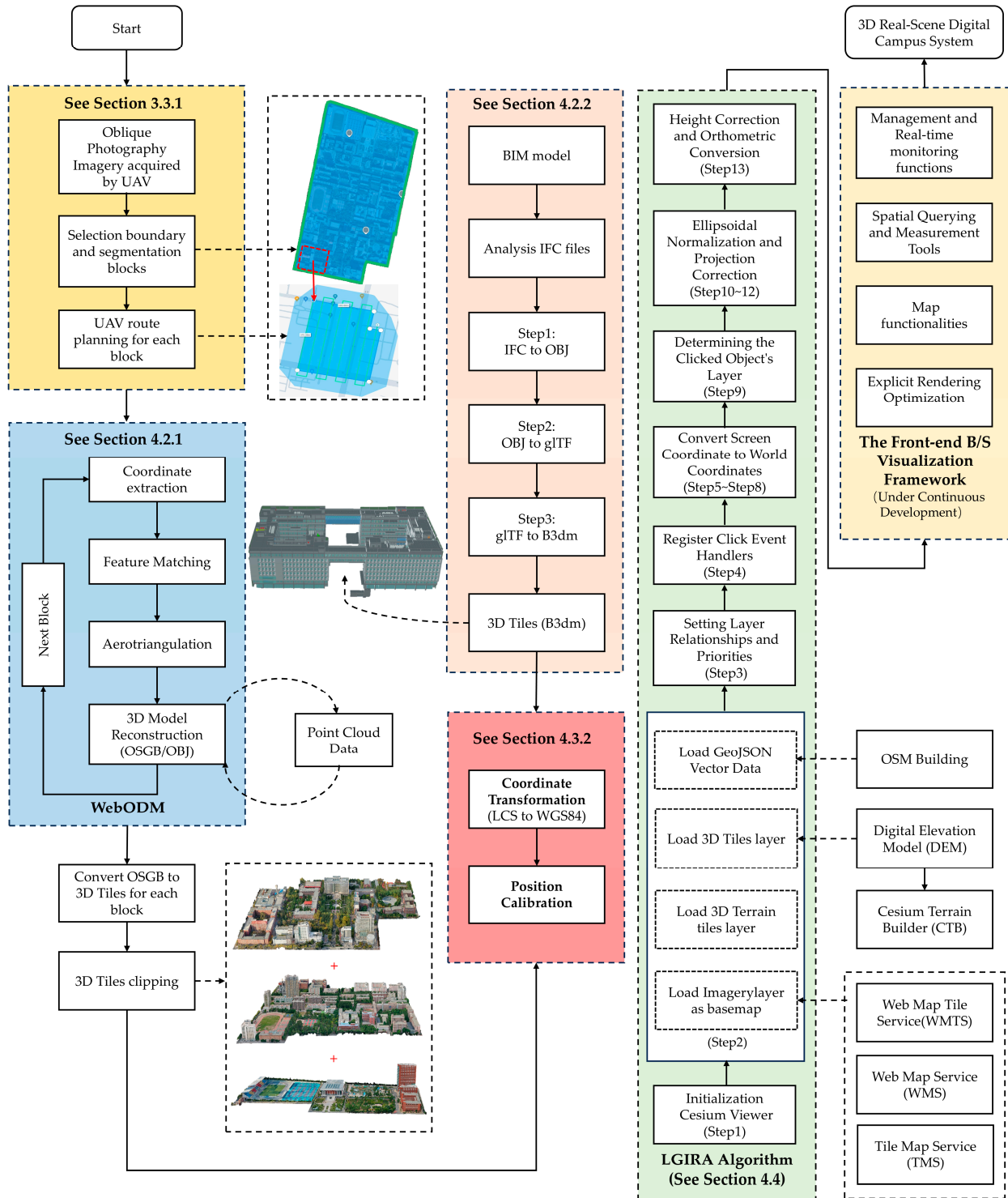


Figure 4. Overall workflow of the proposed approach (A variety of open-source tools and libraries were used in this workflow; see Appendix A).

4.1. Overview of Proposed Approach

Figure 4 presents the comprehensive workflow for constructing a real-scene 3D digital campus, detailing the process of multi-source data integration. This approach combines UAV-acquired oblique photography, BIM models, and other GIS data to generate a highly accurate 3D representation of the study area, which serves as a basis for digital campus management.

The process begins with the acquisition of oblique photography imagery via UAV, followed by large-scale 3D modeling using the open-source software WebODM version 2.5.7, which outputs models in OBJ or OSGB formats. To address occlusions and areas with insufficient detail, point cloud data are utilized for data fusion or manual refinement, resulting in the generation of refined 3D Tile files. The resulting files, initially in the Earth-Centered, Earth-Fixed (ECEF) coordinate system, were subsequently loaded into the Cesium 3D geospatial platform. The BIM model, containing detailed architectural geometry, facility information, and semantic data, is incorporated to support multi-scale digital management applications, including campus layout planning, construction safety, and operation control. To integrate BIM data seamlessly, models are converted from IFC format to 3D Tiles (B3dm), with positional calibration transforming the BIM Local Coordinate System (LCS) into the WGS 84 (EPSG:4326), ensuring compatibility within the Cesium environment. In addition to the above data sources, satellite imagery serves as a base map for the Cesium 3D geospatial platform, combining with 3D terrain tiles to enhance the 3D model's contextual accuracy. The integration of OSM building data and other vector datasets, like points of interest (POI), further enriches the multi-scale real-scene 3D digital campus platform.

The platform developed using the Cesium framework, Node.js, and Vue3, enables dynamic interactions such as querying, perspective switching, location searches, annotations, and measurement tools. Additionally, a Layered Geospatial Interaction Retrieval Algorithm (LGIRA) is proposed to enable users to accurately obtain latitude, longitude, and elevation information for each layer displayed on the screen. This functionality significantly enhances the platform's usability by providing precise spatial data necessary for informed decision-making. Furthermore, a position transformation algorithm and an associated web interface have been developed to allow users to easily adjust model positioning through a web interface, ensuring seamless alignment with the Cesium 3D geospatial platform. The implementation of these features not only enriches the current platform capabilities but also sets the foundation for future expansions, enabling the integration of additional functionalities and data sources.

4.2. Three-Dimensional Tiles Format Conversion and Integration

To enhance the understanding of different data layers loading in the Cesium 3D geospatial platform, a summary of a variety of data layer configurations and their applications has been provided. Table 2 outlines the primary types of data layers, their loading priorities, corresponding Cesium class names, loading methods, and click event retrieval methods, offering detailed insights into the processes of data integration and visualization.

Table 2. Data Layers Configuration in the Cesium Framework.

Layer Type	Priority	Class Name	Loading Method	Click Event Retrieval Method
3D Terrain Tiles	1	<i>Cesium.Viewer</i>	<i>viewer.terrainProvider = new Cesium.CesiumTerrainProvider({...})</i>	Not directly pickable;
Satellite Imagery	2	<i>ImageryProvider</i>	<i>viewer.imageryLayers.addImageryProvider(new Object)</i>	Not directly pickable;
Oblique Photography	3	<i>Cesium3DTileSet</i>	<i>viewer.scene.primitives.add(new Object)</i>	<i>viewer.scene.pick</i>
BIM model	4	<i>Cesium3DTileSet</i>	<i>viewer.scene.primitives.add(new Object)</i>	<i>viewer.scene.pick</i>
OSM Buildings	5	<i>GeoJsonDataSource</i>	<i>viewer.dataSources.add(new Object)</i>	<i>viewer.scene.pick</i>

4.2.1. Conversion of Oblique Photograph Imagery to 3D Tiles

As shown in Figure 4, the conversion of 3D models generated from oblique photography into the 3D Tiles format necessitates several critical steps. The study area was defined in Section 3.1, followed by the planning of flight routes for data collection (See Section 3.3.1). Due to the extensive size of the data, data collection from oblique photography was divided into nine blocks. Considering hardware performance, modeling time, and subsequent optimization requirements, these blocks were combined into three larger regions for modeling: blocks 1 to 3, blocks 4 to 6, and blocks 7 to 9. The data from each region were processed using the open-source software WebODM 2.5.7, resulting in the three 3D models in OBJ or OSGB format of Lanzhou Jiaotong University.

The limitations of oblique photography, particularly regarding occluded areas, were noted during large-scale modeling. Additionally, despite the implementation of circular flight planning for specific buildings, certain structures exhibited inadequate levels of detail and texture precision. To address these challenges, point cloud data were incorporated, significantly improving the model's accuracy by providing supplementary spatial information. Despite this, some 3D modeling software was employed for manual refinements, allowing for precise adjustments and enhancements in the geometric representation of the models. This combination of data sources effectively enhanced the overall geometric precision and visual fidelity of the final 3D models.

To integrate point cloud data with the oblique photography model, it was first necessary to extract key features from both the imagery and the point cloud data. These features included key points, lines, and surface elements such as corners and edges. Feature extraction was performed using tools such as Open3D, CloudCompare, and Point Cloud Library (PCL). The extracted features facilitated the establishment of a matching relationship between the imagery and the point cloud data. Subsequently, the Iterative Closest Point (ICP) algorithm was employed to accurately align the point clouds generated from oblique photography imagery captured at different altitudes. This process ensured that the point cloud data and the 3D oblique photography model were merged within a uniform coordinate system, thereby enhancing the geometric precision of the model. For manual refinement, 3D modeling software such as DP-Modeler V2.3, and ModelFun V4.1 were used to improve the detail and accuracy of the resulting models. The final output was in OBJ or OSGB format, which was then converted into the 3D Tiles format and subsequently loaded into the Cesium framework, as outlined in Table 2.

4.2.2. BIM Data to 3D Tiles

BIM plays a crucial role in the fields of architecture, engineering, and construction, whereas GIS provide valuable context for geospatial applications. GIS serves as a tool to

describe large-scale macro environments, whereas 3D mesh models generated through UAV oblique photography provide high-precision (centimeter-level) representations of the external aspects of these environments. In contrast, BIM focuses on the intricate details of the internal structures of buildings. The integration of these technologies promotes interdisciplinary collaboration and supports the development of a digital campus, as indicated in Table 2, which details the methods for loading BIM data into the Cesium framework.

The Industry Foundation Classes (IFC) standard is a set of data standards specifically designed for the construction industry, with the purpose of describing building elements. Notably, common BIM software, such as Autodesk Revit, AECOSim Building Designer, and TEKLA, supports both importing and exporting of IFC formats. At present, there is no direct method for utilizing BIM data within a Cesium 3D geospatial platform, and research on techniques for converting IFC files directly into 3D Tiles is limited. Following the methodologies outlined in [27,42], the conversion of BIM data into 3D Tiles was divided into three steps: IFC to OBJ, OBJ to glTF, and glTF to B3dm, as shown in Figure 4.

Step1: IFC to OBJ conversion. Initially, common BIM software can export models in the IFC format. To convert the IFC files to OBJ format, the open-source tool *IfcOpenShell* is employed. The resulting OBJ files contain the necessary geometric information for 3D models, while a separate JSON file retains semantic attributes.

Step2: OBJ to glTF Conversion. Subsequently, the data are subjected to further processing in order to convert the OBJ files into glTF format. The glTF format is selected for its efficiency in transmitting 3D model data over networks. Conversion is performed using the open-source tool *obj2gltf*. The glTF format employs a JSON file to organize data, including the overall structure and binary files containing information such as vertices and textures.

Step3: glTF to B3dm Conversion. In the final stage, the glTF files are embedded into b3dm tiles, which are a specialized format used for rendering 3D Tiles in platforms like Cesium. B3dm files encapsulate both geometric data and semantic attributes needed for 3D visualization. The open-source tool *3d-tiles-tools* was employed to package the glTF and JSON files into b3dm for efficient rendering in a WebGL environment.

It is important to note that the coordinate system of the current model is specified as LCS (Local Coordinate System). Coordinate calibration was performed to transform the LCS into the WGS 84 (EPSG:4326) system, thereby ensuring perfect compatibility and accurate positioning within the Cesium environment. The algorithm for this conversion is described in detail in Section 4.3.2.

4.2.3. Integration of OSM Building Data and Vector Layers

The integration of OSM building data into a real-scene 3D digital campus platform can be accomplished through two primary methods: loading GeoJSON data and loading 3D Tiles data. The first method involves loading GeoJSON data obtained from OSM directly using the *GeoJsonDataSource()* method, and the vector data are loaded with the same method. This approach facilitates efficient visualization of building data within the Cesium framework, as detailed in Table 2. The second method utilizes the Feature Manipulation Engine (FME) to convert the GeoJSON data into OBJ or glTF formats. Following this conversion, the tools *obj2gltf* and *3d-tiles-tools* are employed to transform the data into the 3D Tiles format. This method uses the same loading method *Cesium3DTileSet()* as those used for oblique photography and 3D BIM models, as outlined in Table 2.

4.2.4. Integration of 3D Terrain Tiles, and Imagery Layers

In the Cesium framework, the loading mechanisms for 3D terrain tiles and imagery layers differ, yet they adhere to similar principles. As illustrated in Table 2, each of

these elements is loaded using different types of Provider classes that manage external geographic data resources and render them within the scene. Both layers employ a tile-based approach, enabling loading data over the network and displaying it as suitable zoom levels while employing a hierarchical structure to dynamically load data based on viewing requirements.

Three-dimensional terrain tiles are loaded using the *CesiumTerrainProvider()* method, which provides elevation data for global or regional areas, thus serving as the foundational layer upon which imagery layers and other 3D models are rendered. This approach ensures accurate terrain visualization and a real-scene representation of the landscape. Imagery layers overlay satellite images or maps onto the terrain or globe's surface. Data sources for these layers may include TMS, WMTS, or WMS protocols, each managed by different *ImageryProvider* classes. For example, *WebMapTileServiceImageryProvider()* method is utilized for loading WMTS, *UrlTemplateImageryProvider()* handles TMS, and *WebMapServiceImageryProvider()* supports WMS.

By employing these different providers, Cesium ensures efficient integration and rendering of diverse geographic data layers, thereby enhancing the visual representation and user experience in a 3D environment.

4.3. Coordinate System and Transformations

In the Cesium framework, transformations between different coordinate systems are essential for querying, analyzing geographic information, and accurately displaying 3D models. This section provides a comprehensive overview of the coordinate systems used in Cesium, emphasizing the importance of WGS84 (EPSG:4326) for global referencing. It also discusses the methods for transforming the local coordinates system into the global coordinate system WGS84 (EPSG:4326), ensuring proper alignment with geographic references.

4.3.1. Coordinate Systems

The primary coordinate systems employed in the Cesium framework include the Screen Coordinate System, Cartesian Coordinate Systems, and Geographic Coordinate Systems. These systems facilitate accurate spatial referencing both on-screen and in the real world.

The Screen Coordinate System operates on a 2D plane where the origin (0,0) is positioned at the top-left corner of the canvas. The X-axis extends horizontally to the right, while the Y-axis extends vertically downward. This system is primarily used for interactions within the Cesium framework, such as identifying points through click events or rendering models on the screen. Cesium employs Cartesian2 objects to manage operations within this 2D screen coordinate system.

The Cartesian Coordinate System, also known as the global Cartesian spatial rectangular coordinate system or world coordinate system, represents positions in 3D space. In the Cesium framework, the origin is situated at the geometric center of the Earth, with the X-axis extending towards the prime meridian (zero degrees longitude), the Y-axis pointing towards 90 degrees east longitude, and the Z-axis aligned with the North Pole. This system allows for the representation of absolute positions on Earth and functions as the standard 3D coordinate framework for global scenes in the Cesium framework.

The Geographic Coordinate System is based on a 3D spherical surface, defining positions on the Earth's surface through latitude and longitude. This system consists of three components: the angular unit of measurement, the prime meridian, and the reference ellipsoid. Lines of equal latitude are represented by horizontal lines, while lines of equal

longitude are depicted as vertical lines. The default geographic coordinate system for the Cesium framework is defined by WGS 84 (EPSG:4326).

4.3.2. Coordinate Transformation and Position Calibration

Three-dimensional models, including formats such as 3D Tiles (Converted from BIM), and OBJ models, typically operate within a local coordinate system. When these models are imported into the Cesium framework, their local coordinates must be transformed into the global coordinate system WGS84 (EPSG:4326) to ensure proper alignment with global geographic references. Additionally, discrepancies in initial positional or orientational characteristics may arise during the conversion process, leading to inconsistencies with Satellite Imagery. Consequently, spatial adjustments are often necessary, commonly achieved through translation and rotation transformations, which are standard techniques in computer mapping.

To facilitate spatial calibration, translation, rotation, and scaling methods are employed. The origin of the local coordinate system is designated as the center of the current model to ensure accurate translation. As shown in Figure 5, the initial center point P_0 of this local coordinate system is represented in the Cesium framework coordinate system as (x_0, y_0, z_0) , which is typically centered on a geometric feature of the model. Given that many 3D models lack intrinsic positioning data, it is common practice to empirically assign a latitude, longitude, and elevation value. This approach allows for continuous calibration, transforming the local coordinates to align with the global coordinate system WGS84 (EPSG:4326) to achieve a center point $P_1(x_1, y_1, z_1)$. This position calibration is essential for ensuring accurate positioning within Satellite Imagery.

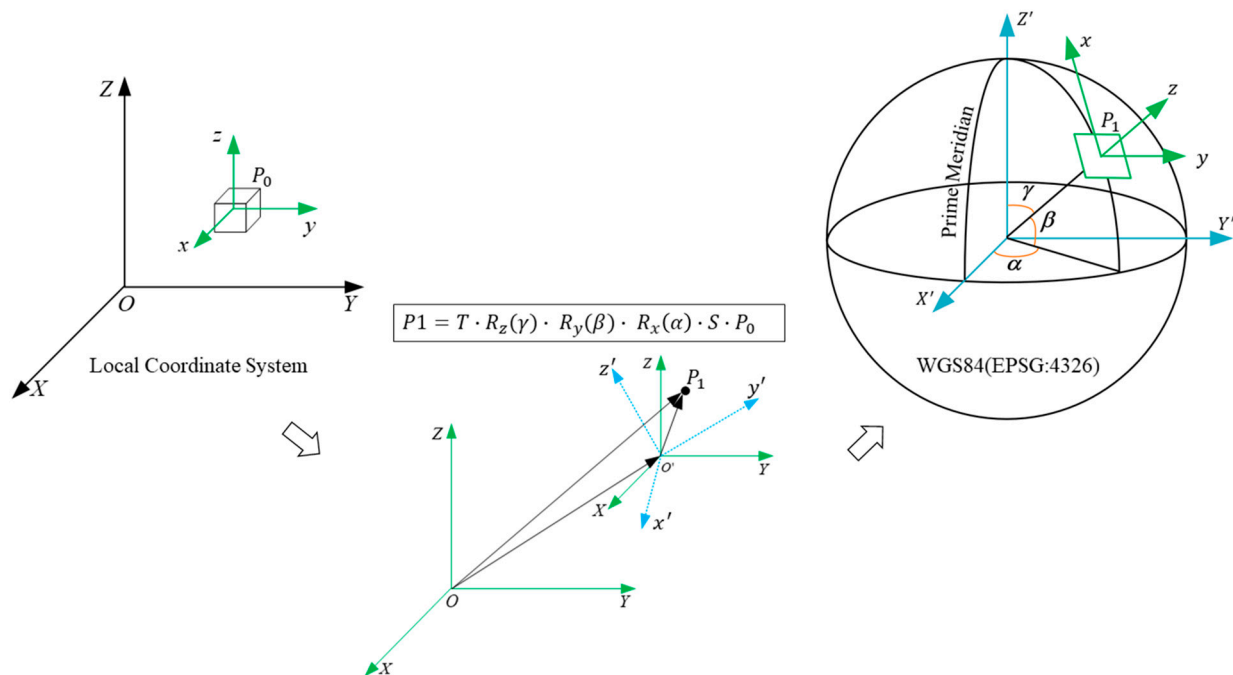


Figure 5. Coordinate transformation.

Assuming the local coordinate system is $O-xyz$, the first step involves translating the origin of the local coordinate system from $O-xyz$ to $O'-xyz$, which effectively shifts it to $(0,0,0)$ in the Cesium framework coordinate system. Following this, a rotational transformation is applied to adjust the orientation of the model, transforming the local coordinate system from $O'-x'y'z'$ to align with the world coordinate system. In this process,

the vector $\vec{OO'}$ represents the translation, while vector \vec{OP} represents the transformed direction. The transformation can be represented as:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = T \cdot R \cdot S \cdot \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \quad (1)$$

where the transformation matrix M is composed of the translation matrix T , the rotation matrix R , and the scaling matrix S :

$$M = T \cdot R \cdot S \quad (2)$$

The translation matrix T can be expressed as:

$$T = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where T_x , T_y , and T_z represent the translation distance along the X-axis, Y-axis, and Z-axis, respectively.

Assuming rotations around the X-axis, Y-axis, and Z-axis by angles α , β , and γ , the rotation matrix R can be expressed as:

$$R = R_z(\gamma) \cdot R_y(\beta) \cdot R_x(\alpha) \quad (4)$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Combining these, the complete rotation matrix R can be derived as follows:

$$R = \begin{bmatrix} \cos(\beta) \cos(\gamma) & -\cos(\beta) \sin(\gamma) & \sin(\beta) & 0 \\ \cos(\alpha) \sin(\gamma) + \sin(\alpha) \sin(\beta) \cos(\gamma) & \cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma) & -\sin(\alpha) \cos(\beta) & 0 \\ \sin(\alpha) \sin(\gamma) + \cos(\alpha) \sin(\beta) \cos(\gamma) & \sin(\alpha) \cos(\gamma) + \cos(\alpha) \sin(\beta) \sin(\gamma) & \cos(\alpha) \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Assuming scaling factors S_x, S_y, S_z , the scaling matrix S can be expressed as:

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

By applying the aforementioned translation, rotation, and scaling transformations, the transformation matrix M is calculated. The matrix can be set using the *Cesium.Matrix4.fromTranslation()* method to achieve an accurate calibration of the model's local coordinate system to the global coordinate system, correct positioning and orientation within the Cesium framework. This comprehensive approach to coordinate transformation and position calibration is crucial for effectively aligning 3D models within the Cesium environment.

4.4. Layered Geospatial Interaction Retrieval Algorithm (LGIRA)

The Layered Geospatial Interaction Retrieval Algorithm (LGIRA) is designed to set layer relationships and priorities, ensuring proper rendering order with high prior layers, before extract precise geospatial information for any data layer—including latitude, longitude, and elevation—from any data layer based on user interactions within a multi-layered 3D environment powered by the Cesium framework. This algorithm facilitates accurate conversion from screen coordinates to geodetic coordinates, accounting for various data layers, such as terrain, satellite imagery, oblique photography models, OSM buildings, and 3D BIM data. The process is executed in several key steps:

Step 1: Initialization Cesium Viewer. The first step involves initializing a Cesium Viewer instance and configuring the necessary parameters to render a real-scene 3D environment. This includes enabling depth picking and any additional options required for effective interaction handling.

Step 2: Loading Data Layers with Assigned Priorities. Data layers are loaded according to their designed loading priority, as outlined in Table 2. To facilitate click event recognition for each layer instance during the loading process, a unique layer identification (e.g., `layer_id == '001'`) is established.

Step 3: Setting Layer Relationships and Priorities. Layer relationships are established, ensuring proper rendering order based on assigned priorities. Higher priority layers (indicated by lower numerical values) are rendered above lower priority layers. This step ensures that the visibility and rendering order of layers is correctly configured.

Step 4: Register Click Event Handlers. An event handler is implemented to capture mouse click events in the Cesium Viewer, enabling the application to respond to user interactions. Specifically, a *Cesium.ScreenSpaceEventHandler()* is created to listen for left-click events on the map canvas. The *setInputAction()* method is used to associate a callback function with the left-click event. When the user clicks, the callback retrieves the object at the clicked position, as shown in Table 2.

Step 5: Conversion from Screen Coordinates to Drawing Buffer Coordinates. Assuming the screen height is h_d , the screen coordinates are (x_s, y_s) in pixels, the first step is to convert the screen coordinates (x_s, y_s) to draw buffer coordinates (x_d, y_d) . In Cesium, the screen coordinates are relative to the top-left corner of the screen, while the drawing buffer coordinates are relative to the bottom-left corner. This conversion can be expressed as:

$$x_d = x_s, y_d = h_d - y_s \quad (10)$$

Step 6: Conversion to Normalized Device Coordinates (NDC). The screen coordinates are converted into NDC, transforming the 2D screen space into a normalized 3D coordinate system ranging from $[-1, 1]$. Cesium performs this transformation using its API, typically

involving functions like *getPickDepth()* to calculate the depth and then converting it into NDC (x_n, y_n, z_n) . The transformation matrix used is:

$$\begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} = \begin{bmatrix} 2/w_d & 0 & 0 & -1 \\ 0 & 2/h_d & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ z_d \\ 1 \end{bmatrix} \quad (11)$$

where w_d and h_d are the screen width and height, respectively.

Step 7: Inverse Projection Matrix to Convert NDC to View Coordinates. In this step, the inverse projection matrix P^{-1} is used to transform the NDC $(x_n, y_n, z_n, 1)$ are transformed into view coordinates (x_v, y_v, z_v, w_v) . The equation used is:

$$\begin{bmatrix} x_v \\ y_v \\ z_v \\ w_v \end{bmatrix} = P^{-1} \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} \quad (12)$$

The inverse projection matrix P^{-1} is defined as:

$$P^{-1} = \begin{bmatrix} \frac{r+l}{2n} & 0 & 0 & \frac{r+l}{2n} \\ 0 & \frac{t-b}{2n} & 0 & \frac{t+b}{2n} \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -\frac{n-f}{2nf} & \frac{n+f}{2nf} \end{bmatrix} \quad (13)$$

where n represents the near plane distance; f is the far plane distance; l, r, t, b are the distances to the left, right, top, and bottom planes, respectively, as shown in Figure 6, where these parameters are labeled for clarity.

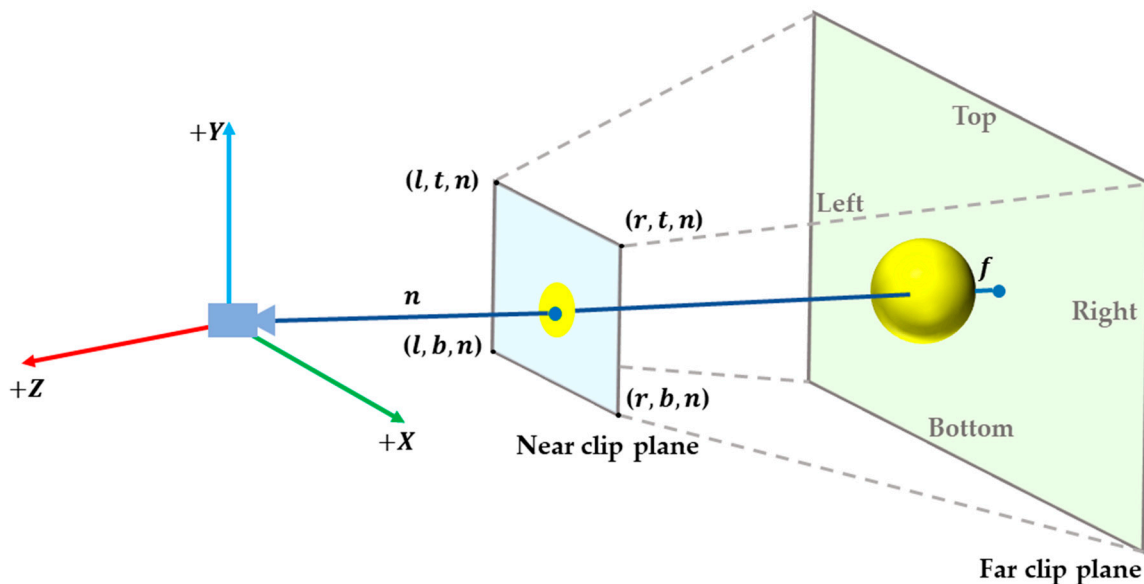


Figure 6. Camera View and Clip Plane Relationship: View Coordinates and NDC.

Step 8: View Matrix to Convert View Coordinates to World Coordinates. The view matrix V is used to transform view coordinates $(x_v/w_v, y_v/w_v, z_v/w_v)$ into world coordinates (x_w, y_w, z_w) using the following equation:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = V \begin{bmatrix} x_v/w_v \\ y_v/w_v \\ z_v/w_v \\ 1 \end{bmatrix} \quad (14)$$

The specific form of the view matrix V is given by:

$$V = \begin{bmatrix} r_x & r_y & r_z & -e \cdot r \\ u_x & u_y & u_z & -e \cdot u \\ d_x & d_y & d_z & -e \cdot d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

where r_x, r_y, r_z represent the right direction vector of the camera, u_x, u_y, u_z represent the up-direction vector of the camera, d_x, d_y, d_z represent the forward direction vector of the camera, and e is the position vector of the camera.

Step 9: Determining the Clicked Object's Layer. A picking operation is performed using the screen coordinates to identify the click object and its corresponding *lay_id*. The retrieval method is determined based on the specific Cesium Class of the layer, as indicated in Table 2.

Step 10: Normalization of World Coordinates. After obtaining the world coordinates, the next step is converting these into the WGS84 (EPSG:4326), including latitude, longitude, and altitude. This requires accounting for the elliptical shape of the Earth, which is modeled by the ellipsoid. The Earth's ellipsoid shape must be accounted for, where the semi-major axis a , b represents the equatorial radius $a = b = 6,378,137$ m, representing the equatorial radius), and the semi-minor axis $c = 6,356,752.314245$ m, represents the polar radius. The normalized world coordinates are calculated as:

$$x_n^2 = \left(\frac{x_w}{a}\right)^2, y_n^2 = \left(\frac{y_w}{b}\right)^2, z_n^2 = \left(\frac{z_w}{c}\right)^2 \quad (16)$$

The normalization factor N is then computed as:

$$N = x_n^2 + y_n^2 + z_n^2 \quad (17)$$

From this, the initial projection point $P_i = (x_i, y_i, z_i)$ is calculated as:

$$P_i = \gamma \cdot (x_w, y_w, z_w) \quad (18)$$

where $\gamma = \sqrt{1/N}$.

Step 11: Iterative correction of the projection point. The initial projection point P_i is iteratively refined using a gradient descent method to ensure that it lies accurately on the ellipsoid surface. The iterative process involves adjusting the parameter λ and the gradient vector g at each step:

$$\lambda = \frac{(1.0 - \gamma) \|c\|}{0.5 \|g\|} \quad (19)$$

$$g = \left(\frac{2x_i}{a^2}, \frac{2y_i}{b^2}, \frac{2z_i}{c^2} \right) \quad (20)$$

In each interaction, λ is gradually adjusted, and the projection point is refined using the following steps:

- (1) Calculate Multipliers x_m, y_m, z_m : These adjust the components of each coordinate. The formulas are:

$$x_m = \frac{1.0}{1.0 + \lambda \cdot \frac{1}{a^2}}, y_m = \frac{1.0}{1.0 + \lambda \cdot \frac{1}{b^2}}, z_m = \frac{1.0}{1.0 + \lambda \cdot \frac{1}{c^2}} \quad (21)$$

- (2) Calculate the Function Value $F(\lambda)$: This function represents the distance between the current projection point and the ellipsoid surface. The formula is:

$$f(\lambda) = x_n^2 x_m^2 + y_n^2 y_m^2 + z_n^2 z_m^2 - 1.0 \quad (22)$$

- (3) Calculate the Derivative $f'(\lambda)$: The derivative is used to determine whether to adjust λ to reduce $f(\lambda)$. The formula is:

$$f'(\lambda) = -2.0 \left(x_n^2 x_m^3 \frac{1}{a^2} + y_n^2 y_m^3 \frac{1}{b^2} + z_n^2 z_m^3 \frac{1}{c^2} \right) \quad (23)$$

- (4) Update λ :

$$\lambda = \lambda - \frac{f(\lambda)}{f'(\lambda)} \quad (24)$$

In each iteration, λ is updated, making the projection point more precise. The interaction stops when the absolute value of $f(\lambda)$ is smaller than the specified tolerance $\varepsilon = 10^{-12}$, and the final corrected projection point P_f is calculated as:

$$P_f = (x_f, y_f, z_f) = (x_m x_w, y_m y_w, z_m z_w) \quad (25)$$

Step 12: Calculating Latitude, Longitude, and Altitude. In this step, Longitude θ is calculated from the projection point's x_f and y_f components, which determine the point's east–west position on the surface of the ellipsoid. The formula is:

$$\theta = \arctan2\left(\frac{y_f}{b^2}, \frac{x_f}{a^2}\right) \quad (26)$$

Latitude ϕ is derived from the projection point's z-component and indicates the point's north–south position on the ellipsoid surface. The formula is:

$$\phi = \arcsin\left(\frac{z_f}{c^2}\right) \quad (27)$$

Altitude η is calculated as the distance between the point and the surface of the ellipsoid along the ellipsoid's normal vector. The formula is:

$$\eta = \sqrt{x_f^2 + y_f^2 + z_f^2} - R \quad (28)$$

where R is the radius of the ellipsoid at the specific latitude, accounting for the ellipsoid's flattening.

Step 13: Correction of Height Information and Conversion to Orthometric Height. To ensure accurate elevation data, the *layer_id* identifies the layer type to retrieve geographic details. Layers like Satellite Imagery, OSM Buildings, and 3D BIM models lack altitude data, but building heights in OSM and 3D BIM are combined with 3D terrain elevations for accurate height retrieval.

After obtaining the WGS84 (EPSG:4326) altitude η , it is converted to orthometric height H by subtracting the geoid height N , where $H = \eta - N$. The geoid height, representing the height above mean sea level, is retrieved via the GeoidEval utility (see Appendix A). This conversion provides true elevation above sea level, essential for precise geospatial applications.

5. Case Study

5.1. Development of the 3D Real-Scene Digital Campus System

This study aims to construct an interactive and comprehensive digital campus platform for Lanzhou Jiaotong University by integrating multi-source data such as oblique photography imagery, BIM model, OSM Building data, 3D terrain tiles, and satellite imagery into a unified system. This integration enhances campus management, planning, and visualization capabilities, achieving a dynamic and precise interactive platform. Developed using Cesium.js and Vue3, as shown in Figure 7, the architectural design of this system emphasizes seamless data integration, dynamic visualization, and a user-friendly interface, promoting the digital transformation of the campus.

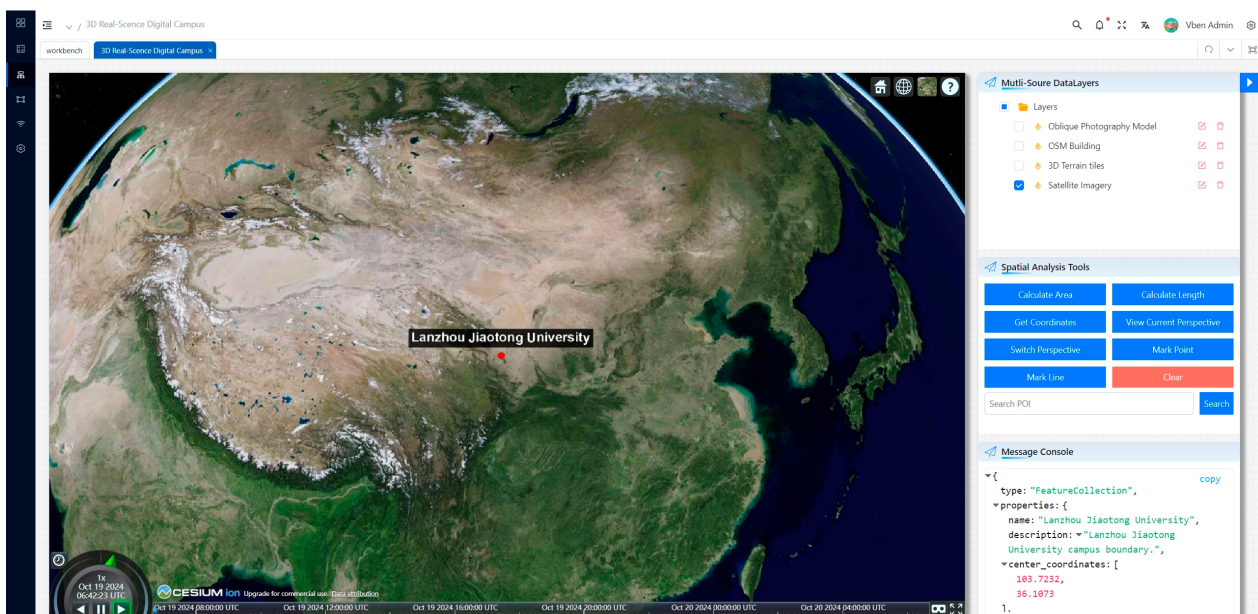


Figure 7. 3D Real-Scene Digital Campus System based on Cesium framework.

The system adopts a front-end and back-end separated B/S software architecture. The front-end utilizes the Vue3 framework to build the user interface and integrates Cesium.js for 3D geospatial visualization. The back-end employs Node.js and the Express framework to handle data requests and API services. Through RESTful APIs, the front-end can dynamically retrieve multi-source data, enabling model loading, rendering, and interaction. Furthermore, the system provides support for cross-platform, multi-dimensional (2D, 2.5D, 3D) digital campus displays, offering a range of map functionalities, including panning, zooming, tilting, switching, resetting, information querying, and building labeling.

5.2. Multi-Source Data Integration Visualization

5.2.1. Oblique Photogrammetry Real-Scene 3D Model

High-resolution oblique photography images were obtained through the use of DJI UAVs. The images were processed using the open-source software WebODM version 2.5.7 to generate three high-precision 3D Tile models, each covering a different area of the campus. To create a comprehensive 3D representation of the campus, the three models were

merged. During this process, feature point matching algorithms were utilized to guarantee seamless integration between the models. Upon loading the models into Cesium, the stitched model was converted to the 3D Tiles format. By meticulously aligning geographic coordinates, the model was accurately loaded into Cesium's geospatial scene, forming a unified 3D Tiles environment, as illustrated in Figure 8.



(a) Region 1 model (from block 1 to block 3)



(b) Region 2 model (from block 4 to block 6)



(c) Region 3 model (from block 7 to block 9)



(d) The entire Lanzhou Jiaotong University model

Figure 8. Stitching of Oblique Photography 3D Tiles Models and Spatial Alignment in Cesium.

By employing high-resolution oblique photogrammetry for 3D real-scene modeling, the system produces a detailed and clear 3D representation, enabling a range of display formats for campus promotion and effectively highlighting prominent architectural features, as illustrated in Figure 9a–d.



(a) The gate of Lanzhou Jiaotong University



(b) Academic library



(c) No.2 teaching building



(d) Tangcheng Square

Figure 9. Oblique Photography 3D Real-Scene Models of Lanzhou Jiaotong University.

5.2.2. Real-Scene 3D Model with Multi-Source Data

In order to enhance the visualization effects of the digital campus, additional data sources were incorporated, including BIM models, OSM building data, 3D terrain tiles, and Satellite imagery. As illustrated in Figure 10, annotations on the integrated digital campus model highlight the detail and accuracy of different data sources, enhancing user interaction and providing contextual information.

During data integration, challenges such as coordinating system discrepancies, file format inconsistencies, and varying resolutions were effectively addressed. Translation and rotation transformations aligned LCS with the WGS84 coordinate system. Data conversion tools unified different formats into the 3D Tiles supported by Cesium. By adjusting resolution and precision parameters, compatibility and consistency among data sources were ensured. The Provider class in Cesium played a critical role in managing the loading, rendering, and caching of external geospatial data resources, enabling efficient visualization and seamless interaction.

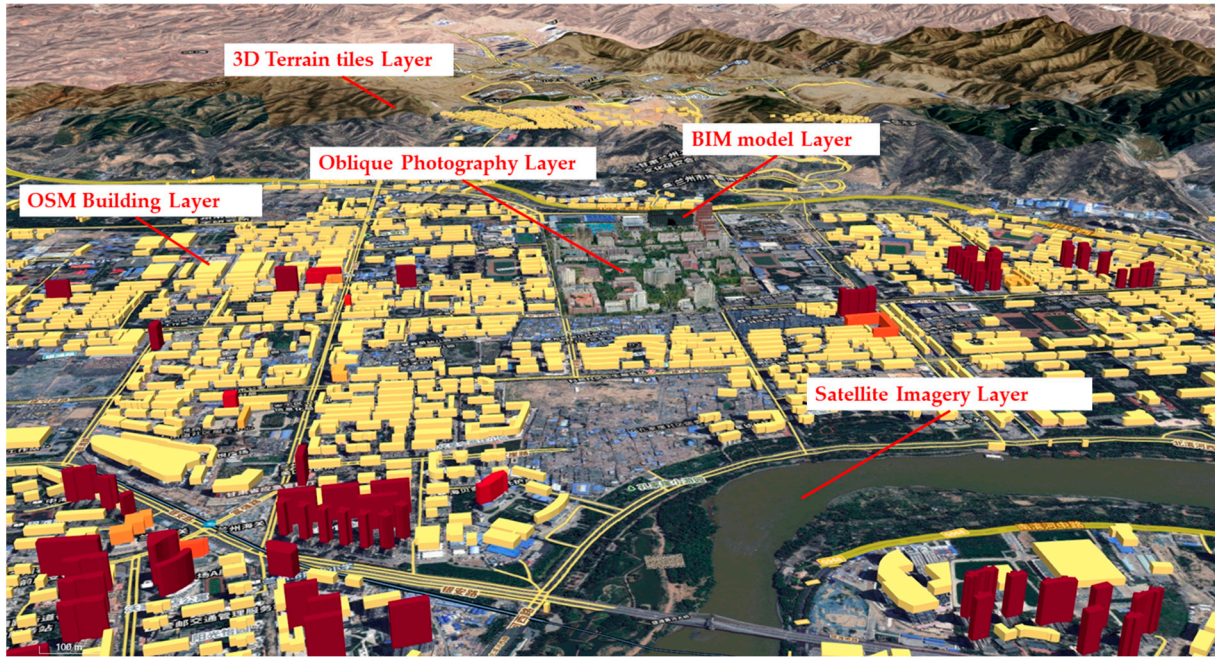


Figure 10. Real-Scene 3D Model with Multi-Source Data Integration.

5.2.3. Implementation of the LGIRA

As shown in Figure 11, the LGIRA (detailed in Section 4.4) was implemented within this system. When a left-click selects any point on the web interface, the LGIRA initiates Step 4, where a click event handler captures the selected point. The system outputs the Clicked Object in the Browser Backend Console Interface, followed by the screen coordinates, which are also displayed in the console. Through Steps 6 to 13 of the LGIRA, the location information is processed and ultimately presented in the web interface, providing precise spatial data that enhances user interaction.

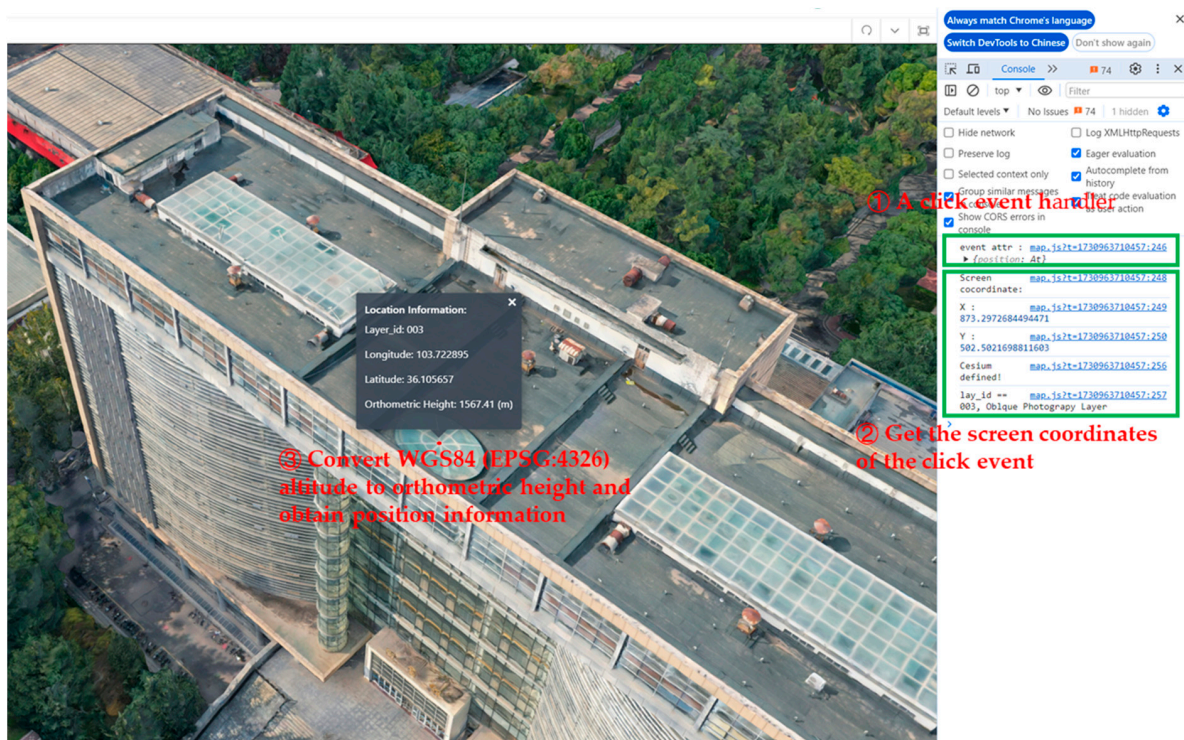


Figure 11. Acquisition of location information based on LGIRA.

5.2.4. Implementation of Coordinate Transformation and Position Calibration

For 3D Tiles, OBJ models, and other local coordinate models, discrepancies in initial positioning or orientation can arise during the transformation process, often leading to misalignment with the 3D terrain tiles layer and satellite imagery layer. As shown in Figure 12, the Coordinate Transformation and Position Calibration algorithm, detailed in Section 4.3.2 was implemented to address this issue. In this figure, the Comprehensive Teaching Building is initially located at Location 1. Using the Model Location toolbar on the right side of the system interface, users can adjust its latitude, longitude, altitude, and rotation angles along the X, Y, and Z-axis. These adjustments allow for precise position calibration, aligning the model accurately to Location 2 on the satellite imagery layer.

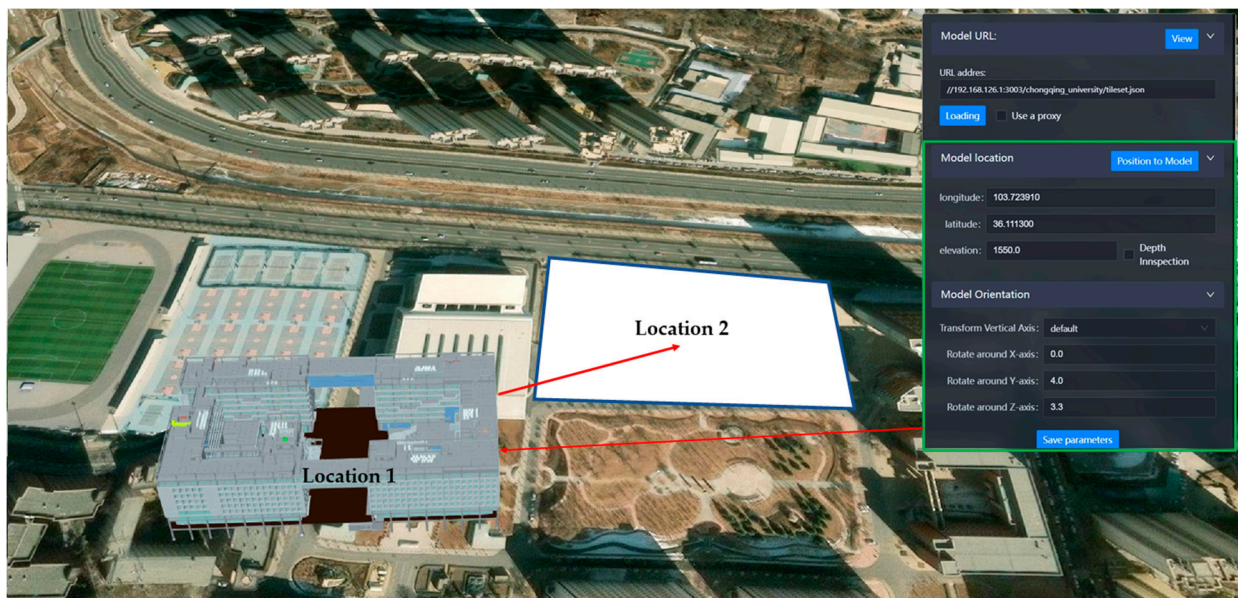


Figure 12. Positional correction of BIM model in 3D Tile format.

5.3. Dynamic Interaction of the 3D Real-Scene Digital Campus System

A significant attribute of the platform is its capacity to facilitate dynamic data visualization and model interaction. To illustrate, the integration of a BIM model will be employed in the construction of a comprehensive teaching building. By transforming the BIM model of the planned building into the 3D Tiles format and loading it into the Cesium framework, it is possible to display the various construction phases of the building in a manner that is both intuitive and comprehensive within the platform.

Figure 13 presents the comprehensive teaching building at different stages of construction. Specifically, Figure 13a displays the current 3D real-scene model of the digital campus generated from oblique photogrammetry. Figure 13b illustrates the foundation construction phase of the building, depicting the layouts of underground structures and infrastructure. Figure 13c shows the building model during the fifth-floor construction stage, highlighting the internal structural framework and floor layouts. Finally, Figure 13d presents the completed building, reflecting the final design.

Through dynamic interaction features, users can switch between construction stages, examining building attributes and structures in detail. Cesium's interactive capabilities facilitate the retrieval of attribute data from the BIM model, including materials, dimensions, and construction progress. This interactivity enhances the user experience and provides robust support for digital campus planning and management.



(a) Current scene



(b) Underground structure



(c) The fifth-floor internal structure

Figure 13. Cont.



(d) Architectural appearance

Figure 13. Dynamic Display of Construction Stages of the Comprehensive Teaching Building.

Figure 14 illustrates the animated weather effects developed for the digital campus platform, including real-time weather and climate change effects, which enhance the realism and user experience, making tours and displays more immersive. For instance, the heavy rain scenario (Figure 14c) and snow and fog conditions (Figure 14d) simulate extreme weather events, allowing for the analysis of climate impacts on campus landscapes and buildings. The integration of meteorological data also supports environmental monitoring and education, helping to inform decisions such as adjusting class schedules based on weather conditions and implementing effective emergency responses during adverse weather. These features not only enrich the visual experience but also expand the platform's utility in educational and research applications.



(a) Spring



(b) Midsummer

Figure 14. Cont.



(c) Heavy Rain



(d) Snow and Fog

Figure 14. Animated Weather Effects in Different Conditions.

6. Result and Discussion

The reconstruction of a large-scale real-scene 3D digital campus using oblique photography-based 3D models remains a relatively underexplored area of research. In this study, the ZED-F9P multi-band Global Navigation Satellite System (GNSS) module, combined with network real-time kinematic (NRTK) high-precision positioning technology, was employed to establish ground control points (GCPs) to improve the accuracy of 3D models generated in WebODM. Furthermore, an evaluation index system was adapted to assess the accuracy of the reconstruction 3D model by error analysis and quantitatively evaluate the large-scale campus 3D reconstruction.

6.1. GCPs and Data Collection

GCPs are essential for correcting distortions within the collected data and anchoring it to a known coordinate system. These points are established based on precise ground-based position measurements, typically obtained using high-precision GNSS devices. The establishment of GCPs relied on NRTK high-precision positioning equipment. In this study, the ZED-F9P GNSS was utilized, which supports multi-frequency GNSS reception, enabling centimeter-level positioning accuracy, rapid convergence times, and reliable performance. When combined with NRTK high-precision positioning service provided by FindCM (<https://www.qxwz.com> (accessed on 18 October 2024)), the carrier phase differential method effectively eliminated errors related to satellite orbits and ionospheric delays, thereby delivering high-precision positioning services. Under optimal conditions, horizontal positioning accuracy achieved up to 2 cm, while vertical accuracy reached up to 5 cm. Figure 15a illustrates the site distribution of the three regional GCPs in the case study area, and GCPs were selected from existing structural features such as manhole covers, road intersections, parking lot lines, and distinct paving tiles, as shown in Figure 15b. Targets were manually placed on the ground to ensure comprehensive coverage and visibility from all camera positions. While ensuring the visibility from all camera positions, the GCPs were distributed as evenly as possible throughout the case study area, covering areas of varying elevation to improve model accuracy.



Figure 15. Location and Feature Selection GCPs for three regions in the Case Study Area.

6.2. System Configuration and Model Parameters

The experiments were conducted on a personal computer configured with the following specifications: Intel Core i9-10900KF CPU (3.70 GHz), NVIDIA GeForce RTX 3080Ti GPU (12GB), 128GB of RAM, and running Windows 10 Education (x64). WebODM is installed on Docker V24.0.6, where Docker is allocated 256GB of disk storage space and 100GB of RAM dedicated to processing inclined photogrammetry data. Utilizing the UAV-based oblique photogrammetry technique depicted in Figure 3, three 3D models were generated, corresponding to Region 1 (blocks 1 to 3), Region 2 (blocks 4 to 6), and Region 3 (blocks 7 to 9). As shown in Figure 16, WebODM provides the Ground Control Point Interface, with the oblique photography image on the left side and the GCPs location information on the right side, which is realized by establishing a link by mouse point and click to associate the oblique photography image map with the GCPs location. WebODM recommends that a minimum of five GCPs be used and that each GCP should be visible in at least five oblique photography images to ensure reconstruction accuracy.

These models were employed to evaluate the performance of the proposed digital campus 3D reconstruction approach. Table 3 presents the fundamental parameters of the three experimental campus regions. It is important to note that the Processing Time does not account for the time spent on WebODM compression and processing of oblique photography images, which is also a very time-consuming process. However, WebODM lacks an effective means of recording this time, meaning the actual total time required to reconstruct the real-scene 3D digital campus regions is greater than the reported Processing Time.

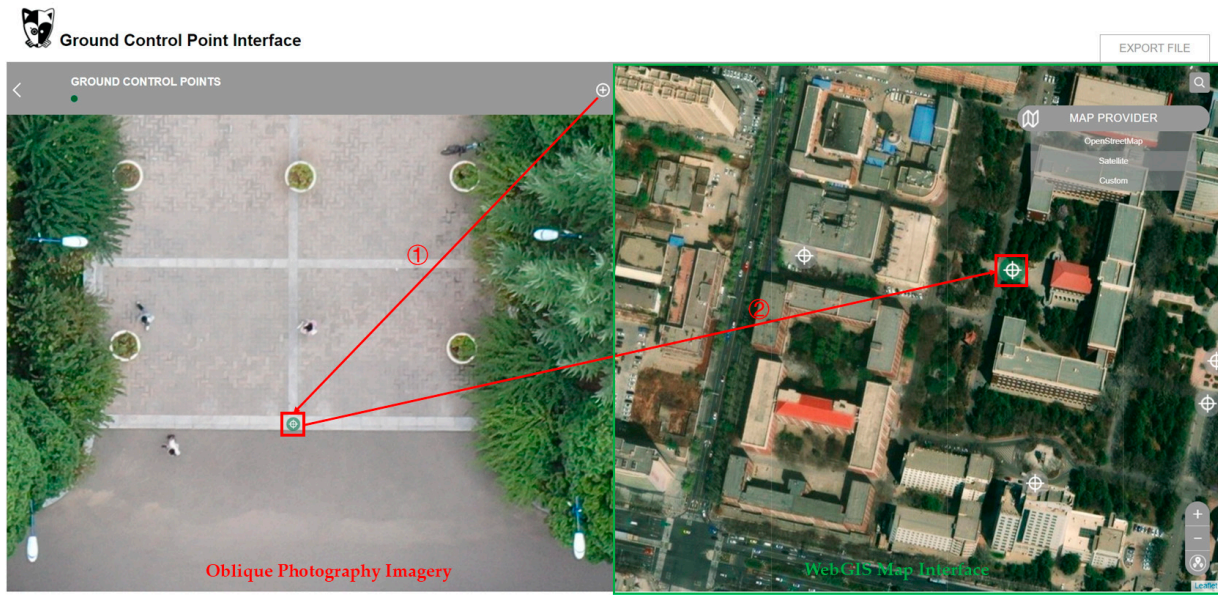


Figure 16. Establishing links between GCPs and positions in Oblique Photography Imagery.

Table 3. Parameters Setting of the Reconstructed 3D models for three Campus Regions in WebODM.

3D Model	Region 1	Region 2	Region 3
Reconstruction Images	1685	1920	1621
GPS-accuracy	0.3 m	0.3 m	0.3 m
pc-quality	high	high	high
Average Ground Sampling Distance (GSD)	5.7 cm	2.6 cm	7.6 cm
Number of GCPs	12	5	8
Reconstructed Points (Dense)	75,806,459	125,380,545	169,082,661
Disk Space Usage	14.05 GB	20.07 GB	13.66 GB
Processing Time	4 h	6 h:26 m	3 h:54 m

6.3. Quantitative Evaluation Results

The reconstruction accuracy of the 3D models was assessed using errors in the X, Y, and Z directions, as well as the overall error. Table 4 presents the error analysis results for the control points in each region. Root Mean Square Error (RMSE), Standard Deviation (SD), and Mean Error were selected as the three metrics for quantitatively analyzing the reconstruction precision. The relevant data were obtained from the ODM Quality Report generated by WebODM.

The experimental results demonstrate that the proposed digital campus 3D reconstruction approach offers significant advantages in terms of reconstruction accuracy. Table 4 shows that the RMSE for GCPs in Region 1, Region 2, and Region 3 were 0.121 m, 0.001 m, and 0.032 m, respectively. The GCP RMSE is a crucial indicator for assessing the alignment between the model and the real-world coordinate system, directly affecting geographic accuracy. This indicates that our modeling process achieves a high level of geographic precision, particularly in applications where accurate geographic coordinates are essential. Additionally, the RMSE for 3D in Region 1, Region 2, and Region 3 were 0.301 m, 0.296 m, and 0.534 m, respectively. This reflects the overall geometric precision of the model, specifically the accuracy of the points in the point cloud or 3D mesh within the three-dimensional space. The 3D error is crucial for ensuring the correctness of the model’s geometry, point cloud density, and detail refinement, highlighting the model’s solid performance in geometric fidelity. The strategic placement of GCPs and the use of low GSD substantially improved the precision of the real-scene 3D digital campus models in this study. By using

consumer-grade UAVs and fully open-source software within the modeling framework, combined with NRTK technology, high-precision 3D modeling was successfully achieved. The results indicate that the proposed approach significantly reduces costs while maintaining comparable levels of accuracy and efficiency, making it an accessible and practical solution for large-scale real-scene 3D digital campus modeling.

Table 4. Residual Statistics of GCPs and the Reconstructed 3D models for Three Campus Regions.

Model Region	GCP	Mean (Meters)	SD (Meters)	RMSE (Meters)	3D	Mean (Meters)	SD (Meters)	RMSE (Meters)
Region 1	X Error	0.008	0.064	0.065	X Error	0.122	0.297	0.321
	Y Error	−0.001	0.170	0.170	Y Error	0.127	0.295	0.321
	Z Error	0.014	0.175	0.175	Z Error	0.206	0.412	0.461
	Total			0.121				0.301
Region 2	X Error	−0.000	0.000	0.001	X Error	0.124	0.538	0.552
	Y Error	−0.000	0.001	0.001	Y Error	0.115	0.331	0.350
	Z Error	0.000	0.001	0.001	Z Error	0.197	0.507	0.544
	Total			0.001	Total			0.296
Region 3	X Error	−0.019	0.052	0.055	X Error	0.235	0.596	0.640
	Y Error	0.007	0.019	0.020	Y Error	0.178	0.421	0.457
	Z Error	0.021	0.058	0.062	Z Error	0.371	1.602	1.644
	Total			0.032				0.534

Despite the success in achieving high precision, data processing efficiency remains significantly influenced by model complexity and hardware configuration. The modeling tasks for the large-scale real-scene 3D digital campus required dividing the scene into three regions for separate modeling, followed by cropping and stitching to achieve the overall model. This indicates a limitation in handling large-scale modeling projects efficiently. Additionally, WebODM does not support GPU acceleration when running on the Windows system, relying primarily on CPU computations. This reliance results in sustained high CPU usage and extended processing times. Another issue is that WebODM does not automatically release Docker’s temporary storage space after model generation, necessitating manual cleanup to free up disk space. Furthermore, the process of adding and verifying GCPs in WebODM is not as efficient or intuitive as in commercial software solutions. These operations tend to be time-consuming, and users cannot verify the accuracy of the GCP placement until the ODM quality report is generated, leading to delays and uncertainty. These limitations highlight areas where WebODM could be improved, particularly in terms of user experience, computational efficiency, and operational efficiency.

6.4. Discussion: Advantages, Limitations, and Future Research

6.4.1. Comparative Analysis and Potential for Further Improvement

The precision metrics achieved in this study demonstrate competitive performance in comparison with other recent large-scale UAV-based 3D modeling studies using commercial 3D modeling software. Specifically, studies [8,43], which employed ContextCapture, and [24], which used Pix4D, all incorporated GCPs and represent some of the most widely used 3D modeling software available on the market today. The RMSE results for GCPs are comparable to these methods, indicating that the precision achieved by the proposed open-source framework approach is on par with some of the most advanced commercial solutions currently available. This suggests that our approach is not only cost-effective but also capable of achieving high accuracy at the same level as our commercial counterparts. However, when compared to the study [30], which employed the DJI Phantom 4 Pro and

Pix4D without GCPs, the RMSE errors are significantly higher, often exceeding one meter. This highlights the critical role of GCPs in enhancing model accuracy, with the results demonstrating a substantial improvement in precision over approaches that lack proper GCPs.

While the proposed approach offers impressive results, there is still potential for further enhancement. For example, study [14] employed DJI Terra along with UAV-based oblique photography and Terrestrial Laser Scanning (TLS) data, achieving centimeter-level precision (5 mm to 8 mm). Although the current proposed approach can effectively fuse point cloud data for registration, TLS data have not yet been incorporated into budget constraints. Future work could integrate TLS data and deep learning techniques, as suggested in [14], to further improve model precision.

The use of TLS would help fill gaps in UAV-based point clouds and deep learning algorithms could enhance the denoising and registration process, leading to more accurate 3D models. The combination of these advanced technologies would significantly improve the quality and precision of real-scene 3D digital campus models, pushing the accuracy from the current decimeter level closer to the centimeter level.

6.4.2. Advantages of the Proposed Approach

The integration of multi-source data significantly enhances the accuracy and interactivity of the real-scene 3D digital campus model, thereby improving campus management, planning, and emergency response capabilities. Incorporating oblique photography imagery, BIM, OSM building data, and 3D terrain tiles resulted in the generation of a highly accurate model that encompasses both the external characteristics of campus buildings and their internal structures. This detailed depiction supports campus planning and resource management by providing precise spatial information essential for facility maintenance, asset management, and emergency response planning. Additionally, the interactive features of the 3D digital campus model, powered by Cesium.js, improve user engagement and interaction with campus spaces. Users can navigate through the virtual environment, perform spatial data queries, and switch perspectives in real time, enhancing the overall user experience for students, faculty, and administrators. This level of interactivity not only aids in better understanding and visualization of the campus layout but also fosters a sense of community by allowing users to engage with the digital representation of their physical environment.

6.4.3. Limitation of the Study

Several limitations and challenges were identified in the proposed approach and platform. Oblique photography modeling demands high-performance hardware, including substantial RAM and GPU configurations, and model optimization often relies on manual processes, which are time-consuming. The implementation of various types of sensors and camera equipment necessary for data collection and real-time monitoring requires considerable manpower and physical resources, posing logistical challenges, especially in large-scale campus environments. Data security remains a critical concern as digital campuses increasingly rely on interconnected systems that handle sensitive information [44]. Ensuring the protection of data against unauthorized access and breaches is paramount to maintaining the integrity and confidentiality of the digital campus model. Achieving a fully functional digital twin campus—where seamless integration exists between the physical and virtual campuses—poses significant challenges. Although a virtual campus has been successfully constructed, real-time data collection and transmission between the physical and virtual environments, powered by AI-driven algorithms and precise control instructions, remain areas for future exploration.

6.4.4. Further Research Directions

Future research will focus on addressing these challenges while enhancing platform functionality. Incorporating UAVs for automated data collection and real-time updates will ensure that the digital campus remains current with minimal manual intervention. Immersive technologies, such as Augmented Reality (AR) and Virtual Reality (VR), will be integrated to expand the platform's user interaction capabilities. VR devices (e.g., Oculus Quest, HTC Vive) and platforms like Cesium for Unreal will enable virtual tours, simulation-based learning, and training applications, while AR technology can overlay digital information onto physical environments to enhance navigation and operational efficiency.

Furthermore, artificial intelligence will be employed to predict construction progress, detect structural anomalies, and optimize resource management. When combined with IoT-enabled sensors, these technologies can facilitate real-time data collection and analysis, providing actionable insights for smarter campus management. Improving rendering performance and processing large-scale models without sacrificing detail or interactivity will remain a primary objective to ensure the scalability and responsiveness of the digital campus platform.

7. Conclusions

This study introduces a novel approach for the construction of a highly accurate three-dimensional digital campus model of Lanzhou Jiaotong University through the fusion of multi-source data. By integrating UAV oblique photography, high-resolution imagery, DEM, BIM, and OSM building data via the 3D Tiles format, compatibility issues were addressed and data accuracy and consistency were improved. This methodology enhances both the precision and realism of 3D models, thereby providing a practical solution for digital campus construction.

The platform encompassed a comprehensive examination of pivotal phases, including data gathering, data cleansing, integration of data from multiple sources, and the deployment of the system. Validation and optimization strategies were employed to guarantee the model's geometric precision and rendering performance. The utilization of Cesium.js facilitated the development of a web-based application with real-time interaction, thereby enhancing accessibility without necessitating the use of specialized software. The platform serves as a valuable tool for campus administrators, offering a comprehensive view of campus infrastructure that supports decision-making in facility monitoring, construction impact assessment, and emergency response, thereby enhancing management efficiency.

In conclusion, the real-scene 3D digital campus represents a significant advancement toward the realization of a fully integrated digital twin campus. Its success depends on ongoing advancements in seamlessly connecting the physical and virtual campuses, addressing data security concerns, and refining the platform's capabilities through practical applications. These advancements will not only enhance the accuracy, interactivity, and functionality of the digital campus but also expand its applications, paving the way for smarter and more dynamic campus management solutions.

The challenges inherent to this field include high data acquisition costs, processing complexity, and performance optimization in web-based applications. Data inconsistencies represent a notable obstacle that requires further research. This study provides valuable insights for similar projects and advances real-scene 3D digital campus construction methodologies. In summary, this study offers an innovative approach for digital campus construction using multi-source data fusion and 3D visualization, contributing to the development of smart campuses, cities, and related fields.

Author Contributions: Conceptualization, Rui Gao and Guanghui Yan; methodology, Rui Gao and Guanghui Yan; software, Rui Gao, Yingzhi Wang and Chunyang Tang; resources, Rui Gao and Ruiting Niu; validation, Rui Gao and Yingzhi Wang; formal analysis, Rui Gao, Tianfeng Yan and Yingzhi Wang; investigation, Rui Gao and Ruiting Niu; data curation, Rui Gao, Tianfeng Yan and Ruiting Niu; writing—original draft preparation, Rui Gao, Yingzhi Wang and Ruiting Niu; writing—review and editing, Guanghui Yan and Tianfeng Yan; visualization, Rui Gao, Yingzhi Wang, Tianfeng Yan and Ruiting Niu; supervision, Guanghui Yan, Yingzhi Wang and Tianfeng Yan; project administration, Rui Gao and Guanghui Yan; funding acquisition, Rui Gao and Guanghui Yan. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China under Grand No. 62161017, No. 62366028, No. 62361034, and No. 62466032, Lanzhou City Youth Science and Technology Innovation Talents Project under Grand No. 2023-QN-130, Gansu Provincial Science and Technology Major Project No. 22ZD6GA041, the Gansu Provincial Key Talent Project under Grant No. 6660010201, Youth Science and Technology Fund of Gansu Provincial Science and Technology Department No. 23JRRA729, and 2024 Graduate Education and Teaching Quality Improvement Construction Project under Grand No. JG202418.

Data Availability Statement: Data can be provide upon request.

Acknowledgments: We would like to express our sincere gratitude to Qingwei Li for his invaluable assistance in the development of Cesium. We also extend our thanks to Chuan Xu, Zhenyuan Chen, Junming Li, Huabin Ha, and Ziyue Zhang for their support during the UAV data acquisition process.

Conflicts of Interest: Author Chunyang Tang was employed by the company Silk Road Fantian (Gansu) Communication Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Open-Source Software and Tools Used in Our Work

CloudCompare

<https://www.cloudcompare.org/> (accessed on 12 December 2023).

WebODM

<https://www.opendronemap.org/webodm/> (accessed on 12 December 2023).

Open3D

<https://www.open3d.org/> (accessed on 5 March 2024).

Point Cloud Library

<https://pointclouds.org/> (accessed on 5 March 2024).

IfcOpenShell

<https://ifcopenshell.org/> (accessed on 15 May 2024).

obj2glTF

<https://github.com/CesiumGS/obj2glTF> (accessed on 15 May 2024).

3d-tiles-tools

<https://github.com/CesiumGS/3d-tiles-tools> (accessed on 15 May 2024).

CesiumJS

<https://cesium.com/platform/cesiumjs/> (accessed on 15 May 2024).

Cesium Terrain Builder

<https://github.com/geo-data/cesium-terrain-builder> (accessed on 12 Oct 2024).

GeoidEval utility

<https://geographiclib.sourceforge.io/cgi-bin/GeoidEval> (accessed on 12 Oct 2024).

References

1. Wu, B.; Xie, L.; Hu, H.; Zhu, Q.; Yau, E. Integration of aerial oblique imagery and terrestrial imagery for optimized 3D modeling in urban areas. *ISPRS J. Photogramm. Remote Sens.* **2018**, *139*, 119–132. [[CrossRef](#)]
2. Nesbit, P.R.; Hugenholtz, C.H. Enhancing UAV-SFM 3D model accuracy in high-relief landscapes by incorporating oblique images. *Remote Sens.* **2019**, *11*, 239. [[CrossRef](#)]
3. Zhu, Q.; Wang, Z.; Hu, H.; Xie, L.; Ge, X.; Zhang, Y. Leveraging photogrammetric mesh models for aerial-ground feature point matching toward integrated 3D reconstruction. *ISPRS J. Photogramm. Remote Sens.* **2020**, *166*, 26–40. [[CrossRef](#)]
4. Shakhatareh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* **2019**, *7*, 48572–48634. [[CrossRef](#)]
5. Van der Linden, S.; Okujeni, A.; Canters, F.; Degerickx, J.; Heiden, U.; Hostert, P.; Priem, F.; Somers, B.; Thiel, F. Imaging spectroscopy of urban environments. *Surv. Geophys.* **2019**, *40*, 471–488. [[CrossRef](#)]
6. Zhang, X.; Yue, P.; Chen, Y.; Hu, L. An efficient dynamic volume rendering for large-scale meteorological data in a virtual globe. *Comput. Geosci.* **2019**, *126*, 1–8. [[CrossRef](#)]
7. Qin, R.; Lin, L. Development of a GIS-based integrated framework for coastal seiches monitoring and forecasting: A North Jiangsu shoal case study. *Comput. Geosci.* **2017**, *103*, 70–79. [[CrossRef](#)]
8. Gu, L.; Zhang, H.; Wu, X. Surveying and mapping of large-scale 3D digital topographic map based on oblique photography technology. *J. Radiat. Res. Appl. Sci.* **2024**, *17*, 100772. [[CrossRef](#)]
9. Yuanyuan, F.; Hao, L.I.; Chaokui, L.I.; Jun, C. 3D modelling method and application to a digital campus by fusing point cloud data and image data. *Heliyon* **2024**, *10*, e36529. [[CrossRef](#)]
10. Yu, S.; Guo, T.; Wang, Y.; Han, X.; Du, Z.; Wang, J. Visualization of regional seismic response based on oblique photography and point cloud data. *Structures* **2023**, *56*, 104916.
11. Yu, K.; Li, H.; Xing, L.; Wen, T.; Fu, D.; Yang, Y.; Zhou, C.; Chang, R.; Zhao, S.; Xing, L.; et al. Scene-aware refinement network for unsupervised monocular depth estimation in ultra-low altitude oblique photography of UAV. *ISPRS J. Photogramm. Remote Sens.* **2023**, *205*, 284–300. [[CrossRef](#)]
12. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **2014**, *6*, 1–15. [[CrossRef](#)]
13. Zhou, X.; Zhang, X. Individual tree parameters estimation for plantation forests based on UAV oblique photography. *IEEE Access* **2020**, *8*, 96184–96198. [[CrossRef](#)]
14. Wang, S.; Yan, B.; Hu, W.; Liu, X.; Wang, W.; Chen, Y.; Ai, C.; Wang, J.; Xiong, J.; Qiu, S. Digital reconstruction of railway steep slope from UAV+ TLS using geometric transformer. *Transp. Geotech.* **2024**, *48*, 101343. [[CrossRef](#)]
15. Wang, J.; Wang, L.; Jia, M.; He, Z.; Bi, L. Construction and optimization method of the open-pit mine DEM based on the oblique photogrammetry generated DSM. *Measurement* **2020**, *152*, 107322. [[CrossRef](#)]
16. Yang, B.; Ali, F.; Yin, P.; Yang, T.; Yu, Y.; Li, S.; Liu, X. Approaches for exploration of improving multi-slice mapping via forwarding intersection based on images of UAV oblique photogrammetry. *Comput. Electr. Eng.* **2021**, *92*, 107135. [[CrossRef](#)]
17. Tang, S.; Li, X.; Zheng, X.; Wu, B.; Wang, W.; Zhang, Y. BIM generation from 3D point clouds by combining 3D deep learning and improved morphological approach. *Autom. Constr.* **2022**, *141*, 104422. [[CrossRef](#)]
18. Han, Y.; Feng, D.; Wu, W.; Yu, X.; Wu, G.; Liu, J. Geometric shape measurement and its application in bridge construction based on UAV and terrestrial laser scanner. *Autom. Constr.* **2023**, *151*, 104880. [[CrossRef](#)]
19. Barrile, V.; Fotia, A.; Candela, G.; Bernardo, E. Integration of 3D model from UAV survey in BIM environment. *The International Archives of the Photogrammetry. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 195–199.
20. Skrzypczak, I.; Oleniacz, G.; Leśniak, A.; Zima, K.; Mrówczyńska, M.; Kazak, J.K. Scan-to-BIM method in construction: Assessment of the 3D buildings model accuracy in terms inventory measurements. *Build. Res. Inf.* **2022**, *50*, 859–880. [[CrossRef](#)]
21. Zhu, Q.; Li, S.; Hu, H.; Zhong, R.; Wu, B.; Xie, L. Multiple point clouds data fusion method for 3D city modeling. *Geomat. Inf. Sci. Wuhan Univ.* **2018**, *43*, 1962–1971.
22. Jarzabek-Rychard, M.; Maas, H.G. Modeling of 3D geometry uncertainty in scan-to-BIM automatic indoor reconstruction. *Autom. Constr.* **2023**, *154*, 105002. [[CrossRef](#)]
23. Yuan, X.; Meng, D.; Ma, X. Application of multi-source data in 3D reconstruction of buildings in complex scenes. *Bull. Surv. Mapp.* **2022**, *6*, 143.
24. Abdelazeem, M.; Elamin, A.; Afifi, A.; El-Rabbany, A. Multi-sensor point cloud data fusion for precise 3D mapping. *Egypt. J. Remote Sens. Space Sci.* **2021**, *24*, 835–844. [[CrossRef](#)]
25. Wu, C.; Chen, X.; Jin, T.; Hua, X.; Liu, W.; Liu, J.; Cao, Y.; Zhao, B.; Jiang, Y.; Hong, Q. UAV building point cloud contour extraction based on the feature recognition of adjacent points distribution. *Measurement* **2024**, *230*, 114519. [[CrossRef](#)]
26. Mao, B.; Ban, Y.; Laumert, B. Dynamic online 3D visualization framework for real-time energy simulation based on 3D tiles. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 166. [[CrossRef](#)]

27. Xu, Z.; Zhang, L.; Li, H.; Lin, Y.-H.; Yin, S. Combining IFC and 3D tiles to create 3D visualization for building information modeling. *Autom. Constr.* **2020**, *109*, 102995. [[CrossRef](#)]
28. Zhan, W.; Chen, Y.; Chen, J. 3D tiles-based high-efficiency visualization method for complex BIM models on the web. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 476. [[CrossRef](#)]
29. Wang, L.; Li, C.; Dai, W.; Zou, J.; Xiong, H. QoE-driven and tile-based adaptive streaming for point clouds. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; IEEE: Piscataway, NJ, USA; pp. 1930–1934.
30. Skondras, A.; Karachaliou, E.; Tavantzis, I.; Tokas, N.; Valari, E.; Skalidi, I.; Bouvet, G.A.; Stylianidis, E. UAV Mapping and 3D Modeling as a Tool for Promotion and Management of the Urban Space. *Drones* **2022**, *6*, 115. [[CrossRef](#)]
31. Lu, M.; Wang, X.; Liu, X.; Chen, M.; Bi, S.; Zhang, Y.; Lao, T. Web-based real-time visualization of large-scale weather radar data using 3D tiles. *Trans. GIS* **2021**, *25*, 25–43. [[CrossRef](#)]
32. Woo, K.; Onsen, A.; Kim, W.S. Implementation of a 3D WebGIS for Dynamic Geo-Referencing of 3D Tiles on the Virtual Globe. *J. Geogr. Inf. Syst.* **2023**, *15*, 440–457. [[CrossRef](#)]
33. Qin, R.; Feng, B.; Xu, Z.; Zhou, Y.; Liu, L.; Li, Y. Web-based 3D visualization framework for time-varying and large-volume oceanic forecasting data using open-source technologies. *Environ. Model. Softw.* **2021**, *135*, 104908. [[CrossRef](#)]
34. Zhang, S.; Hou, D.; Wang, C.; Pan, F.; Yan, L. Integrating and managing BIM in 3D web-based GIS for hydraulic and hydropower engineering projects. *Autom. Constr.* **2020**, *112*, 103114. [[CrossRef](#)]
35. Evangelidis, K.; Papadopoulos, T.; Papatheodorou, K.; Mastorokostas, P.; Hilas, C. 3D geospatial visualizations: Animation and motion effects on spatial objects. *Comput. Geosci.* **2018**, *111*, 200–212. [[CrossRef](#)]
36. Fan, D.L.; Liang, T.L.; He, H.C.; Gou, M.Y.; Wang, M.H. Large-Scale Oceanic Dynamic Field Visualization Based on WebGL. *IEEE Access* **2023**, *11*, 82816–82829. [[CrossRef](#)]
37. Kopeć, A.; Bała, J.; Pięta, A. WebGL based visualisation and analysis of stratigraphic data for the purposes of the mining industry. *Procedia Comput. Sci.* **2015**, *51*, 2869–2877. [[CrossRef](#)]
38. Yang, Z.; Li, J.; Hyyppä, J.; Gong, J.; Liu, J.; Yang, B. A comprehensive and up-to-date web-based interactive 3D emergency response and visualization system using Cesium Digital Earth: Taking landslide disaster as an example. *Big Earth Data* **2023**, *7*, 1058–1080. [[CrossRef](#)]
39. Bi, T.; Yang, X.; Ren, M. The Design and Implementation of Smart Campus System. *J. Comput.* **2017**, *12*, 527–533. [[CrossRef](#)]
40. Pavón, R.M.; Alberti, M.G.; Álvarez, A.A.A.; Cepa, J.J. BIM-based Digital Twin development for university Campus management Case study ETSICCP. *Expert Syst. Appl.* **2025**, *262*, 125696. [[CrossRef](#)]
41. Luo, D.; Tan, G.; Wen, L.; Zhai, S. A study for 3D virtual campus navigation system based on GIS. In Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, China, 12–17 October 2008; IEEE: Piscataway, NJ, USA; pp. 1–5.
42. Chen, Y.; Shooraj, E.; Rajabifard, A.; Sabri, S. From IFC to 3D tiles: An integrated open-source solution for visualising BIMs on cesium. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 393. [[CrossRef](#)]
43. Qiu, Y.; Jiao, Y.; Luo, J.; Tan, Z.; Huang, L.; Zhao, J.; Xiao, Q.; Duan, H. A rapid water region reconstruction scheme in 3D watershed scene generated by UAV oblique photography. *Remote Sens.* **2023**, *15*, 1211. [[CrossRef](#)]
44. Alcaraz, C.; Lopez, J. Digital twin: A comprehensive survey of security threats. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1475–1503. [[CrossRef](#)]

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