International Journal for Housing Science and Its Applications

Publish September 9, 2025. Volume 47, Issue 1 Pages 261-278

https://doi.org/10.70517/ijhsa47122

Construction of a remote sensing big data-driven 3D modeling method for urban management

Yan Zhang[1,2], Yunkun Zou[1,2,*], Lingyuan Zhao[1,2], Yuhan Jiang[1,2], Shisen Liao[1,2] and Zifei Luo[1,2]

[1] HuanTian Wisdom Technology Co., Ltd., Meishan, Sichuan, 620564, China

[2] Joint Laboratory of Spatial Intelligent Perception and Large Model Application, Nanjing University of Aeronautics and Astronautics & HuanTian Wisdom Technology Co., Ltd., Meishan, Sichuan, 620564, China

Corresponding authors: (e-mail: zouyk@htwisdom.cn).

Abstract Urban management issues have become a significant challenge for urban administrators, and urban management must be based on modern computer technology and network communication environments. This paper investigates object-oriented building extraction methods, proposes a semi-global filtering method based on region growing for filtering remote sensing data, and then presents building 3D modeling methods and rendering methods in 3D remote sensing image modeling scenarios. In error comparison experiments between the classical TIN filtering algorithm and the proposed filtering method, the improved algorithm demonstrates superior performance, accurately identifying ground points at terrain steep changes, complex buildings, and complex roads. Its Type I error and total error are significantly lower than those of the classical TIN point cloud filtering algorithm, thereby demonstrating the superiority of the proposed improved algorithm.

Index Terms semi-global filtering method, 3D remote sensing, 3D modeling, urban management

I. Introduction

In today's world, big data is widely applied and has made significant contributions to the development of various fields. Meanwhile, the development of digital urban management also relies on big data to accelerate the pace of smart city construction [1], [2]. The advent of the big data era has transformed the traditional urban management system, enhancing urban management standards and improving the efficiency of addressing urban issues [3]. From everyday aspects of people's lives, such as clothing, food, housing, and transportation, to government decision-making and services, big data plays a pivotal role, making the construction of smart cities possible [4], [5]. Digital urban management not only embodies the implementation of the scientific development concept and promotes the construction of a harmonious society but also enhances the efficiency of government public management and increases public awareness of participation in urban management [6]-[9]. Therefore, researching big data smart applications for urban management holds significant importance.

In the era of big data, digital urban management requires managers to analyze more detailed data and phenomena, extract potential value information from them, and apply this information to urban planning [10]-[12]. Therefore, the more detailed the data, the better, and remote sensing big data technology can meet this requirement. Remote sensing big data technology offers advantages such as high resolution in terms of spatial, spectral, and temporal aspects [13]. For example, high spatial resolution corresponds to geographic spatial information elements. By leveraging this data, managers can promptly understand the detailed changes in urban management elements [14], [15]. Spectral information primarily refers to chlorophyll content, surface reflectance, etc. Managers can use this data to monitor the growth status and nutritional needs of urban vegetation [16]-[18]. Additionally, the high temporal resolution enables managers to quickly understand the historical development process of an entire city, meaning they can grasp the sequencing of urban planning and construction, identify changes in various elements, and ultimately develop refined management plans based on planning outcomes [19]-[22]. Therefore, urban intelligent comparison supported by remote sensing big data can provide comprehensive oversight for digital urban management applications.

Remote sensing data can meet the requirements of detailed urban management. The high-resolution remote sensing images it provides can clearly reflect relatively small details such as the structure, shape, and texture of ground features, thereby satisfying the requirements of element management. Literature [23] indicates that remote sensing data provides a rich source of data for studying the dynamic changes in urban landscapes. By utilizing real-time data to model, simulate, and predict urban development, it provides valuable information for urban planning by management personnel. Reference [24] discusses the processing and application of remote sensing big data technology in urban fine-grained management. Remote sensing big data technology enhances information coordination capabilities among urban departments and improves the level of intelligent

urban management, providing a technical basis for the construction of smart cities. Reference [25] investigated the feasibility of using remote sensing data and geospatial integrated data for urban research. The study demonstrated that the integrated data exhibits extremely high efficiency and complementarity, effectively reflecting the dynamic activities and economic characteristics within cities. Reference [26] emphasized the importance of land use information for urban planning and management. Different integration methods of remote sensing and geospatial data exhibit varying performance in expressing urban land use classifications, providing a deeper understanding for urban land use mapping. Reference [27] studied the integrated application of urban remote sensing data and social media data, which can accurately determine the accessibility of urban services and infrastructure, and is of great significance for optimizing urban management and promoting urban expansion. It is not difficult to see that remote sensing big data technology can transform urban management from extensive to intensive, which is more in line with the requirements of modern urban management.

Currently, cities across the country have accumulated a certain amount of comprehensive management data in the digital urban management applications of the big data era. However, there is significant research potential in how to utilize this data to achieve true smart, big data-driven urban management. Literature [28] evaluated intelligent technologies involved in analyzing remote sensing and geospatial data sources, finding that the combination of artificial intelligence and geospatial data can effectively enhance monitoring and management capabilities of regions, thereby providing effective support for regional management decision-making. Literature [29] utilizes spatio-temporal data fusion and deep learning methods to integrate high-resolution remote sensing data with other relevant information. This data-driven technology maintains high predictive capabilities even in the face of complex and dynamic urban changes, driving urban management toward sustainable development. Literature [30] proposes a novel time-series large-scale mapping method for precise land cover mapping in cities. By fully utilizing spectral and textural features from remote sensing data for comparative analysis, the land mapping results achieve high reliability and superiority, providing effective decision-making basis for urban planning and management. Although the aforementioned studies have successfully conducted in-depth mining and analysis of remote sensing big data, they lack consideration of its application in real-world scenarios. Remote sensing big data-driven 3D modeling methods have significant advantages in this regard and will drive the rapid development of smart city management.

The article first proposes an object-oriented building extraction method, which performs multi-scale segmentation based on remote sensing big data and designs classification algorithms based on image features to extract buildings.

II. Method

II. A. Building extraction method

II. A. 1) Object-oriented image analysis

Extracting data for target categories requires not only designing features based on the target categories but also segmenting them according to the differences and adjacent relationships between categories to reduce misclassification and omission. Object-oriented image analysis methods can fully utilize image features to extract information through object-based segmentation and classification. The quality of image segmentation directly impacts the classification effectiveness of the segmented objects. The internal features of objects formed through image segmentation exhibit a certain degree of consistency. The classification of information extraction requires the design and selection of appropriate classification features, and the establishment of suitable classification thresholds based on attribute feature differences to classify the segmented objects. Segmentation and classification primarily utilize the spectral, geometric, and textural features of images to group pixels with similar features into individual objects, and then classify them based on image features and perform other image analyses.

Image segmentation transforms images from individual pixels into independent entity units. Depending on the segmentation method, it is categorized into edge segmentation and region segmentation. Edge segmentation detects the edges of adjacent objects through gray-level transition points, connects edge points or lines to form closed contours, while region segmentation includes region splitting and region growing methods, which split or merge pixels into different segmentation units based on their feature similarities. Multi-scale segmentation is a segmentation method based on the principles of region growth and merging, and it is currently the mainstream method for image segmentation.

II. A. 2) Deep Learning Semantic Segmentation

Semantic segmentation is a classic digital image processing method that can divide an image into regions with different semantic meanings. Through feature extraction and inference operations from low-level to high-level, all pixels in the image are semantically annotated [31]. The following introduces the UNet semantic segmentation model and model optimization strategies.

(1) UNet Symmetric Semantic Segmentation Model

The UNet network model is widely used in medical image segmentation, and experiments have shown that it can also achieve good results in remote sensing image segmentation. This network has a U-shaped symmetric structure, with the left side being the contraction path for feature extraction and the right side being the symmetric restoration path for upsampling. The feature extraction component primarily obtains feature maps at different levels through convolution and pooling, then uses

upsampling to generate feature maps corresponding to each layer of feature extraction. Through skip connections and feature integration, the channel counts of the corresponding feature maps are combined, ensuring that the final output not only retains contextual information from the extracted features but also accurately preserves spatial information [32]. During model training, image cropping is employed to consider the multi-scale features of the research object while also performing data augmentation to obtain an adequate dataset, thereby enhancing the robustness of the training results.

(2) Model optimization

The primary data extracted for building usage in this study are DEM and digital surface model (DSM). The original image labels are obtained through human-computer interaction using geographic processing software. Subsequently, the generated four-band image map and corresponding labels are used to produce a sample dataset and perform preprocessing. To prevent memory overflow during training, the original images were first cropped using a sliding window and rotated to a 256×256 size sample dataset. Then, data augmentation, image normalization, and label encoding preprocessing methods were applied. Data augmentation primarily uses geometric transformations and color transformations to expand the sample dataset. The transformed images enhance the model's generalization ability and reduce the risk of overfitting during training. The min-max normalization method, as shown in Formula (1), is used to limit the pixel value range through min-max normalization, effectively eliminating the impact of outlier samples and accelerating the convergence of network training. Labels are encoded using one-hot encoding, where each category of flat labels is converted into a layer composed of 0s and 1s. The augmented dataset with a capacity of 15,345 is randomly divided into training, validation, and test sets in a 6:2:2 ratio.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

where $x_{\min}$ is the minimum value of the sample data, and $x_{\max}$ is the maximum value of the sample data.

This paper comprehensively utilizes features of different scales and depths through feature integration, using corresponding methods to complement different features to increase feature diversity. Model optimization is based on the U-net network structure. First, the input image dimensions are adjusted to 4 dimensions, and the kernel function dimensions of each layer are reduced by half to reduce the computational load. Next, the feature extraction and integration process is optimized, using the cross-entropy loss function to evaluate the loss of the current prediction results, as expressed in Formula (2), and selecting the Adam optimizer for backward propagation to update the model parameters.

$$Loss = -\sum_{i=1}^{S} yi.\log \hat{y}l \tag{2}$$

where S is the image size, y_i is the true pixel value, and y_i is the predicted pixel value.

II. A. 3) Object-oriented method for extracting buildings

The study area contains a large number of buildings, green spaces, farmland, and water bodies. A comparative analysis of different land cover features reveals that buildings typically have higher elevations and more regular shapes. However, due to the variety of roof materials and colors, they exhibit irregular spectral characteristics. Using EasyMap software, remote sensing data was imported, and a process tree was created to write a rule set for gradually extracting buildings. First, the imagery was subjected to multi-scale segmentation, with a scale set to 26, shape and compactness weights of 0.2 and 0.3 respectively, and an elevation mean range set to greater than or equal to 7. The algorithm classified building objects by setting the threshold condition to Mean DSM ≥ 7, and the category was designated as the building category.

Reviewing the classification results, it was found that some tall trees were also classified as buildings based on elevation. The gaps between trees caused uneven elevation data, resulting in wavy textures on the DSM image, while building objects had relatively uniform elevation. Therefore, trees and buildings could be preliminarily separated using elevation variance. The classification algorithm was used to remove objects with DSM variance ≥ 6 from the building category. After separation, some trees with fewer gaps between leaves and smaller elevation variances remain. These trees are more lush, resulting in relatively higher reflectance values in the green band. The green leaf index can be used to further separate trees from buildings. The custom green leaf index is defined by Formula (3), adjusted to an appropriate range, and the classification algorithm is applied again to separate the filtered vegetation from the building category.

$$GLI = \frac{((G - R) + (G - B))}{2G + R + B} \tag{3}$$

Among them, R, G, and B represent the red, green, and blue wavelengths, respectively.

II. B. Three-dimensional modeling methods for buildings

II. B. 1) Semi-global filtering method based on regional growth

In this study, the specific implementation method involves filtering out other objects such as buildings and vegetation from the DSM data, i.e., DSM filtering. The preparatory work for filtering involves dividing the DSM data into blocks, obtaining block DSM data, and using the block DSM data as input data to complete the algorithm [33]. The algorithm technical route is shown in Figure 1.

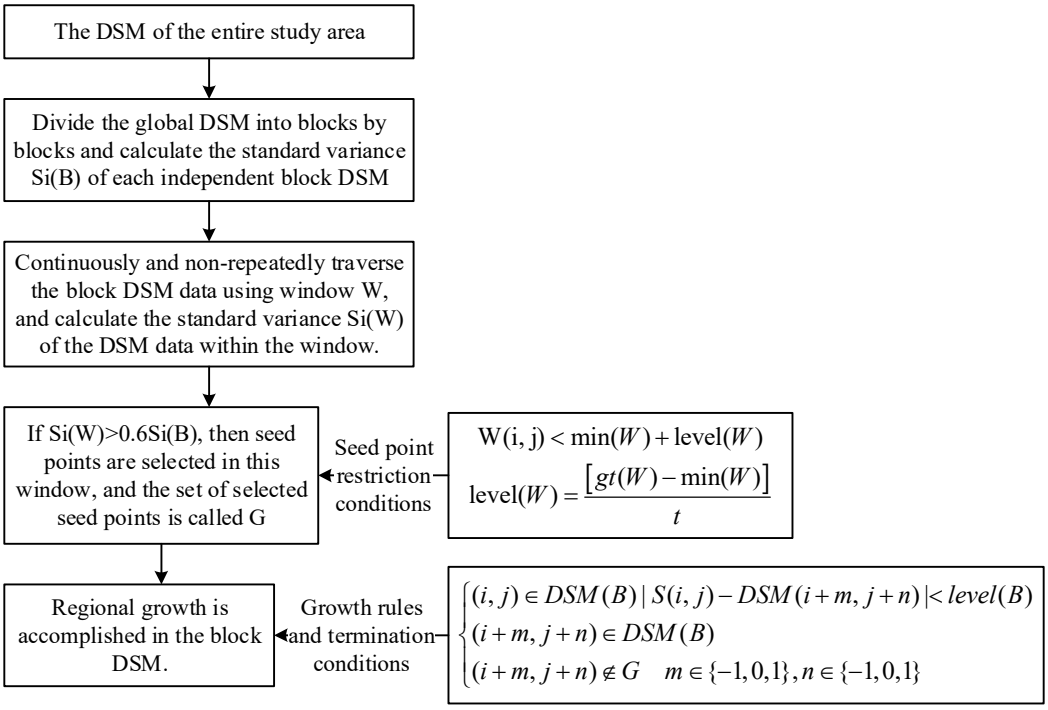

Figure 1: Flowchart of the DSM filtering algorithm

The steps for implementing DSM data filtering are as follows:

(1) Automatic selection of ground seed points first requires determining a square window W, where the size of window W must be larger than the average size of objects in the DSM data block. In this experiment, the selected size is 50×50. The automatic selection of seed points involves moving window W continuously and non-repeatedly across the DSM data block. If all pixels within window W satisfy formula (4), they are considered seed points. However, sometimes window W may completely cover a building or a flat terrain. In such cases, the former has no seed points, and the latter does not require seed points. This is because the former is an intact building, and the latter is relatively flat terrain, both of which can be identified through the growth operation of ground pixels in the surrounding window. To avoid errors caused by these two situations, the algorithm specifies that seed points can only be found in window W when the standard deviation of all pixels within window W is greater than 0.6 times the standard deviation of all pixels in the block DSM data.

$$W(i,j) < \min(W) + level(W) \tag{4}$$

In equation (4), W(i, j) is the pixel point located at (i, j) in the window. min(W) is the minimum value in the current window W. The formula for level(W) is as follows:

$$level(W) = \frac{[gt(W) - \min(W)]}{t} \tag{5}$$

In equation (5), gt(W) represents the threshold corresponding to the maximum interclass variance method in window W. t is a constant coefficient, and the value of level(W) changes with the window W. When the standard deviation of all pixels in window W is large, it indicates that window W is in a region with a steep slope. In this case, gt(W) will take a large value, and level(W) will also be relatively large. This allows for the selection of as many seed points as possible while ensuring the quality of the seed points. Therefore, the value of the threshold level(W) can ensure the adaptive selection of seed points in both flat and complex terrain regions.

(2) Adaptive selection of growth criteria and termination conditions

The growth process of the seed points automatically selected in the previous step is completed in each independent DSM data block. The growth criteria and termination conditions for regional growth in this experiment are as follows:

$$\begin{cases} (i,j) \in DSM \mid S(i,j) - DSM(i+m, j+n) \mid < level(B)(i+m, j+n) \in DSM \\ (i+m, j+n) \notin G \quad m \in \{-1,0,1\}, n \in \{-1,0,1\} \end{cases} \tag{6}$$

In equation (6), S(i, j) represents a seed point or a newly identified ground point during the growth process. Although level(B) is identical to equation (6), its range of values has changed: G is the set of ground seed points selected in the previous step. When the terrain around the seed points in set G is relatively flat, the elevation difference between the seed point and its adjacent ground points is relatively small. In this case, level(B) should take a smaller value; otherwise, excessive growth spread may occur, leading to an increased proportion of non-ground points among the extracted seed points and thereby increasing the error in the final results. The formula for level(B) is as follows:

$$level(B) = \frac{[gt(B) - \min(B)]}{t} \tag{7}$$

(3) Interpolation method

Regional growth can be used to obtain some ground point elevation information. However, this elevation information is not the actual elevation value of the ground. Due to the stringent growth criteria, the number of ground points selected in the above experimental process is limited. Therefore, to utilize this limited ground point elevation information to obtain DEM data, it is necessary to select an appropriate interpolation method. The interpolation method used in this study is the inverse distance weighted interpolation method (IDW). The principle of IDW is to determine the value of the interpolation point based on the inverse of the distance between the interpolation point and the sample points, i.e., the farther the interpolation point is from the sample points, the smaller the influence, and vice versa. The formula is as follows:

$$f(x,y) = \frac{\sum_{i=1}^{n} \left(\frac{1}{d_i^k} * Z_i \right)}{\sum_{i=1}^{n} \frac{1}{d_i^k}} \tag{8}$$

Among them, $f(x,y)$ is the predicted value at coordinate point (x,y). Z_i is the measured value at (x,y). n is the number of sample points surrounding the prediction point participating in the interpolation. d_i is the distance between the prediction point and each known sample point. k is the specified power, generally taken as 1-2.

II. B. 2) Three-dimensional modeling of buildings

(1) Building 3D Model Construction Algorithm

Based on building contour and height information, the Extruder algorithm is used to achieve 3D visualization of the building. The Extruder algorithm process is as follows:

(a) Determine the outer loop edges to be extruded, which will be extruded into faces. The algorithm considers edges belonging to two or more selected faces as inner edges and does not extrude them as part of the boundary loop.

(b) Extrude the determined outer loop edges into faces.

(c) If these loop edges belong to only one face in the entire mesh, all selected faces will be copied and connected to the newly created faces.

(d) If these loop edges belong to two or more faces, the selected faces will be connected to the newly created faces but will not be copied. This prevents unwanted faces from being created inside the mesh. This distinction is very important, as it ensures that the mesh structure produced by extrusion remains well-organized and closed.

(f) Edges that do not belong to the selected faces and form open loop edges will be copied during extrusion and connected to the original edges to form new faces.

(g) Isolated vertices that do not belong to the selected edges will be copied during extrusion and connected to the original vertices to form new edges.

(2) Methods for connecting buildings to terrain

Due to the undulations of the terrain surface, overlaying buildings onto a 3D terrain model may result in floating or sunken states. Based on input points and boundary lines, a Delaunay triangulation is constructed, and the input overhang features are overlaid onto the surface model. By overlaying buildings onto the terrain model, the minimum values of the corner points corresponding to the building's base contour are extracted. The lowest value among these minimums is selected as the building's base height, achieving the goal of flattening the building's base height. Based on this, using building height information and the Extruder algorithm, the overlay of 3D buildings onto terrain is realized.

(3) 3D modeling of building roofs

Building roof information cannot be directly obtained from the building vector data of the housing census. For the classification of building roofs, it is necessary to rely on the digital orthophoto imagery of the region.

Complex building models refer to buildings where adjacent structures are interconnected, and the building shapes are often irregular or have complex geometric forms. In actual modeling processes, the available parameters for complex building models are limited, making it difficult to achieve a fully accurate representation. Therefore, this study focuses on constructing simple building models and partial complex building models. The 3D building model construction process is illustrated in Figure 2.

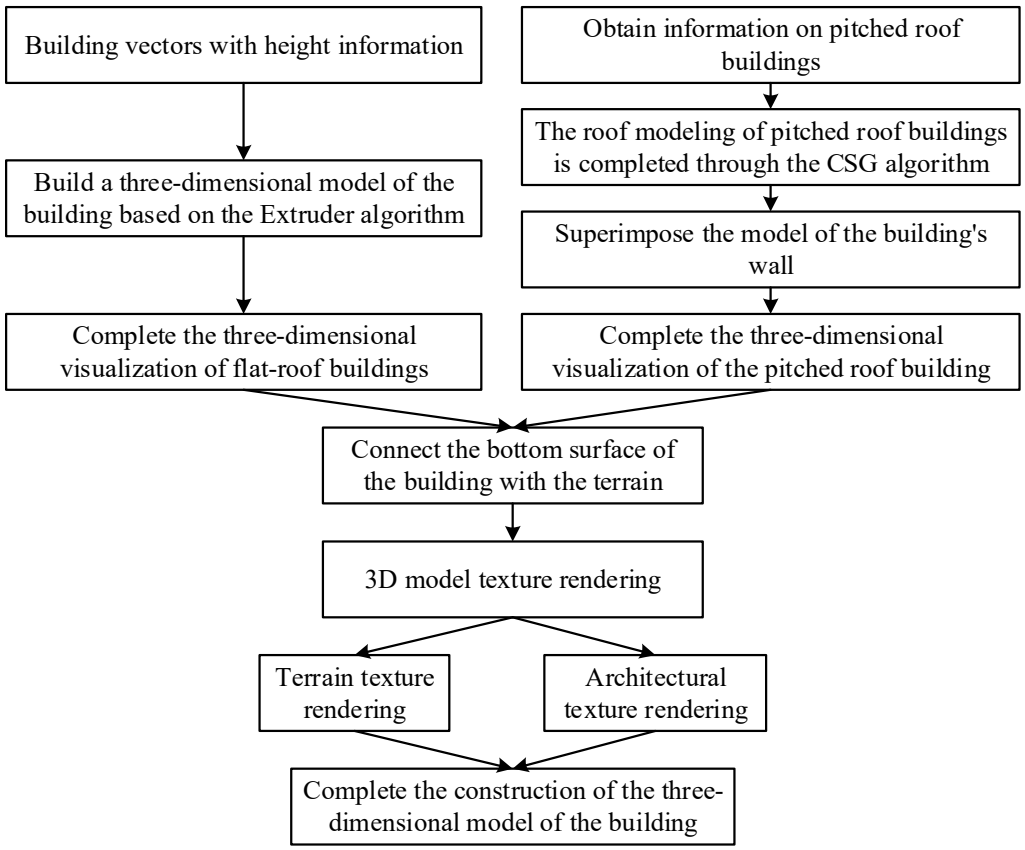

Figure 2: Flowchart for constructing a three-dimensional model of a building

II. B. 3) Potential Applications of Three-Dimensional Models of Buildings

The modeling objects of CIM primarily encompass three categories: urban buildings, infrastructure, and resource and environmental elements. Due to the scalability differences in CIM representation, the national, provincial, and municipal levels of CIM adopt a hierarchical construction approach. When representing object elements in CIM, it is necessary to consider different levels of detail in the model to accommodate the construction of CIM applications across various scales and scenarios. This means that during model data creation, data governance software suitable for CIM processing and handling should be used to achieve hierarchical and categorized construction of CIM. In CIM, the expression precision of each element is categorized into seven levels, with Level 1 being the least detailed and Level 7 being the most detailed.

II. C. Research on 3D remote sensing image rendering

II. C. 1) Physical simulation rendering model for remote sensing image modeling scenarios

In the construction of the diffuse BRDF, this paper employs the Lambert Diffuse model. In experiments conducted within the Unreal4 graphics engine rendering pipeline, this model achieves real-time diffuse rendering while maintaining the authenticity of rendering details, thereby reducing computational overhead in the rendering pipeline, while also meeting the requirements for PBR rendering effects.

$$f_{diff}(l,v) = F_0 / \pi \tag{9}$$

In equation (9), f_{diff} is the surface diffuse reflection component in the BRDF. Its input parameters l represent the light source direction and v represents the observation direction. F_0 is the surface reflectance of the rendered object, whose value is determined by the material's inherent reflective properties. In the PBR pipeline described in this paper, this data is calculated using two parameters, γ_m, γ_s, and the surface color parameter c_{base}. Among these, γ_m is the metallicity of the material surface, which represents the probability of photons being reflected off the material surface in the PBR rendering model. γ_s is the specularity, used to describe the basic reflective characteristics of the material surface. Meanwhile, c_{base} is often derived from the texture mapping process in the rendering pipeline, used to describe the base color of the material. Under the combined influence of the above three factors, the surface reflectance F_0 of an object can be expressed as in equation (10). The parameter γ_0 in the equation is a fixed value introduced for the convenience of adjusting γ_s and is only related to the internal settings of the rendering pipeline.

$$F_0 = C_{base} \cdot \gamma_m + \gamma_0 \gamma_s \cdot (1 - \gamma_m) \tag{10}$$

Compared to diffuse reflection, the calculation of the specular reflection BRDF term f_{spec} is more complex. This process requires comprehensive consideration of the effects of factors such as light, angle, and surface material on the rendering process. The rendering process can be abstracted as follows:

$$f_{spec}(l,v) = F(v,h)G(l,v,h)D(h) / (4(n \cdot l)(n \cdot v)) \tag{11}$$

In the above equation, n represents the normal direction of the grid model surface, while h represents the micro-normal direction in the Cook-Torrance micro-surface model. In most cases, this value can be calculated based on the relationship between the light source direction and the observation direction. When selecting a normal distribution function $D(h)$, the GGX model is widely recognized in the industry. In this model, the parameter γ_r is used to characterize the roughness of the material surface. When the normal direction n and micro-normal direction h are known, the model can be expressed as:

$$D_{GGX}(h) = \pi^{-1}\gamma_r^4((n \cdot h)^2(\gamma_r^4 - 1) + 1)^{-2} \tag{12}$$

For the Fresnel effect function $F(v,h)$, the Fresnel approximation method proposed by Schlick et al. is generally used in the field of computer graphics. This method uses the surface reflectance F_0 to simulate the Fresnel effect in the actual light propagation process, which is mathematically expressed as equation (13):

$$F_{Schlick}(v,h) = F_0 + (1 - F_0)(1 - (v \cdot h))^5 \tag{13}$$

However, in practical applications, since equation (13) requires multiple power operations, it can have a certain performance impact in real-time rendering. Therefore, in rendering environments that emphasize real-time performance, such as Unreal4, equivalent Fresnel calculation methods within the value range are often used. For example, the spherical Gaussian approximation method used in RealShading, as expressed in equation (14):

$$F_{SGA}(v,h) = F_0 + (1 - F_0) \cdot 2^{(-5 - 55473(v \cdot h) - 6.98316)(v \cdot h)} \tag{14}$$

For the calculation of the geometric attenuation factor $G(l,v,h)$, there are numerous different calculation methods in the industry, with the most mainstream being the calculation method based on separated occlusion shadows. This method assumes that the incident and reflective surfaces have the same effect on the light reflection attenuation process, allowing the symmetry to be directly utilized to derive the expression for $G(l,v,h)$ using $D(h)$ and the geometric shadow function. For example, the Schlick-GGX method used in RealShading:

$$G_{S-GGX}(l,v,h) = G_1(l)G_1(v) \tag{15}$$

Among them:

$$G_1(v) = n \cdot v / ((n \cdot v)(1 - k) + k) \tag{16}$$

The k term in the equation reflects the attenuation effect of the material surface on incident light, which is positively correlated with surface roughness. In the Schlick-GGX model, it is defined as the exponential model expressed by the equation:

$$k = 0.125(\gamma_r + 1)^2 \tag{17}$$

This allows us to construct the reflection parameter portion related to the material itself in a traditional PBR model, i.e., the f_r term in the equation. In actual PBR rendering, we also need to consider the impact of ambient lighting in the scene on the rendering results. In the construction of ambient lighting BRDF, the industry-standard method is image-based lighting (IBL) algorithms, whose main model is shown in the equation:

$$f_{ibl} = F_0 \pi^{-1} \cdot \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi/2} \cos\theta \sin\theta L(p,\varphi,\theta) d\theta d\varphi \tag{18}$$

The double integral term in the equation represents the effect of light rays caused by the ambient light centered at point p in the illumination space with spherical distribution of zenith angle θ and azimuth angle φ.

II. C. 2) Compression parameter texture atlas input level model

Based on the TAG texture atlas region generated by the PCG algorithm, the parameter information required for PBR rendering is packaged in a color information unit c_{para} in the form of color data channel component intensity in order of $\gamma_m, \gamma_s, \gamma_r$, thereby compressing and recording the rendering parameters for that location. The texture atlas is compressed using a channel separation-merging method, as shown in Figure 3.

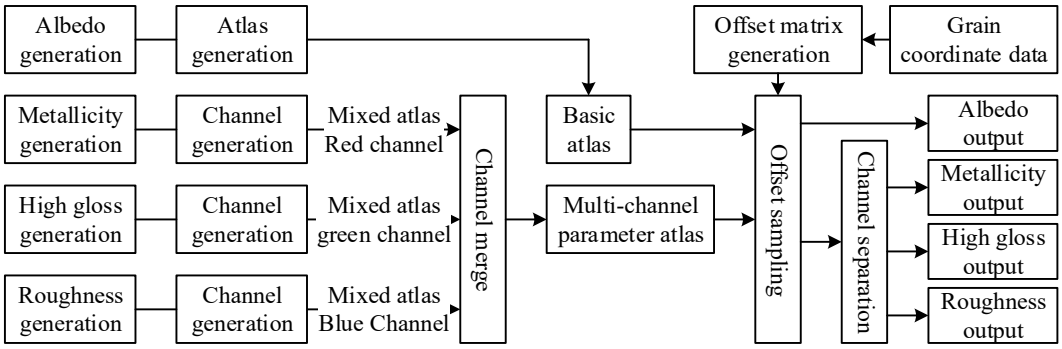

Figure 3: Compress the texture atlas using the channel separation and merging method

Following the TAG input-level model structure described above, two construction methods for the TACP input-level model are presented. The internal structure of the separated texture-type compressed parameter texture atlas input-level model is shown in Figure 4. The internal structure of the merged texture-type compressed parameter texture atlas input-level model is shown in Figure 5.

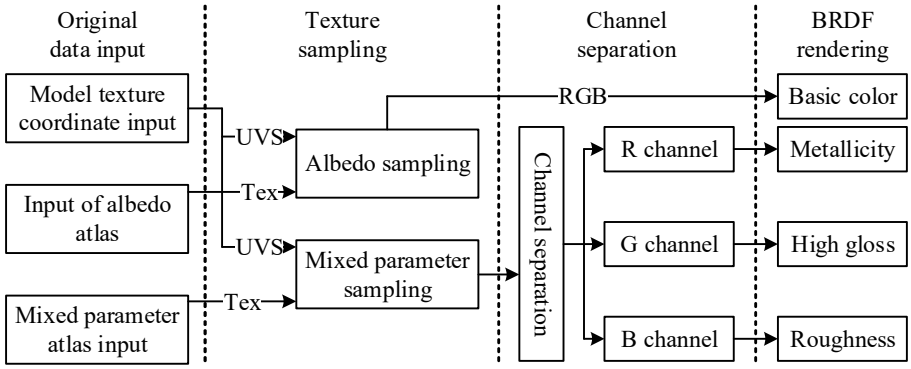

Figure 4: Separation texture-type compression parameter texture atlas input-level model

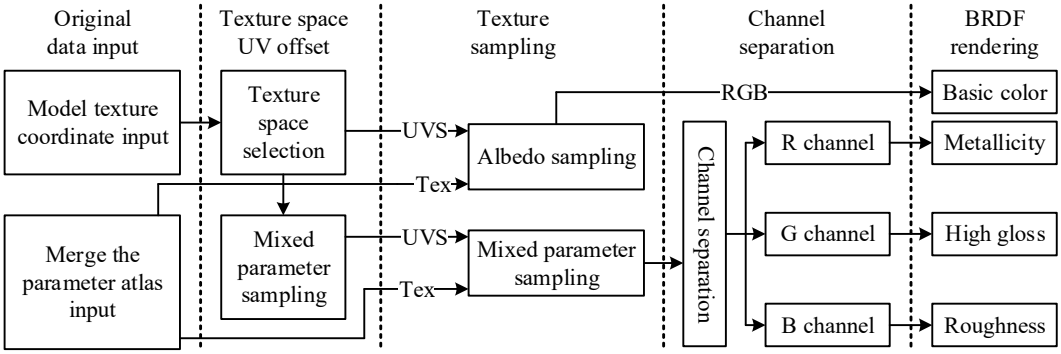

Figure 5: Merge textured compression parameter texture atlas input model

II. C. 3) Building Dynamic Radiation Rendering Model

In some scenarios, lighting does not come from external light sources, but from the rendered building components themselves. Therefore, in these scenarios, it is not enough to use only BRDF models based on reflected light; calculations based on light emitted by light sources must also be included. In this case, the original BRDF model based on the principle of conservation of reflected energy can be expanded into a model that obeys the reflection-radiation dual system, as shown in Equation (19):

$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_\Omega f_r(p,\omega_i,\omega_o) L_i(p,\omega_i)(\omega_i \cdot n) d\omega_i \tag{19}$$

This paper uses the point source radiation energy equation as the radiation emission term for position p in the BRDF model:

$$f_e(p) = C_{emi} \cdot I(t) \tag{20}$$

In the equation, the term c_{emi} represents a radiation reference color derived from an independent input, which determines the emission spectrum of the radiation light. $I(t)$ is a radiation intensity decay control term used to adjust the radiation intensity of self-luminous components on the building. For purely static rendering scenes, this term can be set to a constant value. This paper employs a time-varying dynamic attenuation function $T(\tau,\varphi,t)$ to simulate dynamic light sources in the scene, with the model expressed as:

$$f_e(p) = C_{emi} \cdot \delta_i \cdot T(\frac{t}{\tau} + \varphi) \tag{21}$$

The feature term δ_i in the equation represents the static intensity adjustment value obtained from the input stage of the rendering pipeline, while in the dynamic attenuation term $T(\tau,\varphi,t)$, t is a time variable from the GPU, whose value changes with the hardware device time, and τ and φ are the frequency control parameter and phase control parameter that control the dynamic attenuation effect, respectively.

II. C. 4) Pipeline structure for stylized rendering of three-dimensional remote sensing modeling scenes

This paper achieves the objective of improving the traditional rendering pipeline. The new PBR pipeline is suitable for rendering in three-dimensional remote sensing image modeling scenarios. Based on this rendering pipeline structure, direct associations can be established between various elements in the PBR rendering model and stylized 3D-PCG methods, thereby deriving the relationship model between each rendering component in this PBR pipeline and the results generated by the 3D-PCG algorithm for buildings within the remote sensing scene. As described in the preceding sections regarding the sampling process and parameter mixing process, the rendering parameter acquisition process for this rendering pipeline can be expressed as Equation (22):

$$(C_{base}, C_{emi}, C_{para})^T = S_L^{'} + diag(S_L - S_L^{'})[E_3, 0] C_{vtc} \tag{22}$$

In the equation, c_{vtc} represents the position coordinates of the vertex, s_L and $S_L^{'}$ are the results of the main texture sampling process and the stylized texture sampling process, respectively. If the texture offset matrices corresponding to each data item in the input layer are denoted as M_{base}, M_{emi}, and M_{para}, the main texture sampling process is denoted as $s(M_x, C_{UV})$, and the stylized texture sampling process is denoted as $s^{'}(M_x, C_{UV})$, then S_L and $S_L^{'}$ can be expressed as in equation (23):

$$S_L = \begin{bmatrix} s(M_{base}, C_{UV}) \\ s(M_{emi}, C_{UV}) \\ s(M_{para}, C_{UV}) \end{bmatrix}, S_{L'} = \begin{bmatrix} s'(M_{base}, C_{UV}) \\ s'(M_{emi}, C_{UV}) \\ s'(M_{para}, C_{UV}) \end{bmatrix} \tag{23}$$

Based on this, the expressions for the various BRDF parameters in this PBR pipeline can be obtained. According to the parameter separation method in the aforementioned TACP model and the parameter acquisition results of the rendering pipeline, the calculation process for surface reflectance can be expressed as Equation (24):

$$F_0 = C_{base} \cdot e_1^T C_{para} + \gamma_0 e_2^T C_{para} \cdot (1 - e_1^T C_{para}) \tag{24}$$

The Fresnel effect function represented by equation (25) can be directly obtained from the value of F_o:

$$F_{Schlick}(v,h) = F_0 + (1 - F_0)(1 - (v \cdot h))^5 \tag{25}$$

For the normal distribution function and geometric attenuation function, it is necessary to calculate the surface roughness parameter γ_r. Based on the generation process of c_{para} and the order of its parameters, this parameter can be obtained through matrix multiplication, thereby deriving the expression for the normal distribution function, as shown in Equation (26):

$$D_{GGX}(h) = \pi^{-1}(e_3^T C_{para})^4 / ((n \cdot h)^2((e_3^T C_{para})^4 - 1) + 1)^2 \tag{26}$$

Similarly, the expression for the single-surface attenuation parameter k in the geometric attenuation function can be obtained by the c_{para} operation in a similar manner, as shown in the equation:

$$k = 0.125(e_3^T C_{para} + 1)^2 \tag{27}$$

Similarly, based on the dynamic radiation BRDF model established by the formula, we can construct its radiation light rendering model expression under the rendering pipeline described in this paper:

$$f_e(p) = C_{emi} \cdot e_4^T C_{vtc} \cdot \sin((s(M_\tau, C_{UV}))^{-1} t + s(M_\varphi, C_{UV})) \tag{28}$$

III. Results and Discussion

III. A. Filter testing and algorithm performance evaluation

III. A. 1) Preparation of test data

(1) Experimental data

The experimental dataset consists of airborne LiDAR point cloud data from Vaihingen/Enz and Stuttgart, comprising seven sets of point cloud data from CSite1 to CStite7. Among these, Site1 to Site4 are urban area data with point spacings ranging from 0.8 to 1.7 m. Site5–Site7 are mountainous area data with point spacings ranging from 1.8 to 3.5 meters. The scene data include common terrain and features such as vegetation, bare ground, rivers, buildings, roads, railways, water bodies, bridges, and power lines. Additionally, 15 sample datasets were selected from the seven test scenes, with the characteristics of the test data summarized in Table 1. These sample datasets have been manually classified, with laser points precisely categorized into ground points and non-ground points. The 15 sample datasets broadly cover the primary challenges encountered in point cloud filtering, including the impact of coarse points, features connected to the ground, complex features, terrain discontinuities, vegetation on slopes, and low-lying vegetation, among others.

Table 1: Description of the characteristics of the test data

Test data	Sample number	Topographic features
CSite1	11	There are vegetation and buildings on the steep slopes
	12	More vehicles
CSite2	21	There is a bridge beam
	22	There is a bridge across the road
	23	Complex structure
	24	There are vegetation and steps on the steep slope
CSite3	31	The shape and structure of the building are complex
CSite4	41	There is a large amount of data missing area
	42	There are large buildings
CSite5	51	There is vegetation on the slope
	52	Steep slope, low vegetation, ridge
	53	Staircase form
	54	Smaller buildings
CSite6	61	Low vegetation
CSite7	71	Above the surface of the surface

(2) Selection of experimental parameters

The experiments employed both the classical TIN filtering algorithm and the DSM filtering method. The classical TIN filtering algorithm requires six parameters: the maximum building size $d_{\max}$, the steepest terrain slope angle $\theta_{\max}$, the iteration angle k_θ, the iteration distance k_d, the reduced iteration angle side length $S_{K\theta}$ and the stop mesh side length S_{stop} are required. In this experiment, the parameters $S_{K\theta}$ and S_{stop} were set to their default values, so only the first four parameters needed to be adjusted. In the DSM filtering method, a total of 9 parameters are required. In addition to the parameters of the classic TIN filtering algorithm, three additional parameters are needed to control the smooth surface segmentation of the point cloud, namely the number of neighboring points K_{near}, the normal vector angle difference threshold

T_θ and distance threshold T_d are added to control the segmentation of smooth surfaces in the point cloud. To facilitate algorithm comparison, the shared parameter values are the same for both algorithms in each group of scene experiments. The parameter values for each filtering parameter across the seven scenes are shown in Table 2.

Table 2: Explanations of the parameter values of the two filtering algorithms

Scene	$d_{\max}(m)$	$\theta_{\max}(^\circ)$	$k_\theta(^\circ)$	$k_d(m)$	K_{near}	$T_\theta(^\circ)$	$T_d(m)$
CSite1	21	75	6	1.7	25	32	0.8
CSite2	62	85	6	1.1	25	10	0.1
CSite3	35	90	4	1.7	25	25	0.8
CSite4	67	80	6	0.9	25	32	0.1
CSite5	11	90	6	1.2	10	32	0.3
CSite6	38	80	4	1.3	25	32	0
CSite7	8	89	6	1.4	25	32	0

III. A. 2) Analysis and evaluation of test results

Evaluating the performance of a filtering algorithm involves not only assessing the accuracy of its filtering results based on visual effects but also using quantitative technical metrics to determine whether the results align with the actual terrain structure. While organizing evaluations of classical filtering algorithms, the ISPRS group also established error evaluation criteria for filtering results. In its evaluation report, filtering errors are categorized into three types: Type I errors, Type II errors, and total errors. Type I error, also known as underclassification error, refers to the error of incorrectly classifying ground points as non-ground points. Type II error, also known as overclassification error, refers to the error of incorrectly classifying non-ground points as ground points. Total error refers to the proportion of all misclassified points out of the total number of points in the point cloud.

The probabilities of Type I errors, Type II errors, and total errors are the primary indicators for evaluating the performance of a point cloud filtering algorithm. The purpose of filtering is to maximize the recovery of the true terrain, and the number of non-ground points contained in the ground points obtained after filtering should be as small as possible. Therefore, most filtering algorithms focus on reducing the occurrence of Type II errors, while Type I errors remain significant. However, when manually correcting filtering errors, modifying Type II errors is much easier than modifying Type I errors. Therefore, while reducing Type II errors, we should also focus on minimizing the occurrence of Type I errors. The filtering parameters used by the two filtering algorithms for the Samp11 data are shown in Table 3. The error statistics of the Samp11 filtering results are shown in Table 4.

Table 3: Filtering parameters

Points(pts)	$d_{\max}(m)$	$\theta_{\max}(^\circ)$	$k_\theta(^\circ)$	$k_d(m)$	K_{near}	$T_\theta(^\circ)$	$T_d(m)$
38015	21	75	6	1.7	25	32	0.8

Table 4: Samp11 filtering error statistics

Sample labeling	Total points (pts)	Select the ground point (pts)	Filtering algorithm	Type I error (100%)	TypeII error (100%)	Total error (100%)
Samp11	38015	18812	Classic TIN filter	47.861	2.9855	30.2641
			DSM filtering method	26.053	6.5405	19.232

Samp23 Data: The data features a sparse distribution of vegetation and buildings. Within the building clusters, there is a unique closed circular terrain feature characterized by artificial building boundaries, with terrain points both inside and outside the boundaries. This data can be used to test the algorithm's ability to identify complex building structures. From the DEM results, both filtering algorithms can remove large buildings, retain ground points within the circular structure, and remove buildings along the circular boundary. However, when considering the retention of terrain points around the circular structure, the DSM filtering method performs better, with the generated DEM being closer to the reference DEM. In terms of error distribution and quantity, the errors of both filtering algorithms are primarily Type I errors, and the errors are mainly distributed around the circular structure. However, the DSM filtering method has fewer Type I errors. Overall, despite the high complexity of this scene, the improved algorithm performs better than the classical algorithm in terms of filtering effectiveness. The filtering parameters used by both algorithms for the Samp11 data are shown in Table 5. The error statistics of the Samp23 filtering results are shown in Table 6.

Table 5: Filter parameters for samp11 data by two filtering algorithms

Points(pts)	$d_{\max}(m)$	$\theta_{\max}(^{\circ})$	$k_{\theta}(^{\circ})$	$k_{d}(m)$	K_{near}	$T_{\theta}(^{\circ})$	$T_{d}(m)$
29115	62	85	6	1.1	25	10	0.1

Table 6: Samp23 filtering error statistics

Sample labeling	Total points(pts)	Select the ground point(pts)	Filtering algorithm	Type I error(100%)	TypeII error(100%)	Total error(100%)
Samp23	29115	6935	Classic TIN filter	36.5017	4.9825	21.9953
			DSM filtering method	19.6478	3.9857	12.3582

Samp42 data: This dataset has a high level of scene complexity, so 3D stereoscopic display was used here. Based on the number and distribution of errors, the errors from both filtering algorithms are primarily Type I errors, and they are mainly concentrated in the upper-right corner of the data. Additionally, both algorithms incorrectly retained some low-building points, but the DSM filtering method had fewer misclassified points. The filtering parameters for the two algorithms on the Samp42 data are shown in Table 7. The error statistics for the Samp42 filtering results are shown in Table 8.

Table 7: Filtering parameters of Samp42 data for two algorithms

Points(pts)	$d_{\max}(m)$	$\theta_{\max}(^{\circ})$	$k_{\theta}(^{\circ})$	$k_{d}(m)$	K_{near}	$T_{\theta}(^{\circ})$	$T_{d}(m)$
29115	67	80	6	0.9	25	32	0.1

Table 8: Samp42 filtering error statistics

Sample labeling	Total points(pts)	Select the ground point(pts)	Filtering algorithm	Type I error(100%)	TypeII error(100%)	Total error(100%)
Samp42	43261	9326	Classic TIN filter	11.4587	1.4964	4.7498
			DSM filtering method	8.1893	0.6661	5.2666

Samp53 data: The data features discontinuous terrain with numerous steep slopes and layered discontinuities. There is a small amount of low vegetation on the steep slopes, and no buildings are present. In terms of error counts, the DSM filtering method has significantly fewer misclassification points than the classical filtering method, with misclassification points primarily concentrated in Class II errors, while the classical algorithm's misclassification points are mainly concentrated in Class I errors. From the error distribution perspective, errors in the DSM filtering method are primarily concentrated in areas with smooth terrain changes, while errors in the classical algorithm are primarily concentrated in areas with abrupt terrain changes and fault structures. The DSM filtering method has fewer Type I errors, and Type II errors primarily occur at fault structures. The filtering parameters used are shown in Table 9. The error statistics for the Samp53 filtering results are shown in Table 10.

Table 9: Filtering parameters of Samp53 data for two algorithms

Points(pts)	$d_{\max}(m)$	$\theta_{\max}(^{\circ})$	$k_{\theta}(^{\circ})$	$k_{d}(m)$	K_{near}	$T_{\theta}(^{\circ})$	$T_{d}(m)$
29115	11	90	6	1.2	10	32	0.3

Table 10: Samp53 filtering error statistics

Sample labeling	Total points(pts)	Select the ground point(pts)	Filtering algorithm	Type I error(100%)	TypeII error(100%)	Total error(100%)
Samp53	35621	28654	Classic TIN filter	26.4043	2.2293	26.8879
			DSM filtering method	3.44	35.9745	5.7252

Two point cloud filtering algorithms were applied to all 15 sets of sample data from IPSRS for filtering experiments, yielding error statistics for both algorithms. The error comparison between the two filtering algorithms is shown in Table 11. From the statistical results of the 15 sets in the table, it can be observed that both the classical TIN filtering algorithm and the DSM filtering method exhibit strong robustness and can be widely applied to filtering in various scenarios across urban and forest areas. Additionally, compared to the classic TIN point cloud filtering algorithm, the DSM filtering method performs better,

accurately identifying more ground points, especially in areas with steep terrain changes, complex buildings, and complex roadways. Its Type I error and total error are significantly lower than those of the classic TIN point cloud filtering algorithm.

Table 11: Error comparison of two filtering algorithms

Scene	Error type	Classic algorithm	Improved algorithm	Scene	Error type	Classic algorithm	Improved algorithm
Samp11	I	47.861	26.053	Samp42	I	11.4587	8.1893
	II	2.9855	6.5405		II	1.4964	0.6661
	T	30.2641	19.232		T	4.7498	5.2666
Samp12	I	19.2147	8.1614	Samp51	I	4.4406	2.5625
	II	0.9997	4.6118		II	4.0287	8.03
	T	11.2496	5.06024		T	2.7364	3.7179
Samp21	I	0.4298	1.34361	Samp52	I	19.022	15.6357
	II	10.7339	17.8909		II	4.9763	7.9238
	T	1.5799	6.1252		T	14.9355	13.4742
Samp22	I	38.3099	19.6162	Samp53	I	26.4043	3.44
	II	3.6311	1.8983		II	2.2293	35.9745
	T	28.1676	15.8065		T	26.8879	5.7252
Samp23	I	36.5017	19.6478	Samp54	I	6.769	5.2134
	II	4.9825	3.9857		II	1.42741	5.819
	T	21.9953	12.3582		T	7.1716	5.4734
Samp24	I	38.9172	23.5445	Samp61	I	18.6128	17.0326
	II	12.8355	13.8022		II	4.7594	1.6014
	T	32.3749	20.7255		T	18.6443	15.5978
Samp31	I	5.8402	3.2739	Samp71	I	19.2563	10.4563
	II	3.0084	2.98		II	3.1247	12.8238
	T	3.944	3.3022		T	18.0669	11.6006
Samp41	I	60.9866	38.9762				
	II	0.8067	1.4038				
	T	31.3498	17.9483				

III. B. Model Accuracy Evaluation

III. B. 1) Theoretical accuracy assessment

(1) Image point accuracy of aerial triangulation densification

Assuming that the correction values for the coordinates of the points to be determined are random variables, the variance-covariance matrix of the coordinate correction values can be calculated using least squares adjustment, thereby determining the theoretical accuracy of the coordinates. The root mean square (RMS) of the image pixels can be used to evaluate the quality of the calculation results, with an RMS of no more than 0.5 pixels required to meet accuracy requirements. In this study area, aerial triangulation densification was performed based on ground control points. After joint adjustment, the mean error of the image points was determined to be 0.0013 mm. Since the mean error is less than 0.5 pixels, the image point accuracy of aerial triangulation densification meets the corresponding accuracy requirements.

(2) Checkpoint accuracy of aerial triangulation densification

After aerial triangulation densification and joint adjustment, the error of the check points is the difference between the field measurement coordinate values and the densified coordinate values calculated from the external orientation elements and their corresponding image point coordinates. A total of 38 control points were selected in the test area, and 10 of them were selected as check points. The residuals in the X, Y, and Z directions of the check points are shown in Table 12.

Table 12: The direction residuals of checkpoint X, Y, Z

Point number	R_X/m	R_Y/m	R_H/m
1	0.095	−0.034	0.057
2	0.133	−0.103	0.056
3	0.134	−0.022	0.248
4	0.013	−0.044	−0.167
5	−0.024	−0.092	−0.261
6	−0.047	−0.034	−0.411
7	−0.144	−0.036	0.185
8	−0.08	−0.031	0.241
9	−0.072	−0.024	0.211
10	−0.103	0.03	0.095

According to the formula, the error in the plane coordinates of the aerial triangulation control points is 0.106 m, with a maximum error of 0.165 m. Similarly, the error in the elevation coordinates is 0.221 m, with a maximum error of 0.411 m. The accuracy of the aerial triangulation control points meets the corresponding accuracy requirements.

III. B. 2) Evaluation of actual model accuracy

The actual accuracy assessment of the model includes the planar accuracy, elevation accuracy, height accuracy, and representative segment length accuracy of the model. Planar accuracy, elevation accuracy, and length accuracy are determined by measuring the coordinates and lengths of the model within the three-dimensional model system, followed by field measurements of the planar and elevation coordinates of the checkpoints. The field measurements of the checkpoints utilize the GPS RTK (1+N) measurement method described earlier to determine the geodetic coordinates of the checkpoints within the study area. The field checkpoint coordinates are used as the true coordinates of the checkpoints for evaluating the model's actual accuracy. The differences between the two sets of coordinates and their mean errors are then calculated to evaluate planar accuracy and elevation accuracy, respectively. The accuracy of the three-dimensional model is analyzed based on the planar mean error and elevation mean error of the solution points, and conclusions are drawn. The height accuracy of the model is determined by calculating the height of the building using the elevation coordinates of the building's top and bottom obtained from field measurements as the true value, comparing it with the model height data measured in the three-dimensional system, and then analyzing the model height accuracy. It should be noted that the theoretical accuracy of regional network aerial triangulation is influenced by random errors during the measurement process and the layout scheme of control points. In actual accuracy assessment, it is also influenced by the combined factors of random errors and residual systematic errors. Therefore, there is a certain discrepancy between the theoretical accuracy of aerial triangulation densification and the actual accuracy of field measurements.

(1) Planar accuracy

A total of 20 planar checkpoints were selected in the survey area, and the planar checkpoint accuracy statistics are shown in Table 13.

Table 13: Accuracy statistics of plane inspection points

Serial Number	X/m	Y/m	X'/m	Y'/m	ΔX / m	ΔY / m	ΔXY / m
1	36201.055	71029.763	36205.918	71028.974	0.083	-0.064	0.099
2	36360.906	70708.63	36358.572	70714.394	0.393	-0.268	0.458
3	36347.135	70712.844	36349.567	70712.33	0.085	-0.043	0.077
4	36305.35	70710.651	36306.389	70708.117	0.232	-0.284	0.309
5	36284.833	70703.477	36288.977	70702.463	0.16	-0.19	0.158
6	36249.482	70682.776	36247.97	70683.115	-0.093	-0.145	0.183
7	36213.473	70687.375	36210.411	70690.315	0.425	0.168	0.422
8	36333.716	70716.844	36331.468	70715.759	0.147	-0.128	0.195
9	36374.58	70715.348	36371.573	70715.586	-0.062	-0.119	0.134
10	36382.912	70671.533	36381.57	70668.159	-0.095	0.188	0.226
11	36398.015	70721.788	36398.639	70725.617	0.124	-0.185	0.22
12	36377.485	70714.358	36372.819	70715.888	-0.079	0.09	0.112
13	36202.797	70714.388	36205.62	70722.606	-0.037	-0.197	0.211
14	36194.002	70717.205	36194.469	70714.953	-0.283	0.354	0.443
15	36145.015	70714.541	36150.039	70708.924	-0.031	-0.077	0.116
16	36145.184	70712.592	36137.429	70712.727	0.095	-0.34	0.351
17	36269.719	70715.539	36278.058	70715.211	0.072	0.106	1.111
18	36372.381	70713.437	36374.294	70716.35	-0.097	0.143	0.167
19	36114.823	70723.863	36112.549	70719.119	-0.132	-0.113	0.134
20	36347.301	70711.477	36346.997	70716.665	0.207	0.161	0.265

Statistical results of planar accuracy for the regional model: The mean error in the X direction is 0.173 m, with a maximum error of 0.412 m in the positive direction and 0.231 m in the negative direction. The mean error in the Y direction is 0.176 m, with a maximum error of 0.363 m in the positive direction and 0.325 m in the negative direction. The mean error in the planar coordinates is 0.253 m, and the maximum error is 0.463 m. The planar coordinate differences in the model data are predominantly between 0 and 0.3 m, with a few between 0.3 and 0.5 m. The planar checkpoint accuracy meets the corresponding accuracy requirements.

(2) Elevation Accuracy

A total of 16 elevation checkpoints were selected in the survey area. The elevation accuracy statistics of the elevation checkpoints are shown in Table 14.

Table 14: Elevation accuracy statistics of elevation checkpoints

number	Field measured elevation	The model measures the elevation	Elevation difference
1	640.097	640.014	0.083
2	625.41	625.118	0.292
3	625.488	625.601	-0.113
4	627.208	627.188	0.02
5	627.82	627.94	-0.12
6	629.8	630.055	-0.255
7	631.703	631.771	-0.068
8	626.117	626.071	0.046
9	624.213	624.194	0.019
10	624.418	624.58	-0.162
11	626.543	626.318	0.225
12	625.437	624.902	0.535
13	630.57	631.465	-0.895
14	630.371	630.253	0.118
15	632.16	631.929	0.231
16	632.106	632.109	-0.003

Based on the above, the statistical results for elevation accuracy in the study area model are as follows: the maximum error in elevation coordinates is 0.895 m. The topographic accuracy of the 3D model can be described by the mean elevation error. However, different topographic conditions and mapping scales have different requirements for mean elevation error. The accuracy of elevation checkpoints meets the corresponding accuracy requirements.

(3) Height Accuracy

A total of 20 points were selected in the study area. The height of the buildings was calculated using the elevation coordinate values of the building tops and bottoms measured in the field as the true value, which was then compared with the model height data measured in the three-dimensional system. The model height accuracy was then analyzed. The building feature point height comparison table is shown in Table 15.

Table 15: Height comparison of building feature points

number	Field measurement /m	Model measurement /m	Height difference ΔH / m
H1	12.43	12.89	-0.46
H2	9.34	9.49	-0.15
H3	12.21	12.24	-0.03
H4	10.14	9.95	0.19
H5	11.48	11.75	-0.27
H6	12.63	12.19	0.44
H7	10.42	10.43	-0.01
H8	15.45	15.24	0.21
H9	11.03	11.14	-0.11
H10	12.41	12.86	-0.45
H11	9.33	9.54	-0.21
H12	15.6	15.66	-0.06
H13	9.11	9.11	0
H14	11.33	11.07	0.26
H15	10.76	10.91	-0.15
H16	9.66	9.43	0.23
H17	5.85	5.73	0.12
H18	8.92	9.2	-0.28
H19	15.36	15.27	0.09
H20	5.79	5.87	-0.08

Based on the above, the statistical results for the height accuracy of the study area model are as follows: the maximum error in height is 0.460 m. The accuracy of the height checkpoints meets the corresponding accuracy requirements.

(4) Length accuracy

Generally, buildings are composed of geometric structures, and the quality of these geometric structures has a significant impact on the final model quality. The key indicator for evaluating geometric structures is the length of line segments. Therefore, when evaluating model accuracy, it is also necessary to assess the lengths of representative line segments. In this study, 25 representative line segments were selected in the study area, and their corresponding lengths were measured in the 3D modeling system. The differences between the two were compared and analyzed, as shown in Table 16. The statistical results for length accuracy of the study area model: the maximum error in length is 0.34 m.

From the above-evaluated model's plane, height, and length accuracy values, it can be seen that the accuracy distribution of the three-dimensional modeling using remote sensing data in this study is relatively uniform and fully meets the "Three-dimensional Geographic Information Model Data Product Specifications."

Table 16: Comparison of representative line segment lengths

number	Foreign industry /m	Data length in 3D platform	Measured data - model data $\Delta L / m$
L1	10.37	10.32	0.05
L2	19.52	19.53	-0.01
L3	23	23.05	-0.05
L4	11.06	11.11	-0.05
L5	11.24	11.31	-0.07
L6	11.55	11.55	0
L7	14.46	14.66	-0.2
L8	5.22	5.38	-0.16
L9	11.52	11.49	0.03
L10	11.24	11.48	-0.24
L11	11.4	11.53	-0.13
L12	11.35	11.69	-0.34
L13	8.64	8.7	-0.06
L14	11.69	11.51	0.18
L15	11.4	11.63	-0.23
L16	11.37	11.41	-0.04
L17	6.31	6.13	0.18
L18	8.53	8.44	0.09
L19	11.7	11.58	0.12
L20	14.74	14.48	0.26
L21	12.36	12.3	0.06
L22	13.43	13.54	-0.11
L23	11.48	11.62	-0.14
L24	9.91	9.8	0.11
L25	10.73	10.5	0.23

IV. Conclusion

With the acceleration of urbanization, urban competition has become increasingly intense. As a result, urban management has emerged as a critical issue of widespread concern across nations and regions. This paper explores urban management three-dimensional modeling methods from three perspectives: building extraction techniques, three-dimensional modeling methods, and remote sensing image rendering. The conclusions of this article are as follows:

In the filtering experiments, this paper compares the performance of the classic TIN filtering algorithm and the region-based semi-global filtering method. The filtering comparison results show that the region-based semi-global filtering method has strong robustness and can be widely applied to various filtering scenarios in urban and forest areas. The improved algorithm performs better, with its Type I error and total error significantly lower than those of the classic TIN point cloud filtering algorithm. In the Samp11 Type I error, the computational error of the proposed algorithm is 21.808% lower than that of the classical TIN point cloud filtering algorithm.

In the actual accuracy evaluation experiment of the model, the planar coordinate error of the aerial triangulation densification checkpoints was calculated to be 0.106 m, and the elevation coordinate error was 0.221 m. The accuracy of the aerial

triangulation densification checkpoints meets the corresponding accuracy requirements. Therefore, the three-dimensional model constructed in this paper has good application effects.

Funding

This research was Supported by Open Project Funds for the Joint Laboratory of Spatial Intelligent Perception and Large Model Application (Grant No. SIPLMA-2024-YB-10).

References

[1] Shen, Z., & Li, M. (2018). Big data support of urban planning and management. In Advances in Geographic Information Science (p. 456). Springer International Publishing AG.
[2] Wu, M., Yan, B., Huang, Y., & Sarker, M. N. I. (2022). Big data-driven urban management: potential for urban sustainability. Land, 11(5), 680.
[3] Thakuriah, P., Tilahun, N. Y., & Zellner, M. (2016). Big data and urban informatics: Innovations and challenges to urban planning and knowledge discovery. In Seeing cities through big data: Research, methods and applications in urban informatics (pp. 11-45). Cham: Springer International Publishing.
[4] Babar, M., Arif, F., Jan, M. A., Tan, Z., & Khan, F. (2019). Urban data management system: Towards Big Data analytics for Internet of Things based smart urban environment using customized Hadoop. Future Generation Computer Systems, 96, 398-409.
[5] Zeng, C., Yang, L., & Dong, J. (2017). Management of urban land expansion in China through intensity assessment: A big data perspective. Journal of Cleaner Production, 153, 637-647.
[6] Silva, B. N., Khan, M., Jung, C., Seo, J., Muhammad, D., Han, J., ... & Han, K. (2018). Urban planning and smart city decision management empowered by real-time data processing using big data analytics. Sensors, 18(9), 2994.
[7] Sabri, S., & Witte, P. (2023). Digital technologies in urban planning and urban management. Journal of Urban Management, 12(1), 1-3.
[8] Csukás, M. S., Bukovszki, V., & Reith, A. (2020). Challenges and solutions for organizational design in urban digitalization. European Journal of Sustainable Development, 9(2), 615-615.
[9] Engin, Z., van Dijk, J., Lan, T., Longley, P. A., Treleaven, P., Batty, M., & Penn, A. (2020). Data-driven urban management: Mapping the landscape. Journal of Urban Management, 9(2), 140-150.
[10] Kandt, J., & Batty, M. (2021). Smart cities, big data and urban policy: Towards urban analytics for the long run. Cities, 109, 102992.
[11] Pan, Y., Tian, Y., Liu, X., Gu, D., & Hua, G. (2016). Urban big data and the development of city intelligence. Engineering, 2(2), 171-178.
[12] Rathore, M. M., Ahmad, A., Paul, A., & Rho, S. (2016). Urban planning and building smart cities based on the internet of things using big data analytics. Computer networks, 101, 63-80.
[13] Chi, M., Plaza, A., Benediktsson, J. A., Sun, Z., Shen, J., & Zhu, Y. (2016). Big data for remote sensing: Challenges and opportunities. Proceedings of the IEEE, 104(11), 2207-2219.
[14] Al-Najjar, H., Kalantar, B., & Abdul Halin, A. (2025). Advancements in land cover classification and machine learning techniques for Urban areas using remote sensing big data. Frontiers in Environmental Science, 13, 1584485.
[15] Kuffer, M., Pfeffer, K., & Persello, C. (2021). Special issue "remote-sensing-based urban planning indicators". Remote sensing, 13(7), 1264.
[16] Huang, Y., Tao, Y. U., & HUANG, X. Z. (2018). Agricultural remote sensing big data: Management and applications. Journal of Integrative Agriculture, 17(9), 1915-1931.
[17] Jiang, Y., Yin, J., & Zha, L. (2020). Discussion on remote sensing big data to promote the development of Smart City. Remote Sensing, 8(2), 1-5.
[18] Zhang, S., Xue, Y., Zhou, X., Zhang, X., Liu, W., Li, K., & Liu, R. (2022). State of the art: High-performance and high-throughput computing for remote sensing big data. IEEE Geoscience and Remote Sensing Magazine, 10(4), 125-149.
[19] Li, W. (2021). Mapping urban land use by combining multi-source social sensing data and remote sensing images. Earth Science Informatics, 14(3), 1537-1545.
[20] Du, S., Zhang, X., Lei, Y., Huang, X., Tu, W., Liu, B., ... & Du, S. (2024). Mapping urban functional zones with remote sensing and geospatial big data: A systematic review. GIScience & Remote Sensing, 61(1), 2404900.
[21] Boulila, W., Farah, I. R., & Hussain, A. (2018). A novel decision support system for the interpretation of remote sensing big data. Earth Science Informatics, 11(1), 31-45.
[22] Shi, S., Wang, H., & Ma, X. (2023, November). Key issues on application of remote sensing big data (rsbd): Concepts, scenarios, and challenges. In 2023 2nd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE) (pp. 238-245). IEEE.
[23] Yu, D., & Fang, C. (2023). Urban remote sensing with spatial big data: A review and renewed perspective of urban studies in recent decades. Remote Sensing, 15(5), 1307.
[24] Minghua, C. H. E. N. G., Xiangqiang, L. I., Fuxiao, Z. H. U., Dongchuan, Y. A. N., & Hua, L. E. I. (2020, June). Application of remote sensing big data technology in refined urban management. In 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE) (pp. 397-400). IEEE.
[25] Deng, X., Liu, P., Liu, X., Wang, R., Zhang, Y., He, J., & Yao, Y. (2019). Geospatial big data: New paradigm of remote sensing applications. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12(10), 3841-3851.
[26] Yin, J., Fu, P., Hamm, N. A., Li, Z., You, N., He, Y., ... & Dong, J. (2021). Decision-level and feature-level integration of remote sensing and geospatial big data for urban land use mapping. Remote Sensing, 13(8), 1579.
[27] Shao, Z., Sumari, N. S., Portnov, A., Ujoh, F., Musakwa, W., & Mandela, P. J. (2021). Urban sprawl and its impact on sustainable urban development: a combination of remote sensing and social media data. Geo-Spatial Information Science, 24(2), 241-255.
[28] Dritsas, E., & Trigka, M. (2025). Remote sensing and geospatial analysis in the big data era: A survey. Remote Sensing, 17(3), 550.
[29] Jadhav, S., Durairaj, M., Reenadevi, R., Subbulakshmi, R., Gupta, V., & Ramesh, J. V. N. (2024). Spatiotemporal data fusion and deep learning for remote sensing-based sustainable urban planning. International Journal of System Assurance Engineering and Management, 1-9.
[30] Ding, Q., Shao, Z., Huang, X., Altan, O., & Hu, B. (2022). Time-series land cover mapping and urban expansion analysis using OpenStreetMap data and remote sensing big data: A case study of Guangdong-Hong Kong-Macao Greater Bay Area, China. International Journal of Applied Earth Observation and Geoinformation, 113, 103001.

[31] Jundi Jiang,Yueqian Shen,Jinhu Wang,Yufu Zang,Weitong Wu,Jinguo Wang... & Vagner Ferreira. (2025). Boosted bagging: A hybrid ensemble deep learning framework for point cloud semantic segmentation of shield tunnel leakage. Tunnelling and Underground Space Technology incorporating Trenchless Technology Research,164,106842-106842.

[32] Chijioke Leonard Nkwocha & Ning Wang. (2025). Deep learning-based semantic segmentation with novel navigation line extraction for autonomous agricultural robots. Discover Artificial Intelligence,5(1),73-73.

[33] Amini Amirkolaee Hamed,Arefi Hossein,Ahmadlou Mohammad & Raikwar Vinay. (2022). DTM extraction from DSM using a multi-scale DTM fusion strategy based on deep learning. Remote Sensing of Environment,274.