

# A novel framework utilizing 3D Gaussian Splatting to construct building geometry for urban wind simulations

Peisheng Zhao <sup>a,b</sup>, Chao Li <sup>b,c</sup>, Jianxun Jiang <sup>b</sup>, Lingwei Chen <sup>b</sup>, Xiaolu Wang <sup>a</sup>, <sup>\*</sup>

<sup>a</sup> School of Environment and Civil Engineering, Dongguan University of Technology, Dongguan 523808, Guangdong, China

<sup>b</sup> School of Civil and Environmental Engineering, Harbin Institute of Technology, Shenzhen 518055, Guangdong, China

<sup>c</sup> Guangdong Provincial Key Laboratory of Intelligent and Resilient Structures for Civil Engineering, Harbin Institute of Technology, Shenzhen 518055, Guangdong, China



## ARTICLE INFO

**Keywords:**  
CFD  
Urban wind field  
3D Gaussian splatting  
Point cloud  
Architectural geometry

## ABSTRACT

Computational fluid dynamics (CFD) simulation is an essential method in urban wind environment assessment. Rapid and accurate establishment of detailed architectural geometric models is a crucial prerequisite for high-quality CFD simulations. Utilizing point cloud to generate these models is one of the mainstream methods. However, due to the lack of attention to the efficiency and accuracy of building point cloud generation, and the inability to extract sufficient details when establishing geometric topology, the geometric models generated by existing methods are not suitable for CFD simulations. To address these issues, this study proposes an algorithm framework based on 3D Gaussian Splatting. First, sparse points are generated using images taken by drones through COLMAP. Subsequently, a scene segmentation method based on these points is proposed to divide the entire scene into uniform blocks. For the first time, the 3D Gaussian Splatting is introduced and improved to generate high-precision building point clouds with a significant speed increase. Additionally, an integrated algorithm is designed to extract vegetation, terrain, and buildings in the point cloud, then simplify and refine the building contours to generate geometric models suitable for CFD simulation. Compared with traditional methods, this framework successfully extends the scene to the city scale with adjustable details. The accuracy of point clouds improves by an average of 12%, the generation speed increases by 3–5 times, the geometric model detail level reaches LoD2 and LoD2.5, and can be further improved with customized flight plannings. Finally, pedestrian comfort analysis based on CFD simulations and WebGIS visualization was conducted, with the simulation results monotonically converging, and the grid convergence index reaching 3.76% in the pressure field. These results indicate that the algorithm framework is suitable for typical urban wind field simulations.

## 1. Introduction

Urban wind environment research occupies a central position in the interdisciplinary fields of environmental science, geography, and climatology, significantly impacting urban safety, economy, and health (Yi & Zheng, 2021). Computational Fluid Dynamics (CFD) is widely used in urban wind environment research (Frahat & Arisha, 2023; Oh, Yang, & Choi, 2024; Zhao, Li, Cao, Yi, & Liu, 2024), where the level of detail in architectural geometric models directly influences the quality of CFD results. Traditional manual methods such as CAD usually require a significant amount of time, making it difficult to meet the demand for numerous architectural geometric models needed for urban wind environment research. With the development of urbanization, the pace of urban building iteration and renewal has accelerated. In the face of

emergencies, rapidly generating accurate urban architectural geometry is crucial for urban wind environment research.

In existing research, building models for large urban areas can be rapidly generated by stretching building outline vector maps based on height information using GIS data (Kwon & Kim, 2014; Qiu, He, Li, & Zhu, 2023). However, the geometric models generated by GIS suffer from issues such as missing details and insufficient accuracy of building height information. Additionally, GIS data is challenging to update in time. In contrast, establishing geometric models through point clouds can obtain real-time data and capture more building details. This method is generally divided into two steps: building point cloud generation and geometric topology generation. Traditional researches generate building point clouds by using airborne LiDAR or cameras to capture optical images and employing multi-view geometry

\* Corresponding author.

E-mail address: [wangxiaolu@dgut.edu.cn](mailto:wangxiaolu@dgut.edu.cn) (X. Wang).

algorithms (MVS) (Furukawa & Ponce, 2009; Galliani, Lasinger, & Schindler, 2015; Zhou, Wang, Love, Ding, & Zhou, 2019). Alternatively, methods using neural networks such as Convolutional Neural Networks (CNNs) (Han, Leung, Jia, Sukthankar, & Berg, 2015; Menze & Geiger, 2015) and MVSNet (Yao, Luo, Li, Fang, & Quan, 2018; Yu & Gao, 2020) have gradually surpassed traditional methods in both speed and accuracy. For geometric topology generation, Poisson surface reconstruction (Kazhdan, Bolitho, & Hoppe, 2006), Delaunay triangulation algorithm, and alpha-shape (Edelsbrunner, Kirkpatrick, & Seidel, 1983) are commonly used to reconstruct building point clouds into geometric models. However, these methods in practical applications have some issues.

Firstly, the depth estimation algorithms they used cannot segment and parallelize the scene when processing large volumes of data, resulting in significant disadvantages in speed, accuracy, and memory consumption. Secondly, the generated building geometric models exhibit defects such as uneven surfaces, holes, and protrusions (Edelsbrunner et al., 1983; Kazhdan et al., 2006). Additionally, they are extremely sensitive to the surrounding vegetation points, often causing buildings to stick to vegetation (Yu & Gao, 2020), making them unsuitable for direct application in CFD studies.

In recent years, the rapid development of deep learning has significantly advanced the speed and accuracy of 3D reconstruction. For example, Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) initialize the scene as an implicit radiance field, producing outcomes that closely approximate reality. The challenge of implementing MVS in city-level 3D reconstruction is effectively addressed by dividing the scene into different levels of detail or segmenting it into equally sized blocks for training (Tancik et al., 2022; Turki, Ramanan, & Satyanarayanan, 2022). The NeuS method (Wang et al., 2021) trains Signed Distance Fields (SDF) to directly generate building geometric models. 3D Gaussian Splatting (3DGs) (Kerbl, Kopanas, Leimkühler, & Drettakis, 2023) uses 3D Gaussian ellipsoids to replace point clouds for initializing and rendering 3D scenes, thereby rapidly obtaining dense building point clouds. CityGaussian (Liu et al., 2024) expands the scene to the city level by pre-training a rough scene prior and then dividing the entire scene accordingly for parallel training.

On the other hand, the generation of geometric models from point clouds has also developed rapidly (Heidari, Navimipour, & Unal, 2022). The Open Geospatial Consortium CityGML 2.0 standard classifies architectural details into five levels: LoD0, LoD1, LoD2, LoD3, and LoD4 (van Rees, 2013). Among them, LoD1 means the building geometry model contains only height information without roof details. LoD2 adds details to the main roof. LoD3 builds on LoD2 by adding the smaller elements of the roof. Gu, Zhang, Shuai, Xu, and Xu (2024) proposed a workflow for urban wind field simulation based on drone photogrammetry. This workflow utilizes technologies such as deep learning and geometric complexity quantification to construct 3D models of urban building clusters for CFD simulations. Alemayehu and Bitsuamlak (2022) established an architectural model with an LoD3 level of detail by combining GIS building contour data with point cloud data. Fu, Paden, and García-Sánchez (2024) established vegetation geometries with different levels of detail using point clouds and conducted CFD simulations to study the impact of different LoD of tree models on urban wind fields. Paden, Peters, García-Sánchez, and Ledoux (2024) proposed a workflow for urban microscale simulation, capable of generating architectural geometry with detail levels of LoD1.2, LoD1.3, and LoD2.2 (Ricci et al., 2017), respectively. Younis, Bitsuamlak, and Sushama (2024) simplified the BIM building model and constructed a high-resolution regional climate simulation-driven CFD model to study the thermal performance of elevated buildings in permafrost regions of the Arctic under the background of climate change.

There are two crucial preconditions for establishing urban building geometry suitable for CFD calculations: the rapid generation of large-scale dense building point clouds and the extraction and construction of geometric models with detailed roof structures. However, existing

studies encounter issues of memory and time consumption at the city level (Fu et al., 2024; Liu et al., 2024; Mirzaei, 2021). Multiple devices and resources are typically required to complete the task, with additional time spent training a rough scene representation as a prior (Alemayehu & Bitsuamlak, 2022; Fu et al., 2024). Currently, although some studies attempt to use dense point clouds to generate geometric models suitable for CFD (Alemayehu & Bitsuamlak, 2022; Fabbri & Costanzo, 2020; Sun et al., 2021), they often neglect the speed and quality of point cloud generation. The resulting geometric models exhibit low detail levels and fail to retain sufficient building roof details. Additionally, buildings have not been effectively separated from surrounding vegetation, resulting in adhesion between buildings and vegetation in the generated geometric models. Therefore, further exploration is necessary to generate high-detail geometric models at the city level that meet the needs of CFD research while ensuring both speed and quality.

To address the issues of memory and time consumption in generating building geometry models suitable for urban CFD research, while improving the level of detail in the models, this paper proposes an automatic building geometry generation framework based on 3DGs. The framework includes dense point cloud generation and building geometric model generation. Taking Shenzhen, Guangdong Province as an example, the initial point cloud is generated based on Structure-from-Motion (SfM) (Snavely, Seitz, & Szeliski, 2006) using a drone-captured dataset. A “divide and conquer” strategy is proposed, dividing the scene based on the mapping relationship between 3D points and 2D image feature points, training each block in parallel to accelerate point cloud generation, and expanding the reconstructed scene to the city level. For the first time, 3DGs is introduced and improved, initializing the scene as Gaussian points and continuously densifying to obtain dense point clouds. Improved VDVI (Wang, Wang, Wang, & Wu, 2015) is used to extract vegetation point clouds, followed by the design of a height scanning algorithm to extract terrain points. The RANSAC (Fischler & Bolles, 1981) algorithm and frequency statistics are then combined to extract candidate roof planes, and DBSCAN (Ester, Kriegel, Sander, Xu, et al., 1996) clustering combined with point density is used to filter out the final planes. Morphological erosion and dilation, along with the Canny operator (Canny, 1986), are used to extract and optimize the plane edge contours. Finally, pedestrian comfort assessment based on CFD simulations and WebGIS visualization was conducted to demonstrate the future applicability of this framework.

## 2. Method

The algorithm framework proposed in this study primarily consists of two parts: dense point cloud generation and building geometric model establishment. The workflow of this framework is shown in Fig. 1. First, the image datasets captured by the drone are downsampled to reduce the resolution. Then, using the open-source software COLMAP, Structure-from-Motion (SfM) (Snavely et al., 2006) is employed to generate the initial sparse point cloud and the mapping relationship between 2D feature points and 3D points. Based on the sparse point cloud and the mapping relationship, the entire scene is divided into multiple blocks. Using the improved GaussianPro model, parallel-trained blocks are employed to generate dense building point clouds. Finally, an integrated algorithm is proposed to obtain regular building geometric models, which includes extracting vegetation and building point clouds, and refining building plane contours.

### 2.1. Scene division

#### 2.1.1. Generation of sparse point clouds

The 3DGs model and scene segmentation require camera parameters and sparse point clouds as inputs. COLMAP (Schönberger, Zheng, Frahm, & Pollefeys, 2016), an open-source software, is commonly used to preprocess datasets to obtain these data. Initially, SIFT (Lowe,

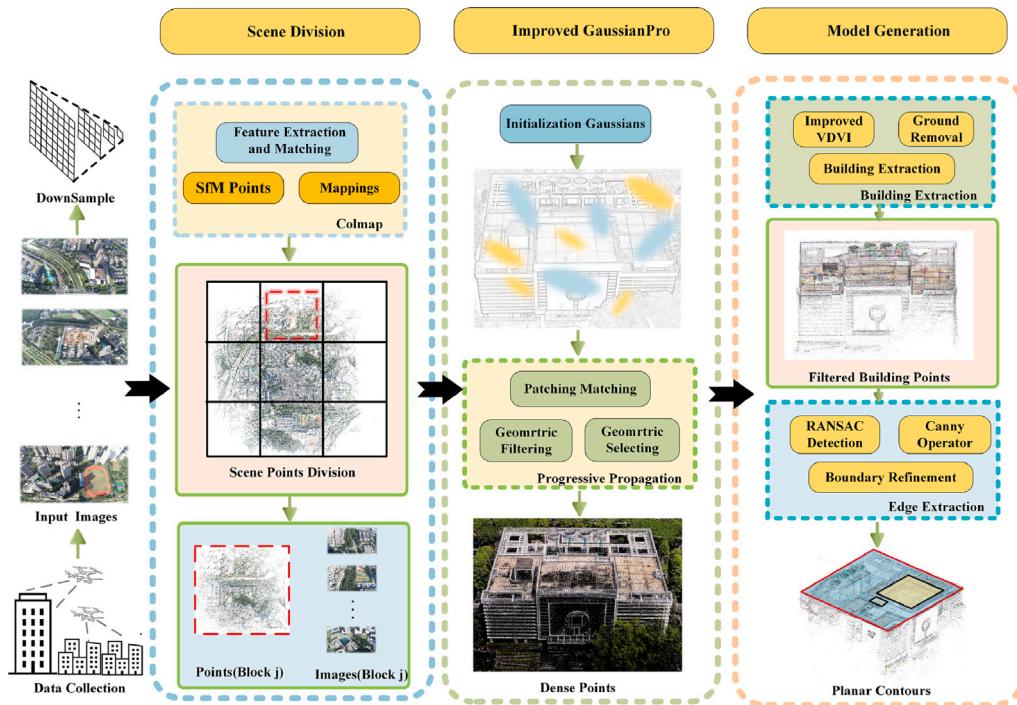


Fig. 1. Overall workflow of the algorithm framework.

2004) is used to extract feature information from the images, which consists of the number and location of feature points, the scale of the camera's distance from the object, and gradient directions. These feature information are then used for feature matching and geometric verification (Lowe, 1999). Subsequently, two images with good feature matching results are selected, their poses are estimated, and 3D points are generated through triangulation. The next matching image is selected based on the principle of the highest number of matching points, and the pose estimation and triangulation operations are repeated to obtain the sparse point cloud and camera parameters. The flowchart of data processing is shown in Fig. 2.

#### 2.1.2. Data partitioning based on mapping relationships

Due to limitations in computing resources such as video memory, directly reconstructing large urban scenes is evidently challenging. This paper adopts a divide-and-conquer strategy similar to Mega-NeRF (Turki et al., 2022), which partitions the scene based on camera parameters. The mapping relationships between 2D keypoints in images obtained by COLMAP and sparse points are used to accurately partition the scene into uniform blocks. Each block is then assigned a 3DGS model for training. The specific scene partitioning and data segmentation methods are described as follows.

Specifically, after extracting feature points (2D keypoints) from each image, feature matching pairs the images and feature points. Then the initial points (3D points) are obtained through sparse reconstruction. As shown in Fig. 3, this process establishes the mapping relationship between 3D points in the point cloud and 2D keypoints in the images. This study first uses statistical outlier removal (SOR) filtering to eliminate outlier noise points. Next, a 3D bounding box is used to enclose the entire scene along the point cloud boundary, and it is divided into uniform blocks along the xy-plane. The dataset is partitioned based on the mapping relationship between the 3D points in each subregion and the 2D keypoints in the images. Notably, the 3D points in the sparse point cloud correspond to keypoints present in multiple images. Similarly, the 2D keypoints in an image have mapping relationships with the 3D points of multiple blocks. Therefore, a ratio threshold is set to filter out images that contribute significantly to a subregion.

When the proportion of 2D keypoints corresponding to a subregion in an image exceeds this threshold, the image is considered to contribute significantly to that subregion and is thus assigned to it. After data partitioning, each block contains the corresponding sparse point cloud, images, and camera parameters.

#### 2.2. Dense point cloud

##### 2.2.1. 3D Gaussian splatting

This study employs the 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) method to generate dense point clouds of buildings. Unlike traditional MVS methods, this method does not require the estimation of normal vectors, significantly accelerating the reconstruction process. The input for 3DGS comprises a set of images and the SfM sparse points. Initially, each SfM point is initialized as a 3D Gaussian distribution with properties of position mean ( $\mu$ ), covariance matrix ( $\Sigma$ ), and opacity ( $\alpha$ ). This ellipsoid-like and differentiable distribution facilitates continuous training to optimize and densify it. The Gaussian distribution  $G(x)$  is defined by its center point (mean) and the full 3D covariance matrix  $\Sigma$  (Zwicker, Pfister, Van Baar, & Gross, 2001):

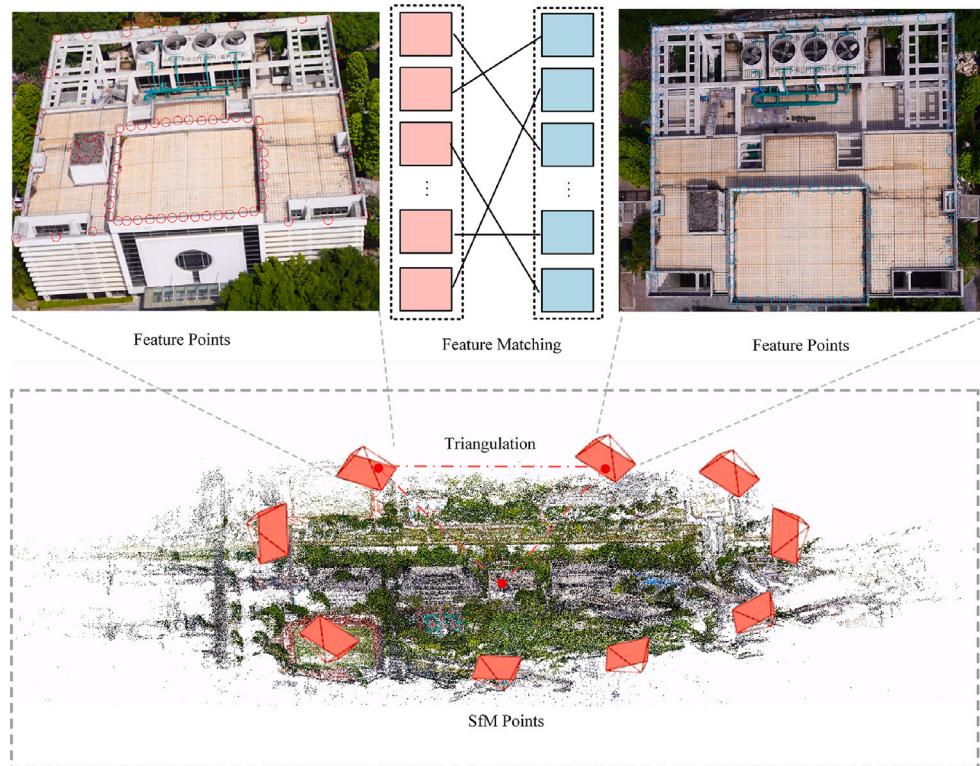
$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (1)$$

Clearly,  $\Sigma$  is geometrically differentiable and can be directly optimized to alter the position, size, direction, etc., of the Gaussian ellipsoid to obtain an accurate 3D Gaussian distribution. The covariance matrix  $\Sigma$  is defined to describe an ellipsoid, which ensures that it is positive semi-definite and physically meaningful. Therefore, it can be obtained:

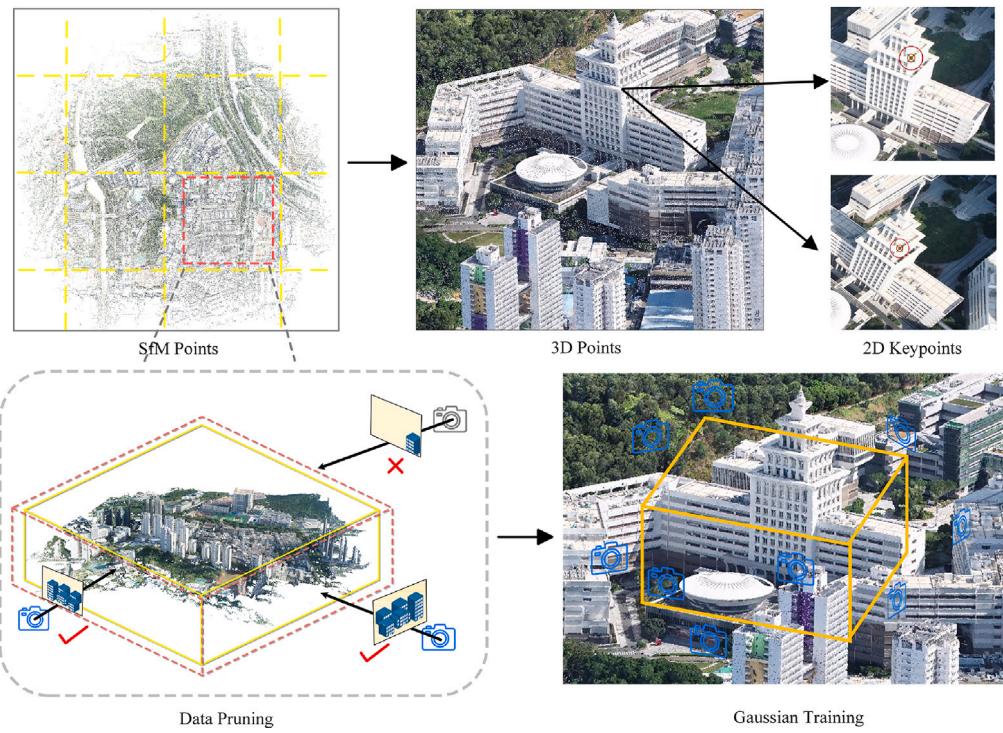
$$\Sigma = \mathbf{R} \mathbf{S}^T \mathbf{R}^T, \quad (2)$$

where  $\mathbf{R}$  is the rotation matrix, and  $\mathbf{S}$  is the scale matrix.

After initialization, the Gaussian distribution may have regions lacking geometric features and regions where the Gaussians cover excessively large areas. These regions are referred to as under-reconstructed and over-reconstructed regions, respectively. Adaptive Gaussian density control can optimize these areas. By replicating existing Gaussians in the under-reconstructed areas and splitting Gaussians in the over-reconstructed areas, these issues can be effectively addressed.



**Fig. 2.** The working principle of Structure-from-Motion.



**Fig. 3.** Scene partitioning based on mapping relationships.

This framework builds upon GaussianPro (Cheng et al., 2024) to generate point clouds and further optimizes adaptive Gaussian density control based on 3DGS. Compared to 3DGS, which directly clones and prunes Gaussian points, GaussianPro uses a progressive propagation

strategy to supervise the densification of Gaussian distributions, resulting in more accurate Gaussian scenes. Specifically, 3D Gaussians are first projected onto a 2D plane to form normal and depth maps, as shown in Fig. 4. Then, each pixel in the depth and normal maps is

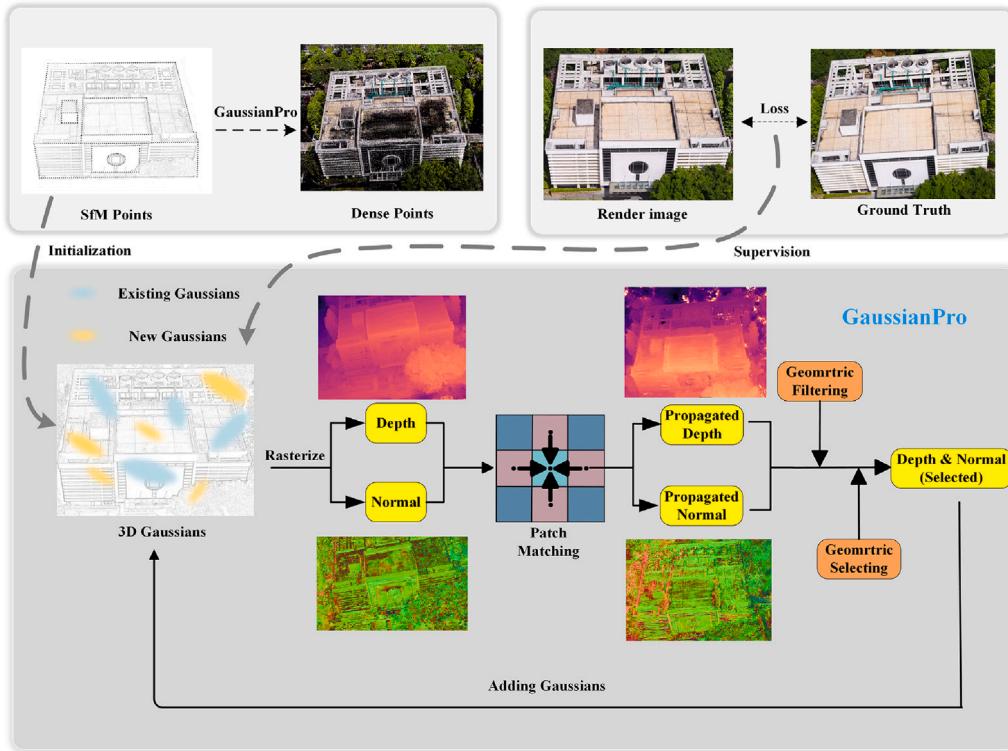


Fig. 4. The working principle of GaussianPro.

converted to a 3D plane( $d, \mathbf{n}$ ), where  $\mathbf{n}$  is the normal vector rendered for that pixel, and  $d$  is the distance from the camera coordinate origin to the plane. The calculation formula is as follows:

$$d = z\mathbf{n}^T \mathbf{K}^{-1} \tilde{p}, \quad (3)$$

where  $z$  is the rendering depth value of the pixel,  $\mathbf{K}$  is the camera intrinsic, and  $\tilde{p}$  is the homogeneous coordinate of the pixel.

After defining the plane, the checkerboard pattern defined in ACMH (Xu & Tao, 2019) is used to select adjacent pixels for conversion to a 3D plane as a candidate plane. The optimal plane is determined by path matching, and for pixel  $p$ , it is converted to  $p'$  using a homography transformation  $\mathbf{H}$ :

$$\tilde{p}' \simeq \mathbf{H}\tilde{p}, \quad (4)$$

$$\mathbf{H} = \mathbf{K} \left( \mathbf{W}_r - \frac{\mathbf{t}_r \mathbf{n}_k^T}{d_k} \right) \mathbf{K}^{-1}. \quad (5)$$

Where  $\mathbf{W}_r$  and  $\mathbf{t}_r$  are the relative transition matrices from the reference perspective to the adjacent perspective. Finally, Normalized Cross Correlation (NCC) (Mildenhall et al., 2021) is employed to verify the color consistency of  $p$  and  $p'$ , and the plane with the best color consistency is selected as the optimal plane. Multi-view geometric consistency check (Mildenhall et al., 2021) is utilized for geometric filtering to remove inaccurate depths and normals. The absolute value of the relative error between the filtered depth and the original depth is calculated. By setting a threshold, it is determined whether the Gaussians in a region are correctly generated. Finally, geometric selection is performed, and incorrect pixels are reprojected back into 3D space. New Gaussians are then initialized using 3DGS method.

After adaptive density control, it is essential to compare the rendered image with the real one and minimize the error between them to supervise the optimization of the Gaussian splatting point. First, the screen is divided into  $16 \times 16$  blocks, and the relative depth of each Gaussian point is calculated. Based on the calculation results, a single fast GPU Radix sort (Merrill & Grimshaw, 2010) is used to sort

the Gaussian points from near to far. Finally,  $\alpha$ -blending (Edelsbrunner et al., 1983; Kopanas, Philip, Leimkühler, & Drettakis, 2021; Mildenhall et al., 2021) is used to get the color of the pixels in the image. The camera is connected to the pixels in the image to form a ray. The color and opacity of the Gaussian points intersecting with the ray are used to calculate the color of the pixel through Eq. (6), resulting in the rendered image.

$$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} 1 - \alpha_j \quad (6)$$

where  $N$  is the number of Gaussian points,  $c_i$  is the color of the Gaussian point,  $\alpha_i$  and  $\alpha_j$  is the opacity of the Gaussian point.

For parameter optimization, the rendered image is compared with the original image. The  $L_1$  and D-SSIM losses are calculated. Additionally, the consistency between normal and propagated normal, the angular loss  $L_{normal}$ , and the scale regularization loss  $L_{scale}$  of NeuSG (Chen, Li, & Lee, 2023) are introduced:

$$L_{normal} = \sum_{p \in Q} \left\| \hat{N}(p) - \bar{N}(p) \right\|_1 + \left\| 1 - \hat{N}(p)^T \bar{N}(p) \right\|_1, \quad (7)$$

$$L = (1 - \lambda)L_1 + \lambda L_{D-SSIM} + L_{planar}, \quad (8)$$

$$L_{D-SSIM}(I_1, I_2) = 1 - SSIM(I_1, I_2), \quad (9)$$

$$L_{planar} = \beta L_{normal} + \gamma L_{scale}. \quad (10)$$

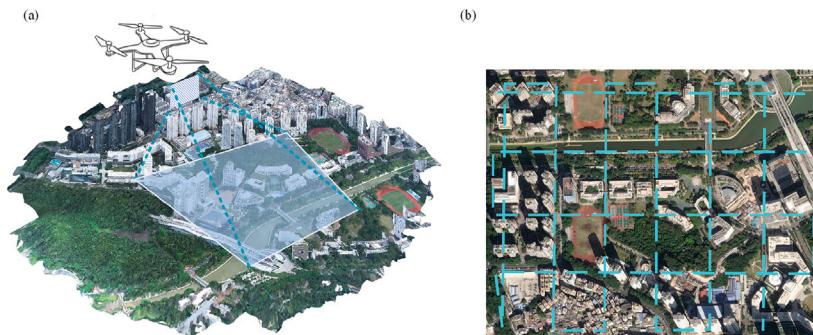
Where  $\hat{N}$  is the normal for rendering and  $\bar{N}$  is the normal for propagation.  $I_1$  and  $I_2$  are the real image and the rendered image respectively. The position, color, covariance matrix, spherical harmonic coefficient, and opacity of the point are optimized by the loss function.

## 2.2.2. Implementation

The input for the 3D Gaussian Splatting model consists of a set of overlapping images captured by the UAV. The tallest building in the study area is 120 m high (Fig. 5). The UAV's flight path is shown in



**Fig. 5.** Study area.



**Fig. 6.** UAV flight path.

**Fig. 6.** Using a well-patterned flight path for data acquisition improves efficiency and ensures data quality, resulting in 1560 images. Reducing the image resolution to 1k using cubic spline interpolation significantly reduces computing resource consumption with minimal impact on point cloud accuracy.

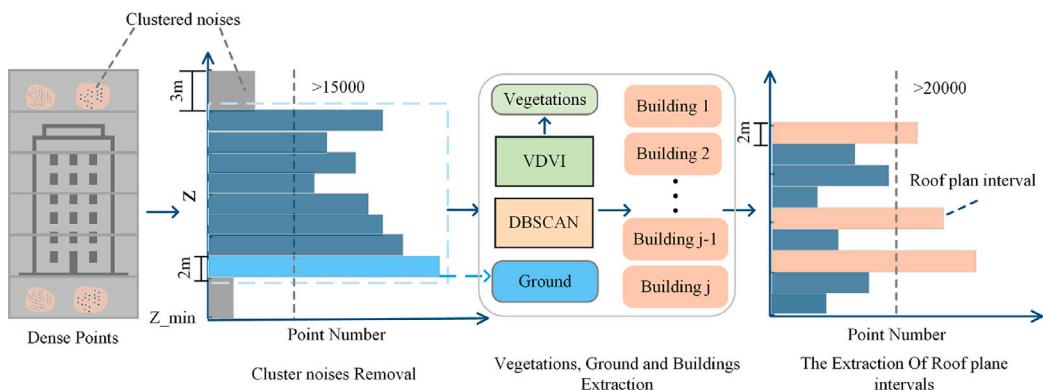
We used an Nvidia RTX4090 24 GB graphics processing unit (GPU) to test our method. To generate the required building point cloud, we set the total number of training iterations to 12,000 and the encryption interval to every 150 operations. For other parameters, refer to GaussianPro.

### 2.3. Model generation

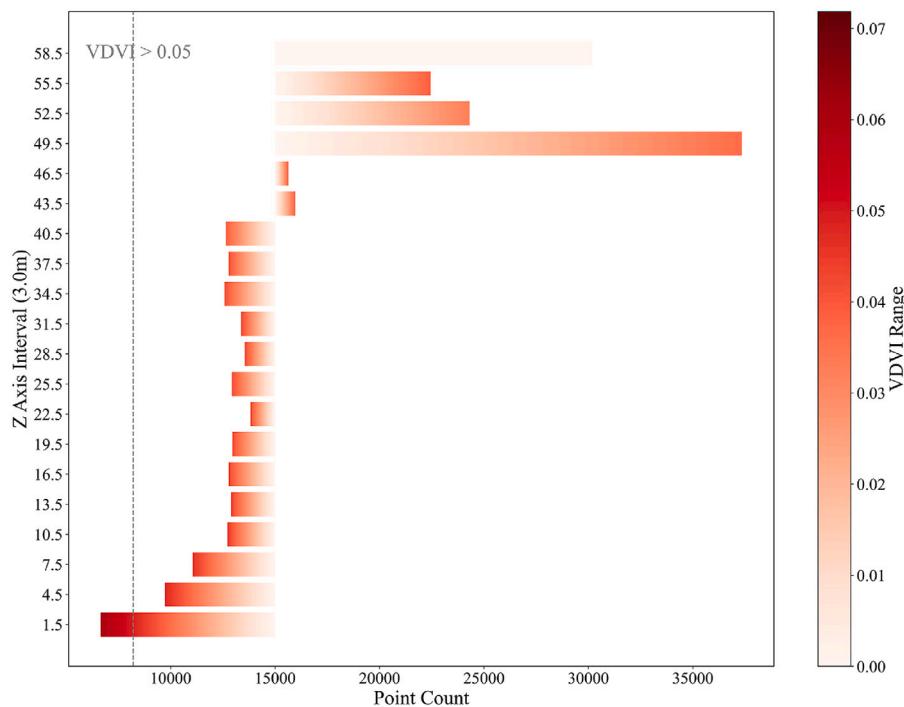
#### 2.3.1. Building points extraction

After obtaining dense point clouds, noise and non-building point clouds (such as vegetation and terrain) are usually present in the scene, significantly affecting subsequent model reconstruction. Due to the adaptive density control of 3DGs, the number, location, and other parameters of Gaussian points within the scene are optimized during

training. As a result, the generated scene point cloud has virtually no hard-to-remove noise points inside the scene or around the buildings, with noise points only appearing in clusters outside the scene. This paper proposes a height interval statistical algorithm (HISA) to remove noise and non-building point clouds. Specifically, the maximum value  $z_{\max}$  and minimum value  $z_{\min}$  of the point cloud on the  $z$ -axis are obtained first. The interval is then segmented, with the segment size set to 3 m for noise removal. The number of points in each interval is counted (Fig. 7). A threshold of 15 000 is set to remove clustered noise. The number of point clouds in the main area will be substantial, while the number of points in the height intervals where extreme error points are located will be very small. Therefore, extreme error points can be easily removed by setting a threshold. Subsequently, by calculating the modified VDVI index at each point, the vegetation point clouds are separated (Fig. 8). The leftover point cloud is subjected to further segmentation (where the interval is configured to 2 m). Subsequently, the quantity of points in each interval is recalculated. Finally, the points in the final interval are chosen and retrieved as the ground (Fig. 7).



**Fig. 7.** The workflow of height interval statistical algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 8.** Distribution and VDVI index of points along the Z-axis (vertical).

For vegetation point cloud extraction, 3DGS generates blue and red anomaly points during the training process. Consequently, while extracting green vegetation point clouds, these anomaly points should also be removed. This paper addresses this issue by improving the visible-band difference vegetation index (VDVI) (Wang et al., 2015), with the following formula:

$$VDVI_{green} = \begin{cases} 0 \\ \frac{2G-(R+B)}{2G+R+B} \end{cases}, \quad (11)$$

$$VDVI_{blue} = \begin{cases} 0 \\ \frac{2B-(R+G)}{2B+R+G} \end{cases}, \quad (12)$$

$$VDVI_{red} = \begin{cases} 0 \\ \frac{2R-(G+B)}{2R+G+B} \end{cases}. \quad (13)$$

Where  $R$ ,  $G$ , and  $B$  are the values of the three channels in the color respectively.

As shown in Fig. 8, points with a VDVI index exceeding the threshold value (set as 0.05 in this paper) are classified as vegetation. Given

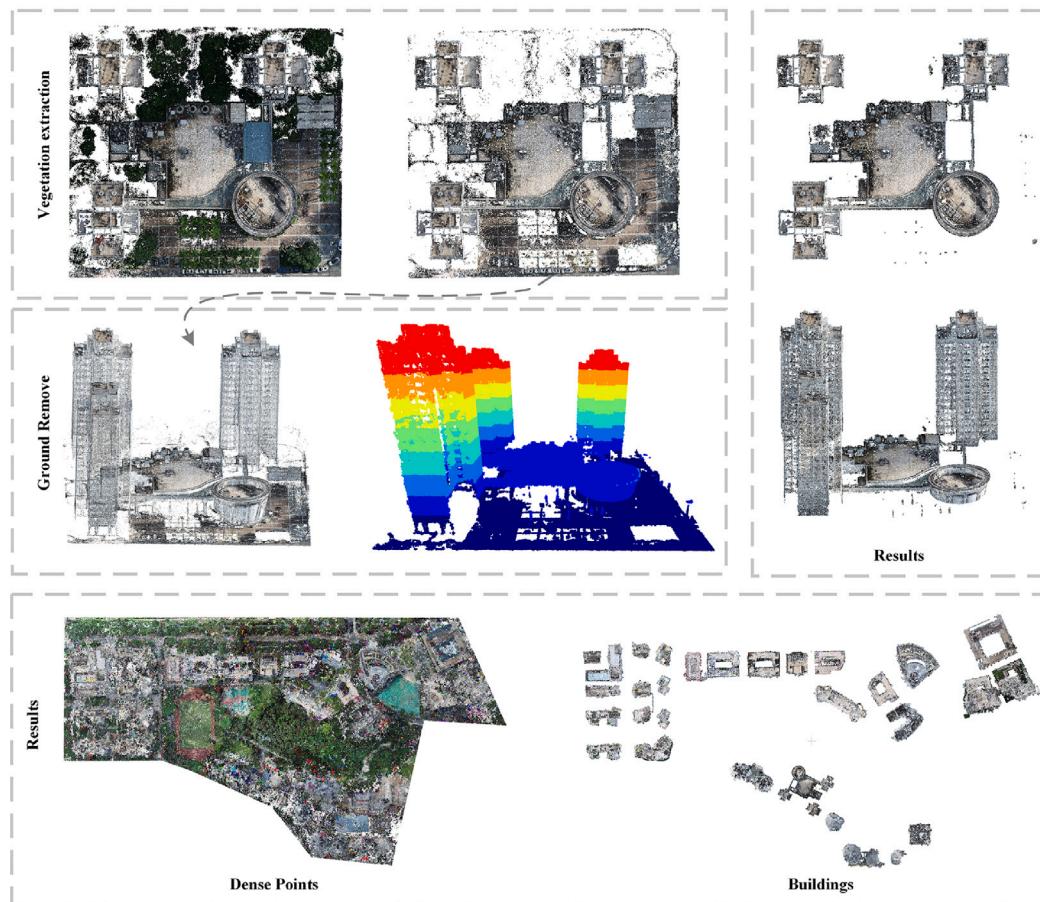
that vegetation height is generally low, the vegetation point cloud falls within the height range with low  $Z$ -axis coordinate values.

Simultaneously, terrain points are located in the interval with the smallest  $Z$  coordinate value. HISA can be applied again after processing the vegetation point cloud to obtain a highly stratified result without vegetation points (Fig. 9). The bottom layer of the result represents the terrain points.

The results of vegetation and terrain point cloud extraction are shown in Fig. 9. In the HISA results, the dark blue section of the layered point cloud represents terrain points. Subsequently, DBSCAN (Ester et al., 1996) was used to quickly cluster and extract single building points for geometric model establishment.

### 2.3.2. Roof contour extraction

To generate regular geometric models through point clouds, it is necessary to extract and refine the building contours. Firstly, HISA filters out the roof plane intervals with a new point cloud quantity threshold (20 000). The segment interval size is set to 1.5 m to extract all intervals containing roof planes (Fig. 7). Then, RANSAC (Fischler & Bolles, 1981) is used to extract all candidate building plane point clouds



**Fig. 9.** Vegetation, terrain and building points extraction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

within these intervals. At last, DBSCAN (Ester et al., 1996) combined with point density is used to identify the real planes. Since the point density of real plane clusters is significantly higher than that of non-plane clusters, the combination of density and DBSCAN can effectively identify and remove non-plane cluster points (Fig. 10).

Subsequently, morphological dilation and erosion are applied to extract roof planes, filling edge defects and internal holes, and restoring the edge shape of the plane. As shown in Fig. 11, each cluster in the plane is initially projected into a binary matrix. Next, a structuring element moves over the matrix. For dilation, when the structuring element intersects with the binary matrix, the value at its origin is retained. For erosion, the value at the origin is retained when the structuring element is completely contained within the object; otherwise, it is removed. Finally, the Canny operator (Canny, 1986) is used to extract the plane boundaries, with the longest boundary taken as the plane contour of building roof.

### 2.3.3. Contour refinement

The roof contour extracted by Canny operator is rough and requires a reduction in contour points and regularization. This algorithm framework initially employs the Ramer–Douglas–Peucker (RDP) (Douglas & Peucker, 1973) algorithm to minimize the number of contour points while preserving the basic shape of the original contour curve. As the number of contour points decreases, computational efficiency improves while avoiding the problem of uneven contours. For a polyline connecting multiple points, connect the start and end points to form a straight line, then calculate the distance  $d$  from each remaining point to the straight line, and select the maximum distance  $d_{\max}$  (Fig. 12(a)). If  $d_{\max}$  is less than the threshold, the line connecting the start and end points approximates the polyline. Conversely, when  $d_{\max}$  exceeds

the threshold, connect the point to both the start and end points (Fig. 12(b)). Then, repeat the above steps to simplify until the final contour is achieved (Fig. 12(d)).

After simplification, the contour is refined using the split, merging, and intersection simplification rules proposed by Zhang, Yan, and Chen (2006). When the projection of a horizontal inclined segment on the  $x$ -axis is longer than its projection on the  $y$ -axis, and the  $y$ -axis projection is below the threshold, a split operation is performed (Fig. 13(a)). Fig. 13(b) shows the intersection operation, which restores the right angles lost due to RDP simplification. For cases where the distance between two parallel line segments is below the threshold, replace them with a broken line segment (Fig. 13(c)).

The contour refinement results are shown in Fig. 10. The optimized planar contour eliminates sharp corners and short edges that affect the quality of the subsequent CFD mesh. The resulting building outline is then used to generate a model in Rhinoceros. Specifically, the contour points are classified by plane number, and contour points belonging to the same building plane are constructed as closed polygons. Finally, surfaces are generated through the polygon boundaries and extruded to the corresponding height to form prismatic entities.

## 3. Results and discussion

### 3.1. Dense point cloud

This study is based on the GaussianPro model, which has been improved to quickly generate accurate dense points. The dense point cloud of the study area is shown in Fig. 9. Fig. 14 shows the training results and dense points of selected buildings. From left to right, the images show the initial sparse point cloud, the rendering scene of

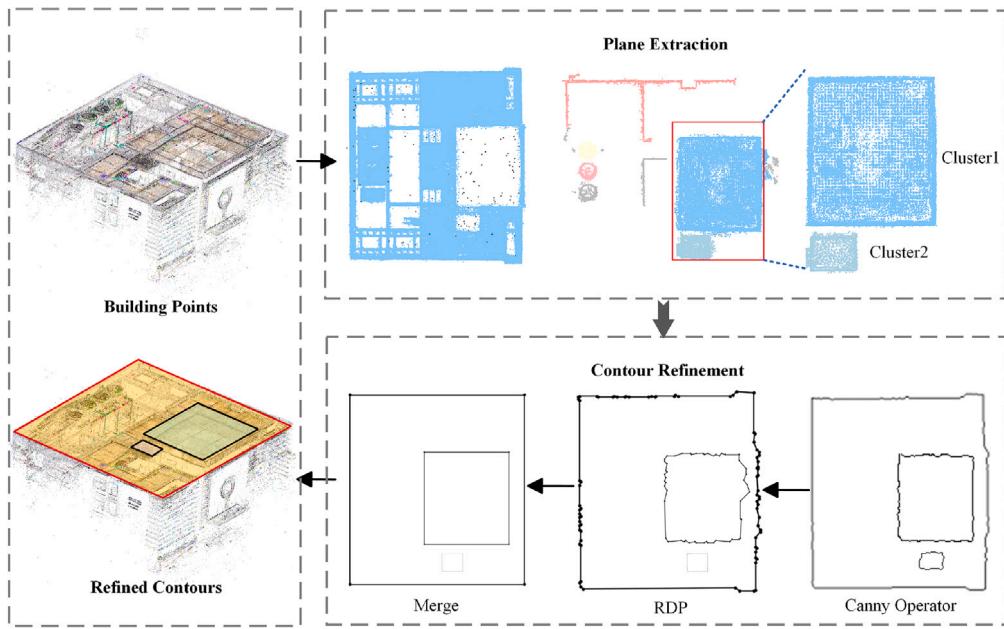


Fig. 10. Building contour extraction and refinement.

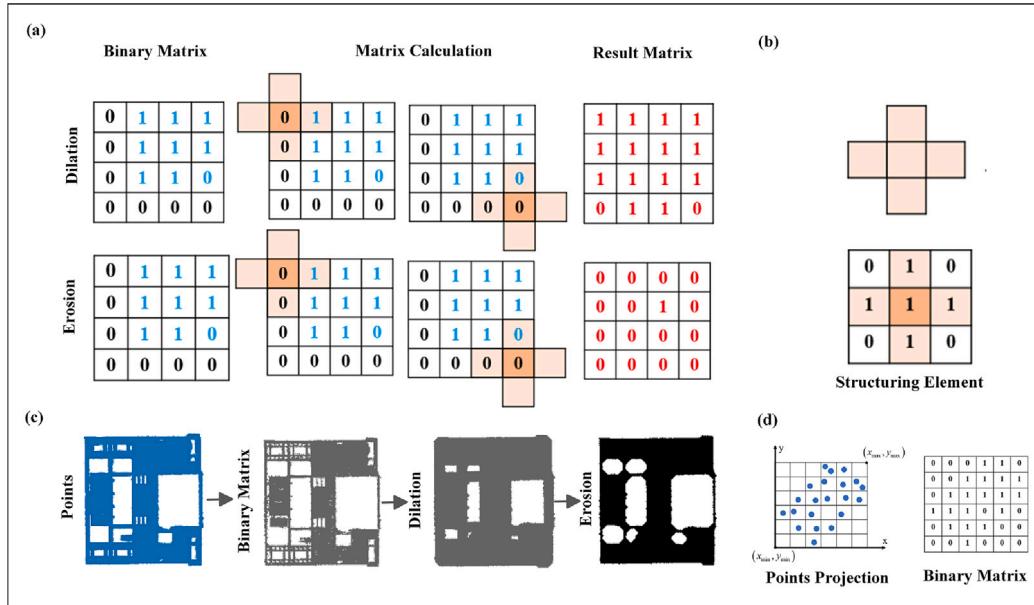


Fig. 11. Dilatation and erosion.

the Gaussian ellipsoid, the Gaussian ellipsoid in the scene, and the final dense point cloud. It can be seen that the scene is composed of numerous Gaussian ellipsoids with their centers of mass corresponding to the positions of the points in the dense point cloud. Most of the Gaussian ellipsoids in the scene are correctly positioned, and the size differences are minimal. This indicates that during the training process, the 3DGS model captured sufficient building details to produce higher quality dense point clouds.

We used the evaluation indicators suggested by MVSNet (Yao et al., 2018) to assess the quality of the generated point cloud in terms of accuracy and completeness. The evaluation indexes included Accuracy (Acc), and completeness (Comp) for both the distance metric (Aanæs, Jensen, Vogiatzis, Tola, & Dahl, 2016) (lower is better) and the percentage metric. Additionally, Precision (Pre), Recall (Rec), and f-score (Knapsch, Park, Zhou, & Koltun, 2017) (higher is better)

were considered. Due to limited computing resources, only five representative buildings (B1, B2, B3, B4, B5) from Fig. 14 were selected for evaluation. The results are presented in Table 1.

The results indicate that the dense point cloud generated by the 3DGS method has high accuracy (Acc), but relatively low completeness (Comp) and recall (Rec) indices. This is attributed to the 3DGS method's use of large Gaussian ellipsoids to cover local areas with insignificant color changes (Cheng et al., 2024), resulting in a loss of point cloud details. Due to vegetation obstruction, B2 lacks details on the side of the building, leading to poorer completeness performance (Kim, Lee, & Lee, 2024). However, these issues only occur in the facades near the bottom areas and local regions with dark colors, having minimal impact on subsequent geometric model construction. B3, with less surrounding vegetation, exhibits higher completeness (Comp) and recall (Rec) indices compared to other buildings. The f-scores of B1 and B4 are lower

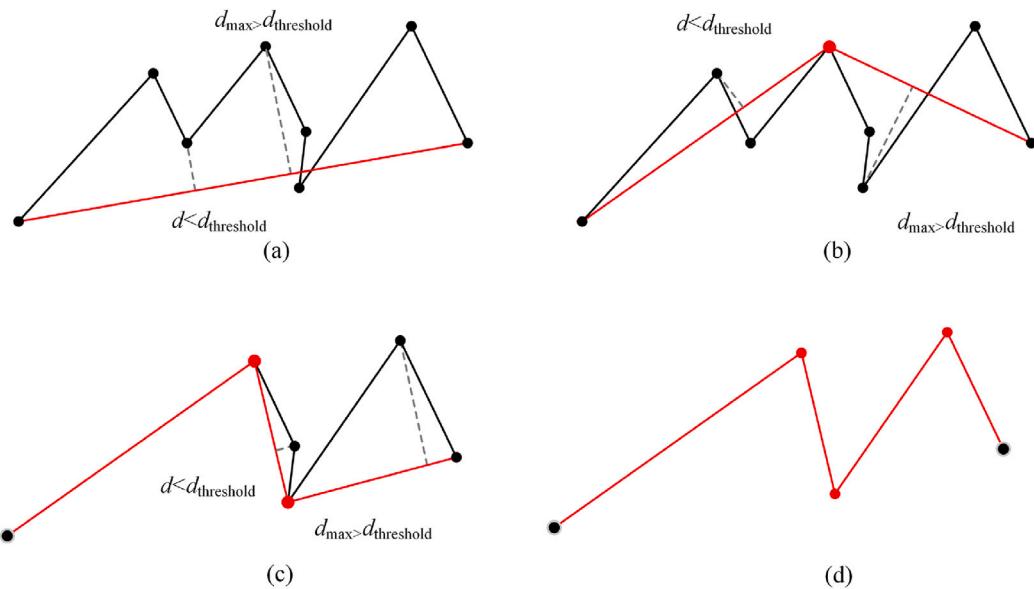


Fig. 12. The flowchart of RDP.

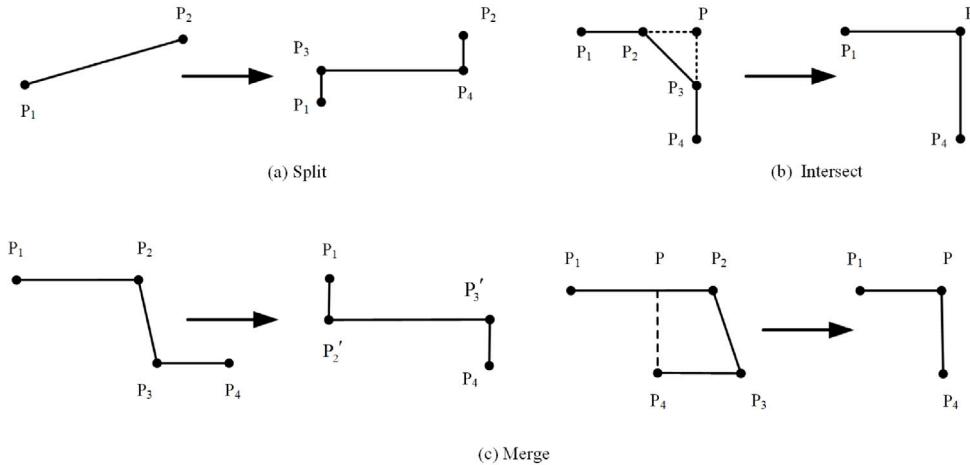


Fig. 13. Split, intersect and merge.

**Table 1**  
The quality evaluation results of selected buildings.

|    | Acc<br>(m) | Comp<br>(m) | Pre<br>(<0.2 m) | Rec<br>(<0.2 m) | f-score<br>(<0.2 m) | Pre<br>(<0.5 m) | Rec<br>(<0.5 m) | f-score<br>(<0.5 m) |
|----|------------|-------------|-----------------|-----------------|---------------------|-----------------|-----------------|---------------------|
| B1 | 0.2251     | 0.3701      | 64.25           | 32.16           | 42.87               | 94.61           | 73.12           | 82.49               |
| B2 | 0.2067     | 0.8105      | 97.08           | 57.43           | 70.14               | 98.66           | 84.31           | 90.93               |
| B3 | 0.2071     | 0.2057      | 89.82           | 61.20           | 72.79               | 98.83           | 90.20           | 94.32               |
| B4 | 0.2168     | 0.3960      | 65.24           | 35.01           | 45.57               | 89.86           | 77.98           | 83.50               |
| B5 | 0.2648     | 0.3755      | 76.56           | 61.25           | 68.05               | 90.63           | 81.70           | 85.93               |

compared to other buildings because their roof structures are black, which are similar to the background color set by the 3DGS model, resulting in poor training effects and lower point cloud quality in these areas. Despite these issues causing a loss of local building details, the overall point cloud quality still meets the requirements for building geometric reconstruction.

We also compared the accuracy and efficiency of our method with other methods using the same dataset. The results are presented in Table 2, and Fig. 15 illustrates the differences with other methods more clearly. Although the 3DGS method is slightly lower in completeness compared to MVS methods, it has higher accuracy than MVS and other methods like NeRF. Compared to COLMAP, the highest quality method among the others, our proposed algorithm framework improves

accuracy by an average of 12% and generates results much faster. It is worth noting that although Context Capture (CC) is slightly faster than other methods, it is still 2–3 times slower than our method, and the quality of the point cloud generated from downsampled images is far inferior to the results of other methods and 3DGS.

Dataset quality directly affects the precision of point clouds and geometric models. Factors such as weather and sunlight during UAV data collection significantly impact the quality of the image dataset (Rosnell, Honkavaara, & Nurminen, 2012). To address weather impacts, This study tested two weather conditions: sunny and sufficiently bright overcast. On sunny days, strong light can cause overexposure (Fig. 16(b)), failing to capture building details. Increased brightness can make white parts of buildings appear uniform, worsening detail loss during 3DGS

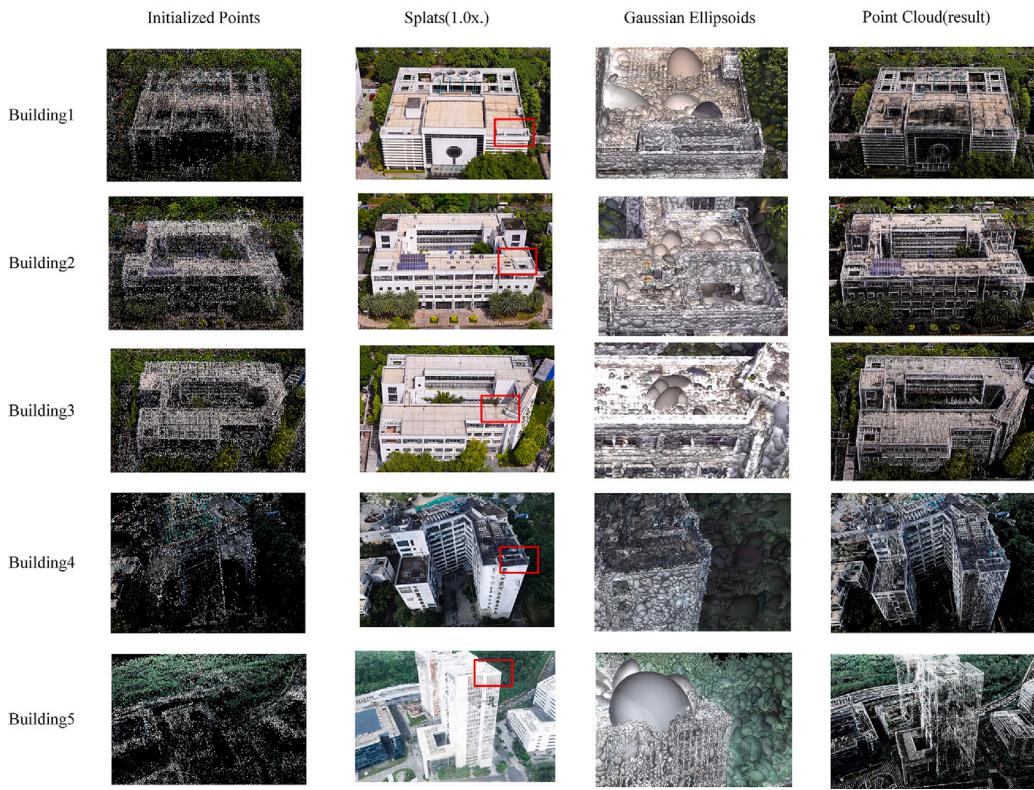


Fig. 14. Dense point cloud of selected buildings.

**Table 2**  
Comparison with other methods in Accuracy and speed.

| Method                          | B1         |             | B2         |             | B3         |             | B4         |             | B5         |             |
|---------------------------------|------------|-------------|------------|-------------|------------|-------------|------------|-------------|------------|-------------|
|                                 | Acc<br>(m) | Time<br>(h) |
| Ours                            | 0.2251     | 0.28        | 0.2067     | 0.25        | 0.2071     | 0.28        | 0.2168     | 0.20        | 0.2648     | 0.30        |
| COLMAP (Han et al., 2015)       | 0.4632     | 1.50        | 0.3137     | 1.23        | 0.2238     | 1.43        | 0.1379     | 0.88        | 0.3507     | 1.23        |
| Context capture                 | –          | 0.72        | –          | 0.67        | –          | 0.77        | –          | 0.47        | –          | 0.70        |
| NeuS (Wang et al., 2021)        | 0.7257     | 6.50        | 0.6431     | 4.06        | 0.5981     | 5.58        | 0.7942     | 5.45        | 0.7641     | 5.65        |
| NeuDA (Fabbri & Costanzo, 2020) | 0.7043     | 4.38        | 0.5074     | 3.80        | 0.6550     | 5.90        | 0.7825     | 3.18        | 0.8543     | 5.03        |

training. Therefore, exposure of images needs to be reduced, typically set to  $-0.3$ , but not too low to avoid overly dark images which hinder point cloud generation. Shooting is usually done during solar noon. In the afternoon and evening, the sun's angle creates large shadows on buildings, and reduced light intensity leads to color uniformity, further darkening originally dim objects (especially vegetation) (Fig. 17(a)), resulting in large detail loss in dense point clouds during 3DGS training (Fig. 17(b)). Overcast but sufficiently bright conditions are suitable for UAV shooting, providing moderate and consistent light without building shadows, ensuring uniform dataset quality.

In addition, the overlap ratio of the image data also affects the efficiency and accuracy of the research framework in this study. If the ratio is too high, it will increase the generation time of the point cloud. If it is too low, sufficient scene details cannot be captured. Usually, setting it to 75% is appropriate.

### 3.2. Building geometry model

The results of the geometric models are shown in Fig. 18. From left to right, (a) (e) show the building point cloud, planar contour, and geometric model, respectively. (i) shows the overall scene result, with the green part indicating vegetation. The geometric model obtained by the algorithm in this paper has regular planar contours and retains roof plane details well, without sharp angles and unevenness that could

cause CFD calculations to diverge. And the levels of detail of the models reach LoD2 and LoD2.5. Since most building facades only contain elements such as windows and balconies, which will result in short planes and edges when generating the geometric model, affecting the quality of subsequent CFD mesh division. Additionally, the buildings shown in Fig. 18(a), (d), and (e) have significant issues with facade details in the point cloud due to vegetation obstruction and background color settings, while roof details are well preserved. Hence, the geometric model generation does not focus on building facade details, effectively avoiding geometric quality issues caused by facade details, while ensuring the regularity of the overall model and adding more roof plane details.

### 3.3. Analysis of computational resources

The computational resource requirements for this framework are primarily focused on CUDA Video Random-Access Memory (VRAM) and physical memory. Firstly, for dense point cloud generation, 3DGS generates and stores millions of Gaussians during training. Therefore, it is recommended to use VRAM of no less than 24 GB for large-scale reconstruction tasks (Wang et al., 2021). When using devices with lower VRAM, the number of Gaussian points must be reduced by adjusting parameters like the encryption interval and the total number of encrypted training instances.

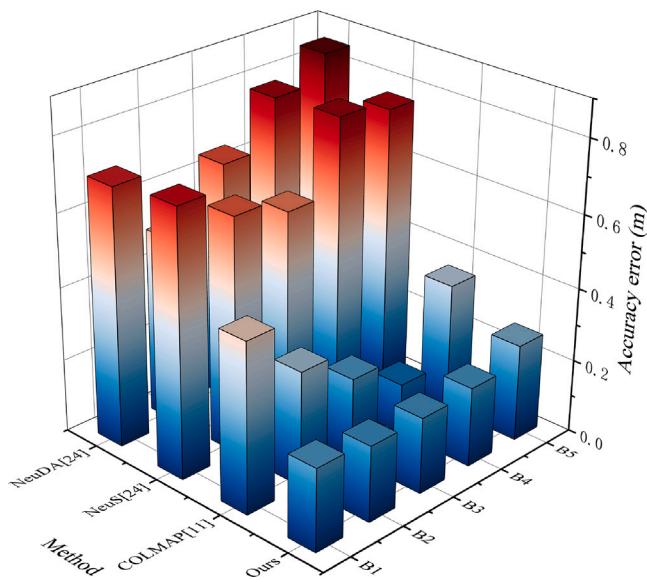


Fig. 15. Comparison of the accuracy of our algorithm and other methods on test data.

To verify this, the application was tested on different devices: RTX 4090 (24 GB), RTX 3090 (24 GB), Tesla T4 (16 GB), and RTX 3080ti (12 GB). The RTX 4090 and RTX 3090 successfully reconstructed the dense point cloud with the original parameters (Scheme A: 150 and 12 000). The Tesla T4 could reconstruct the dense point cloud after adjusting parameters (Scheme B: 500 and 10 000). The RTX 3080ti and RTX 2080ti could only reconstruct the dense point cloud of a single building, indicating insufficient VRAM for larger scenes. The results of points generated by each graphics card show significant differences between Scheme A and Scheme B as shown in Fig. 19. Despite the limitations, Scheme B can still generate a relatively complete dense point cloud of buildings in small-scale scenes or single buildings. In the field of geometric model generation, since this study does not use machine learning or deep learning models for generating geometric models, there are no high demands for VRAM and physical memory.

As for subsequent CFD simulations and related applications, the total number of grids in the subject of this study is 18 million, requiring at least 128 GB of physical memory for CFD simulation. For the fine grids used in grid convergence index (GCI) calculations, the number of grids reaches 37.9 million, necessitating 256 GB of physical memory to conduct CFD simulations and applications.

### 3.4. Remaining limits

Both traditional MVS methods and the 3DGS method used in this research have noticeable shortcomings in larger and more intricate environments. As the reconstruction scene grows, drones often need multiple flights to gather image data, resulting in inconsistent data quality and reduced precision of the generated point cloud. Additionally, multiple drone flights increase time costs. Secondly, both methods require feature extraction, feature matching, and aerial triangulation algorithms to generate a sparse point cloud, which is then used for dense reconstruction. As the scene size increases, the number of image datasets captured by drones also increases, significantly prolonging the time needed to reconstruct the sparse SfM point cloud (Stathopoulou & Remondino, 2023). Generating a sparse point cloud from thousands of highly overlapping images can take several hours or even days.

Moreover, in complex shapes and densely built environments, the point clouds generated by the 3DGS and MVS methods fail to produce detailed building geometry. For example, the geometric model generation algorithm proposed in this study utilizes height point frequency

information to extract building planar details. However, for buildings with complex shapes (as shown in Fig. 20), due to the presence of many small planes that are difficult to extract solely through point frequency, it cannot generate building geometry that sufficiently expresses these details.

Finally, in scenarios where buildings are more densely packed, such as urban villages, the generated point clouds result in buildings being closely connected. This causes methods like DBSCAN (Ester et al., 1996), which rely on point density to extract different building point clouds, to become ineffective, often resulting in a single large geometric structure. This paper uses urban villages as an example to demonstrate the persisting deficiencies by generating the geometric model of the area, as shown in Fig. 21. From Fig. 21(a), it is evident that buildings in the urban village are almost interconnected, leading to closely connected structures in the dense point cloud. Consequently, during geometric generation, it becomes challenging to separate each building into individual entities, resulting in a large geometric structure, as shown in Fig. 21(b).

## 4. Application

### 4.1. CFD simulation

#### 4.1.1. Computational domains and grids

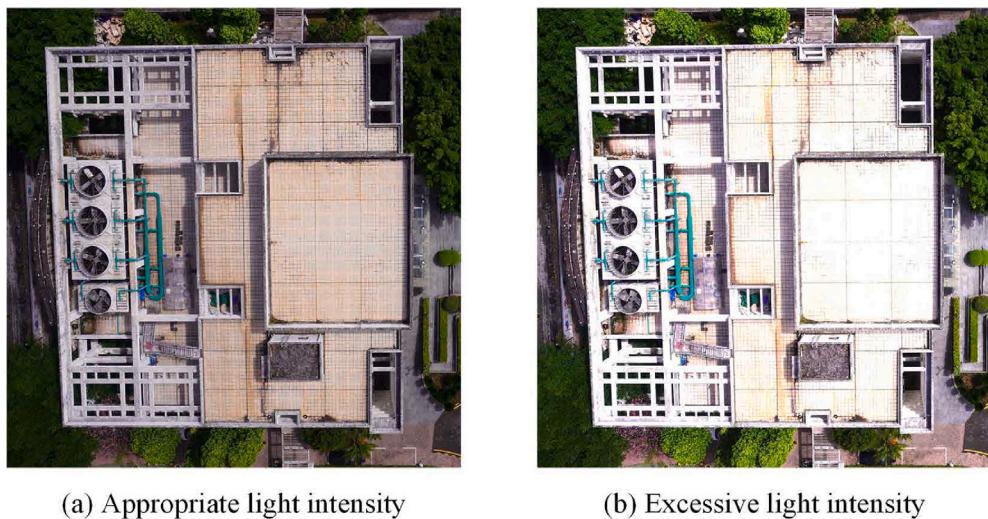
CFD fluid simulation is conducted to further verify the validity of the building geometry. We designed the domain scope based on best practice guidelines (BPGs) (Franke, Hellsten, Schluenzen, & Carissimo, 2011; Tominaga et al., 2008). Fig. 22 shows the dimensions of the computational domain, where the upwind direction is set to five times the height of the tallest building ( $H_{\max}$ ) in the study area within the densified zone. The sides are set to  $5 H_{\max}$  from the densified zone, the downwind direction is set to  $15 H_{\max}$  from the densified zone, and the height is set to  $6 H_{\max}$  (Tominaga et al., 2008). The densified zone is set as a cylinder, covering all buildings in the study area, with a diameter of  $20 H_{\max}$  and a height of  $2 H_{\max}$ . The inlet boundary of the computational domain is set to velocity-inlet, and using a user-defined function (UDF), the corresponding average wind speed and turbulence intensity of the C-type rough terrain are applied to the inlet boundary of the computational domain. The outlet boundary is set to pressure-outlet, while the sides and top are configured with symmetry boundary conditions. The ground and buildings are set with no-slip wall boundary conditions.

We generated three sets of meshes to verify mesh convergence, ensuring that the continuous refinement ratio of cell edge length ( $r$ ) is 1.3. A larger refinement ratio is prohibited as it would result in the number of fine mesh cells exceeding 40 million. The number of cells is 7.9 million for the coarse mesh, 16.8 million for the base mesh, and 37.9 million for the fine mesh. The mesh resolutions are 1.44 m for the coarse mesh, 1.2 m for the base mesh, and 1 m for the fine mesh. The base mesh is shown in Fig. 23.

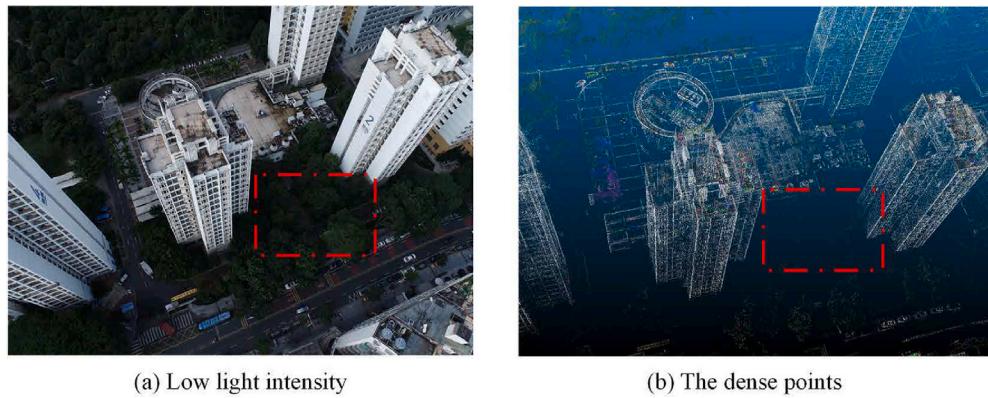
#### 4.1.2. Numerical simulation

The  $k-\omega$  SST RANS turbulence model is used for three-dimensional incompressible Newtonian fluid simulation. The Couple algorithm is utilized to solve the steady-state velocity and pressure, employing a second-order spatial method for the simulation. For each component, including mean wind speed ( $U$ ), pressure ( $p$ ), turbulent kinetic energy ( $k$ ), turbulent kinetic energy dissipation rate ( $\epsilon$ ), and specific dissipation rate ( $\omega$ ), a value of  $1 \times 10^{-6}$  is used to evaluate the convergence of the solution.

There are two methods to consider the influence of vegetation on the flow field in numerical simulation. One method is to modify the wall function to account for the drag effect of roughness, but this is only applicable to smooth or low roughness terrain (Xie, Voke, Hayden, & Robins, 2004). This method is simpler but less accurate in complex terrains. The other method is the Canopy Fluid Model, which



**Fig. 16.** The comparison between appropriate and excessively high light intensities. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 17.** The performance of reconstruction with different light intensities.

simulates the influence of vegetation on the flow field by adding the drag source term to the N-S equation in the space occupied by the canopy. Specifically, cells in the corresponding area were identified using vegetation contours before CFD simulations. Then, source terms including momentum,  $k$ , and  $\epsilon$  are added to these cells to simulate the effect of vegetation. The formula of vegetation resistance source term is as follows:

$$S_u = -C_d a |\mathbf{u}| U, \quad (14)$$

$$S_k = C_d a (\beta_p |\mathbf{u}|^3 - \beta_d |\mathbf{u}| k), \quad (15)$$

$$S_\epsilon = C_d a (c_\epsilon \beta_p |\mathbf{u}|^3 - c_\epsilon \beta_d |\mathbf{u}| \epsilon). \quad (16)$$

Where  $S_u$  refers to the loss of wind speed due to viscous drag forces induced by roughness canopy,  $S_k$  and  $S_\epsilon$  represent the balance of turbulence generation and dissipation caused by roughness canopy,  $C_d$  is the drag coefficient depends on the surface occluder's roughness type,  $a$  is the leaf area density,  $\mathbf{u}$  is the fluid component vector representing the downstream, expansion, and vertical velocity components, and  $U$  is the average downstream velocity.  $k$  and  $\epsilon$  are turbulence kinetic energy and its dissipation rate.  $\beta_p$ ,  $\beta_d$  and  $c_\epsilon$  are model constants (Amorim, Rodrigues, Borrego, & Costa, 2010). In this study, all CFD simulations were conducted on a rack-mounted high-performance computer equipped with AMD EPYC 7573X processors (a total of 384 cores).

Fig. 24 shows the canopy prism of vegetation points. Once the spatial distribution and coefficients of vegetation have been determined,

the effect of vegetation on the wind field can be incorporated into the simulation process using user-defined function (UDF).

Figs. 25 and 26 shows the wind speed gradient and turbulent kinetic energy results at 2 m in the study area, demonstrating that all models yield stable and continuous outcomes. Notably, wind speed varies significantly around the upwind edges of certain buildings, with noticeable hotspots in narrow streets or alleys between buildings. This occurs because the wind accelerates when flowing through narrow spaces, creating a “wind tunnel effect”, which increases the wind speed gradient and may affect pedestrian comfort in these areas. In open areas without many buildings obstructing, the wind speed changes more gradually, resulting in smaller wind speed gradients and turbulent kinetic energy.

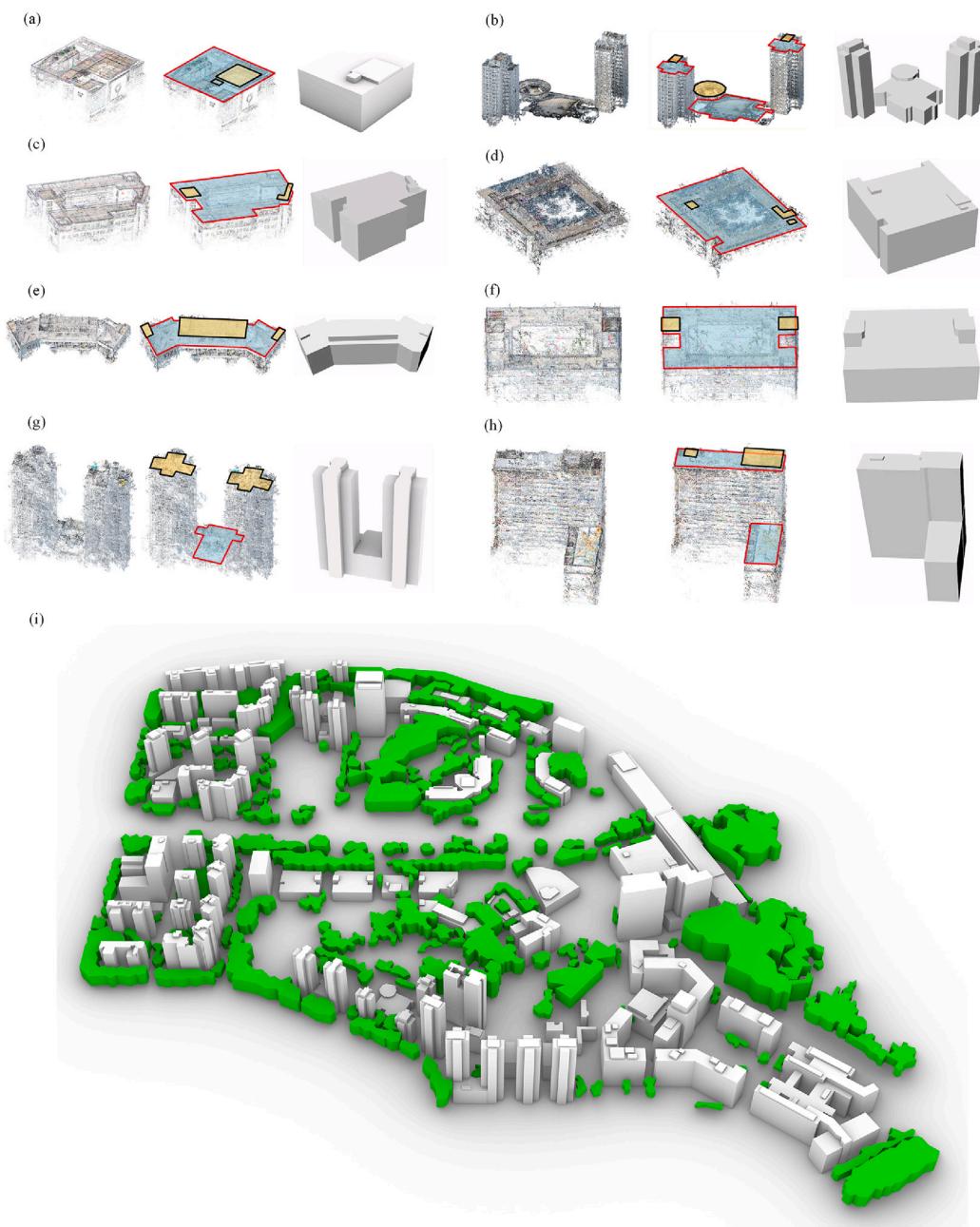
#### 4.1.3. Mesh convergence analysis

To demonstrate mesh independence and validate geometric topology, this study conducted a Richardson extrapolation-based convergence analysis using the grid convergence index (GCI) (Roache, 1997). 32 random monitoring points were arranged, and the GCI for their respective velocity, pressure, turbulence intensity, and turbulence dissipation rate were calculated. The GCI calculation formula is:

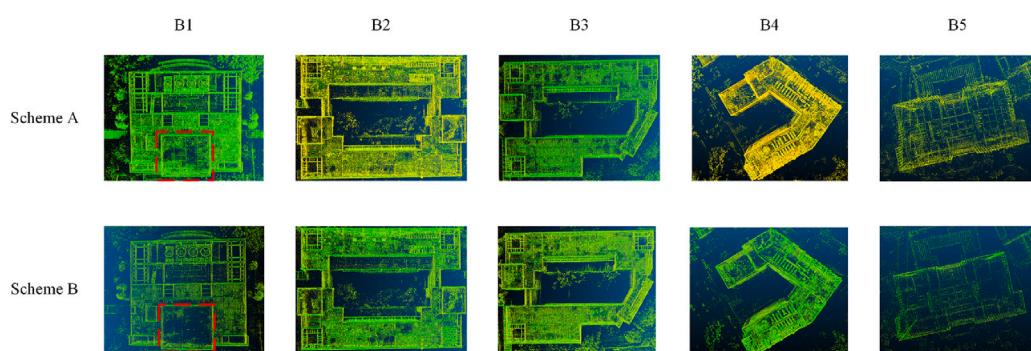
$$G_F = (100 \times F_s \times \epsilon_{FB}) / (r^p - 1), \quad (17)$$

$$G_B = (100 \times F_s \times \epsilon_{BC}) / (r^p - 1), \quad (18)$$

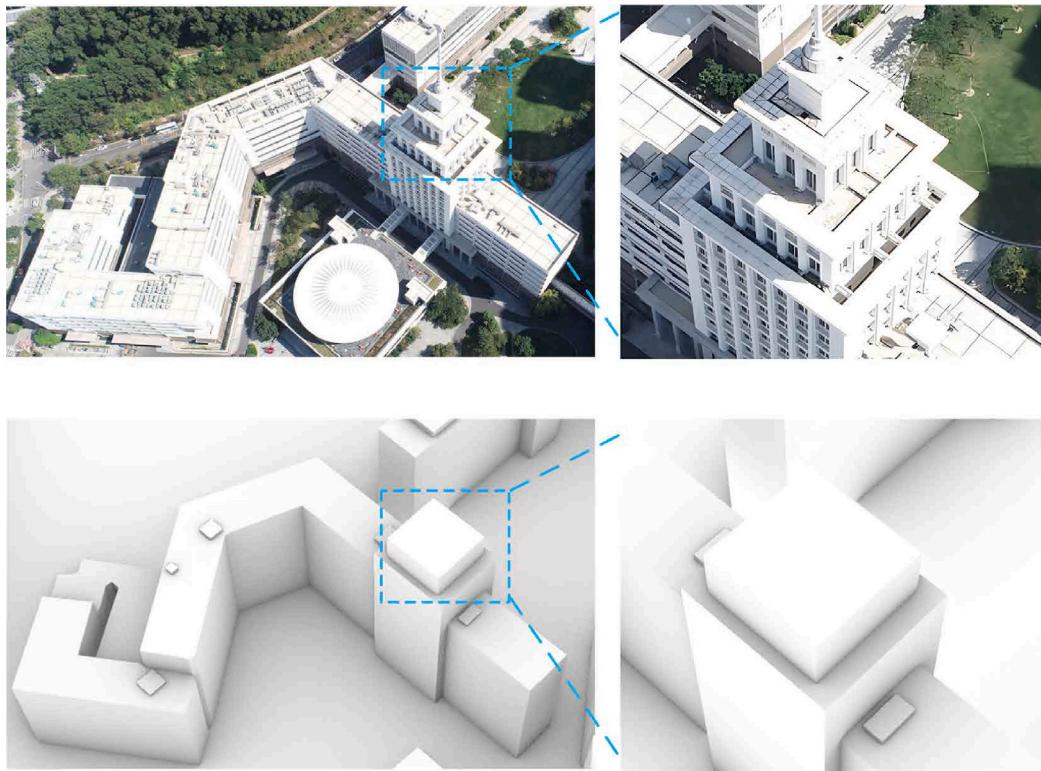
$$G_C = r^p \times G_B. \quad (19)$$



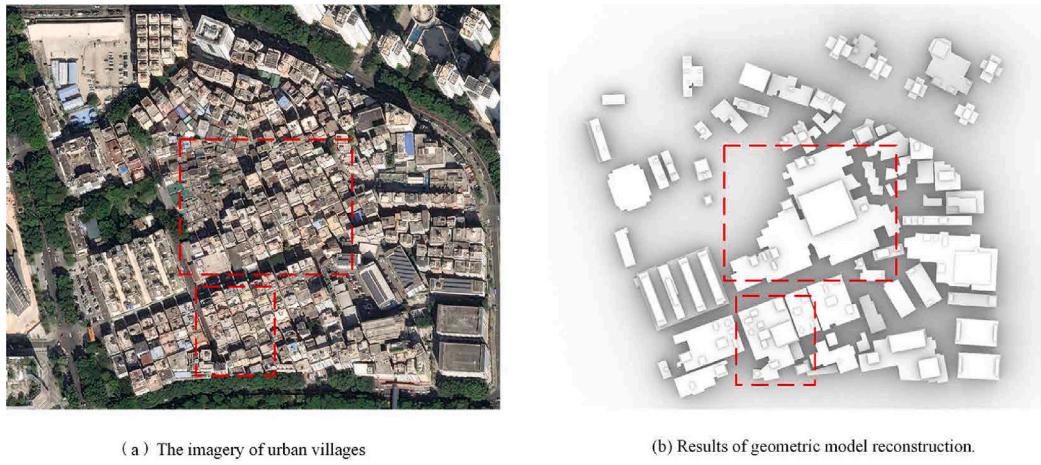
**Fig. 18.** The results of the geometric model reconstruction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 19.** Comparison of point clouds obtained from Scheme A and Scheme B.



**Fig. 20.** The reconstruction performance complex architectural.



**Fig. 21.** The reconstruction performance of buildings with higher architectural complexity.

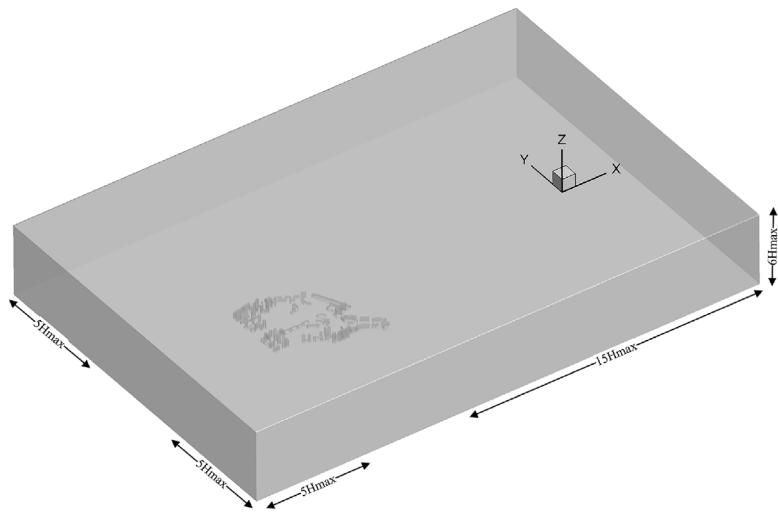
Where  $F_s (=1.25)$  is the safety factor,  $\varepsilon_{ij} = \left| \frac{f_j - f_i}{f_i} \right|$ , i and j represent the mesh (F: Fine, B: Base, C: Coarse), f is the variable of the selected monitoring point.  $p = \ln \left[ \frac{f_C - f_B}{f_B - f_F} \right] / \ln r$  is the observed order of accuracy of the algorithm. The grid settings and GCI results are shown in [Table 3](#), where the maximum values for the velocity and pressure fields are 3.76%, and the maximum value for turbulence quantities is 4.89%, indicating the reliability of the simulation results.

In addition to calculating the GCI index, this study further selects the optimal grid by comparing the wind speed ratio distribution at various monitoring points across three grid schemes. [Fig. 27](#) illustrates the scatter plot of wind speed ratios at different monitoring points. It is evident that the simulation results of the coarse grid and the base grid have significant errors, with many points exceeding 10%. In

**Table 3**  
Number of grids and GCI value.

| Mesh   | Cell(M) | GCI (%)          |      |      |               |
|--------|---------|------------------|------|------|---------------|
|        |         | $U_{\text{mag}}$ | $p$  | $k$  | $\varepsilon$ |
| Coarse | 7.90    | 5.20             | 5.57 | 7.24 | 6.96          |
| Base   | 16.8    | 3.51             | 3.76 | 4.89 | 4.70          |
| Fine   | 37.9    | 2.46             | 2.24 | 4.31 | 3.30          |

contrast, the error between the fine grid and the coarse grid is relatively smaller. The base grid can be considered to have a certain level of grid independence. Taking into account computational accuracy and cost, the base grid is selected for other working conditions and subsequent pedestrian comfort evaluations.



**Fig. 22.** Computational domain model.

#### 4.2. Analysis of pedestrian comfort levels

The above CFD results indicate that the framework can generate high-detail building geometry. To further verify the framework's practicality, this study uses the generated building geometry to conduct a pedestrian comfort analysis for the selected research object.

##### 4.2.1. Evaluation guidelines

Evaluating the quality of pedestrian wind environments requires two types of data: the aerodynamic characteristics of the site and the long-term wind speed and direction data from meteorological observation stations near the study area, which are used to determine the probability distribution function of long-term wind speeds. To determine the statistical characteristics of the wind, the Weibull distribution, a widely recognized normal wind distribution probability model (Holmes, Paton, & Kerwin, 2007), is used to analyze the meteorological observation wind speed data. Based on the wind's statistical characteristics, this study uses the Peak Over Threshold (POT) method to evaluate pedestrian comfort categories (Willemse & Wisse, 2007) and calculates the exceedance probability using the average wind speed (M, 2012; Ministry of Housing and Urban-Rural Development of the People's Republic of China, 2014).

$$P_\theta(\bar{V}_{\text{ped}} > V_{\text{THR}}) = A_\theta \cdot \exp \left[ - \left( \frac{V_{\text{THR}} - \mu_\theta}{c_\theta} \right)^{k_\theta} \right] \quad (20)$$

Among them,  $\bar{V}_{\text{ped}}$  represents the average wind speed at pedestrian height,  $V_{\text{THR}}$  denotes the threshold wind speed for discomfort or danger,  $\theta$  is the location parameter,  $P_\theta$  signifies the cumulative probability of wind speed exceeding  $V_{\text{THR}}$ ,  $A_\theta$  represents the frequency of the wind direction angle  $\theta$ ,  $c_\theta$  is the scale parameter of the probability distribution function, and  $k_\theta$  is the shape parameter.

This study classifies the comfort of the wind environment using Table 4. When evaluating with the daily average maximum wind speed data from meteorological stations, annual exceedance counts should be used. When using hourly wind speed, hourly exceedance probability should be applied (Ministry of Housing and Urban-Rural Development of the People's Republic of China, 2014).

##### 4.2.2. Meteorological data analysis

This study gathered wind direction and speed data from nearby meteorological stations and calculated the frequency of each wind direction. The wind direction frequency rose diagram is shown in Fig. 28. Nine main wind directions were selected for evaluation, namely  $0^\circ$ ,  $30^\circ$ ,  $60^\circ$ ,  $90^\circ$ ,  $120^\circ$ ,  $150^\circ$ ,  $180^\circ$ ,  $210^\circ$ , and  $240^\circ$ , with the total frequency reaching 92.7% of the prevailing wind speed. Subsequently, the

**Table 4**  
Extended land beaufort scale showing wind effects on people.

| Comfort categories | Maximum wind speeds                          |               |              |
|--------------------|--|---------------|--------------|
|                    | 52 times/year                                | 12 times/year | 1 times/year |
| I                  | 3.6  | 5.4           | 15.2         |
| II                 | 5.4  | 7.6           | 15.2         |
| III                | 7.6  | 9.9           | 15.2         |
| IV                 | 9.9  | 12.5          | 15.2         |
| V                  | Fails to satisfy the aforementioned criteria |               |              |

**Table 5**  
Statistics of meteorological data and Weibull distribution parameters.

| Wind        | Mean wind speed (m/s) | $A_\theta$ | $c_\theta$ | $k_\theta$ | $\mu_\theta$ |
|-------------|-----------------------|------------|------------|------------|--------------|
| $0^\circ$   | 2.5395                |            | 0.0495     | 1.1649     | 0.0778       |
| $30^\circ$  | 3.0548                |            | 0.0999     | 1.3025     | 0.0567       |
| $60^\circ$  | 1.6341                |            | 0.0599     | 1.2198     | 0.0783       |
| $90^\circ$  | 4.1963                |            | 0.2430     | 1.6961     | 0            |
| $120^\circ$ | 4.2102                |            | 0.1758     | 1.4404     | 0            |
| $150^\circ$ | 1.2905                |            | 0.0510     | 1.0        | 0.0376       |
| $180^\circ$ | 1.3938                |            | 0.0917     | 0.9999     | 0.0936       |
| $210^\circ$ | 2.0116                |            | 0.0847     | 1.2035     | 0.0801       |
| $240^\circ$ | 2.0678                |            | 0.0713     | 2.1229     | 1.2351       |
|             |                       |            |            |            | 0.0734       |

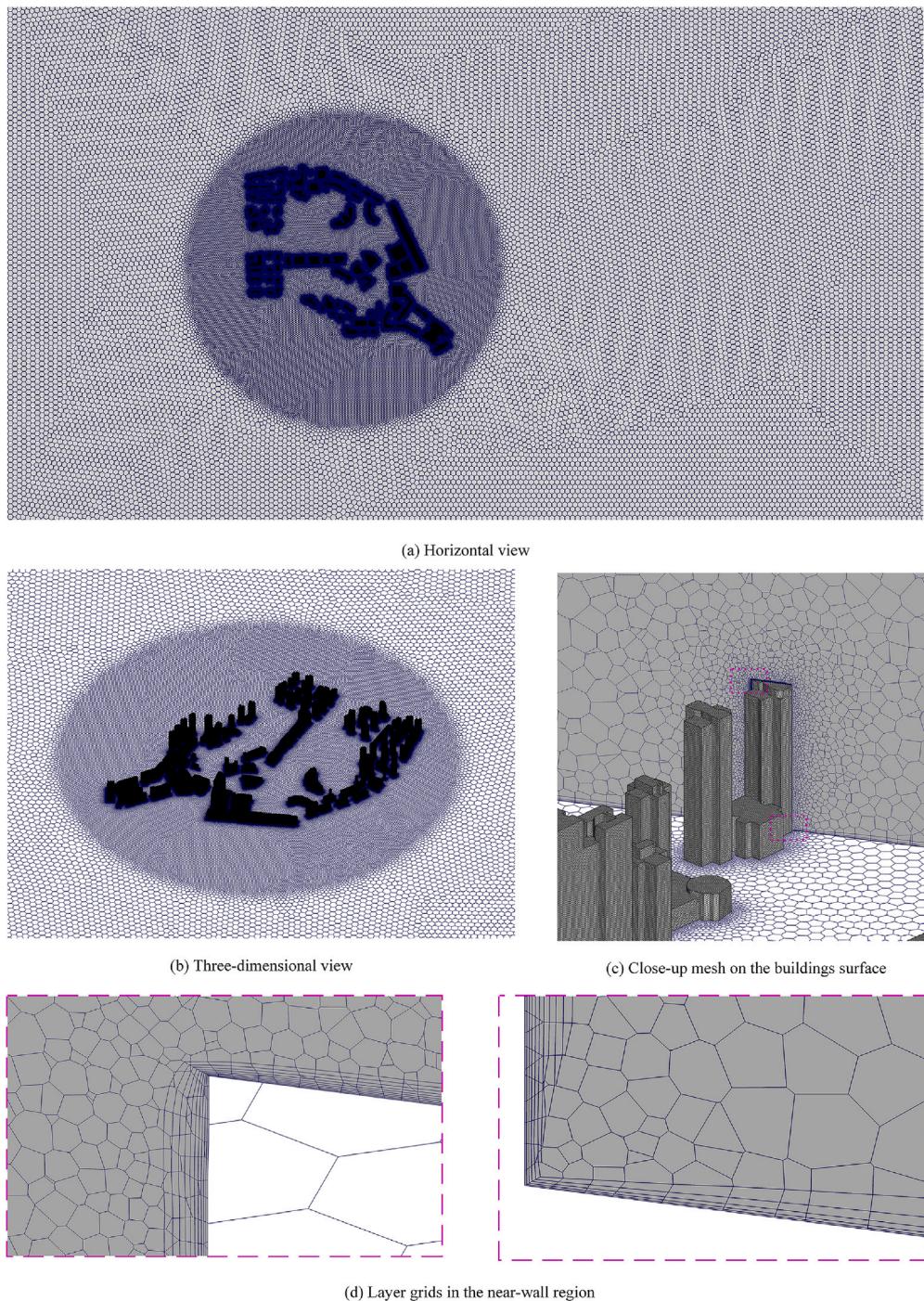
Weibull distribution parameters were then estimated, and the probability density function (PDF) curves of wind speed for different directions are shown in Fig. 29. Frequency and parameter results are shown in Table 5, where the  $A_\theta$  represents the frequency of each wind direction relative to the total occurrences of the main wind directions.

##### 4.2.3. Wind speed ratio

This study used the established grid and model settings to conduct CFD simulations under various conditions for 9 wind direction angles, obtaining the wind speed ratio for the research area. To ensure the accuracy of numerical calculations, a scaled model with a length scale ratio of 1:400 was used. The wind speed ratio  $r_{i,\theta}$  at a specific location  $i$  for a given wind direction angle  $\theta$  is calculated using the following formula:

$$r_{i,\theta} = \frac{\bar{V}_{i,\theta}^{\text{ped}}}{V_S^{\text{ref}}} \quad (21)$$

Where  $\bar{V}_{i,\theta}^{\text{ped}}$  represents the average wind speed at pedestrian height at the specified location under the given wind direction, and  $V_S^{\text{ref}}$  is the wind speed at the reference height for the incoming flow in the research area. In this study, the pedestrian height is set at 2 m, the reference height at 10 m, and the reference wind speed at 5 m/s.



**Fig. 23.** Mesh configuration of the buildings in the study area.

Thirty monitoring points were selected, mainly in pedestrian areas and around large, complex buildings, to evaluate comfort, as these areas are more susceptible to adverse wind speeds. The locations of these monitoring points are shown in Fig. 30. After obtaining the wind speed ratio data, the exceedance probability is calculated using Eq. (20). Subsequently, the comfort category can be determined according to Table 6. When an area meets three relative comfort conditions, it is considered to meet the wind environment comfort requirements for its specific function.

#### 4.2.4. Pedestrian comfort assessment

The contour and vector diagrams of wind speed ratios at pedestrian height for different wind directions are shown in Fig. 31, which capture

the wind field characteristics at pedestrian height. In the areas near measurement points 12, 18, 22, and 23 for the 0° and 30° wind direction angles, narrow channels formed by parallel buildings upstream reduce airflow cross-sectional area, creating a channel acceleration effect, and thus, wind speeds are higher compared to other areas (Zhang et al., 2021). Around measurement points 10 and 12, sharp building corners cause the airflow to contract sharply, increasing the speed. Additionally, the separation and reattachment of the airflow at these sharp corners further alter the airflow state, increasing the wind speed.

At a 0° angle, apart from the channel acceleration effect and building-corner effect in some areas, a large red area appeared around the large building to the right of measurement point 16. This is because the airflow can maintain a relatively stable flow velocity before hitting

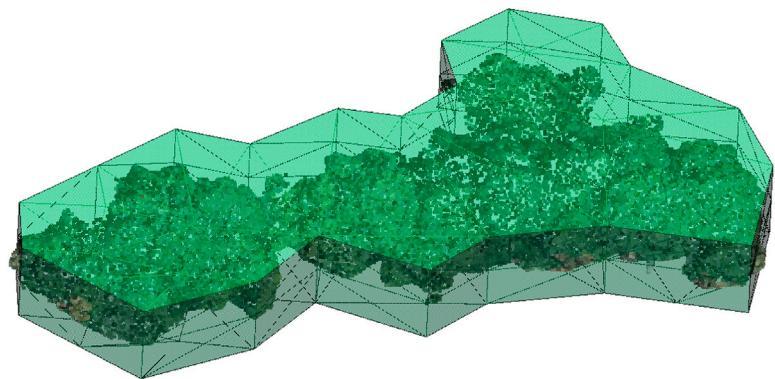


Fig. 24. Vegetation points and prisms.

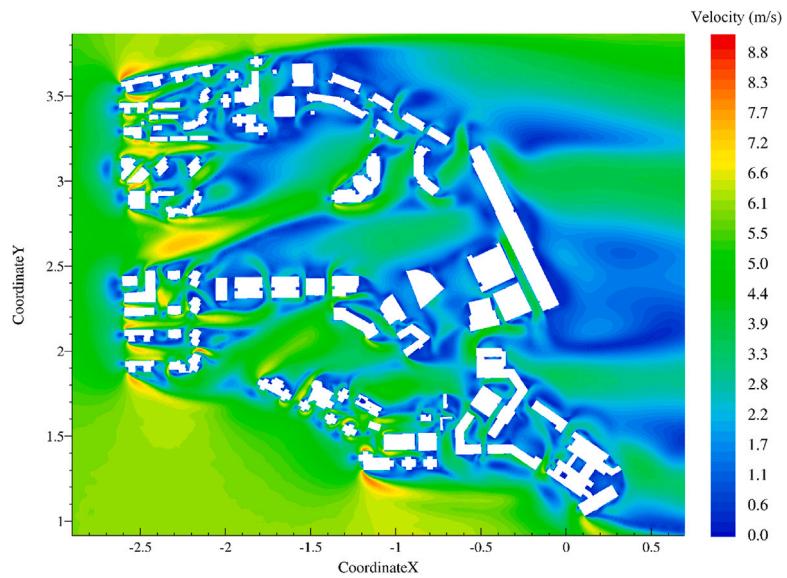


Fig. 25. Velocity magnitude at 2 m in the study area.

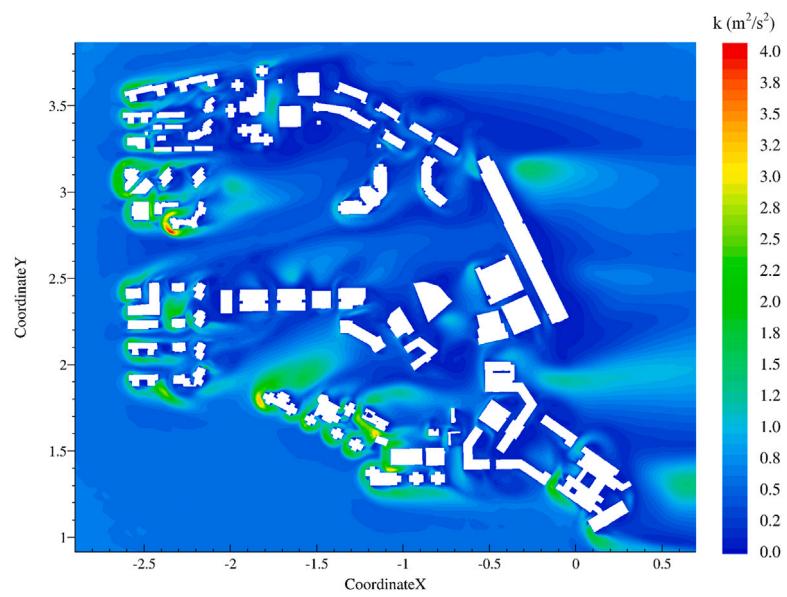


Fig. 26. Turbulent kinetic energy fields at 2 m in the study area.

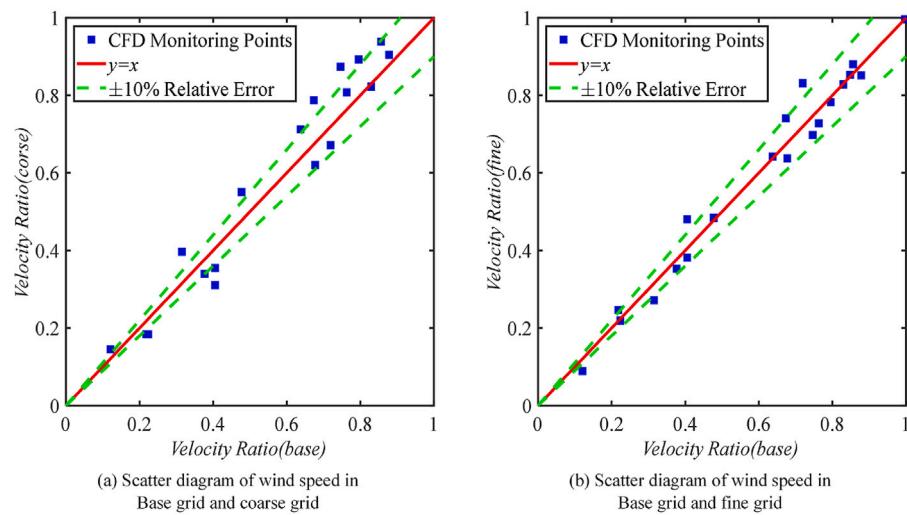


Fig. 27. Scatter diagrams of wind velocity in different grid schemes.

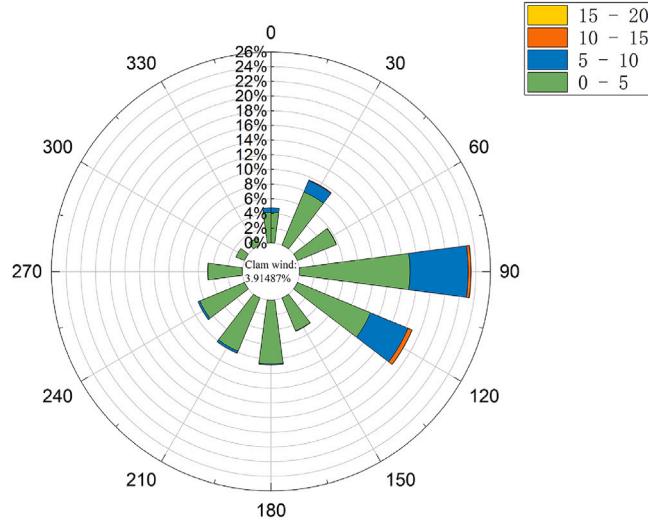


Fig. 28. The wind speed and direction frequency rose diagram of meteorological data.

the building surface. Moreover, due to the obstruction of the building, the airflow accumulates and accelerates at the front edge, resulting in an increase in wind speed. At 150°, 180°, and 210°, most high wind speed areas occur because buildings directly face the oncoming wind without any buffering effect from other structures. This allows the airflow to impact the building surfaces smoothly, leading to higher wind speeds. Meanwhile, in some lower-left corner building areas, the wind is continuously accelerated by the passage effect, forming high-speed winds. In the middle of the study area and areas far from the incoming wind, the combined effects of building obstruction, dissipation, and wind speed diffusion result in relatively low and stable wind speeds.

Based on the comfort classification standards in Table 4, the comfort of 30 monitoring points was evaluated using data from nearby meteorological stations and CFD simulation results. The results, shown in Table 6, indicate that most monitoring points fall under Category I and II, meeting pedestrian comfort requirements. Some points, such as points 25 and 28, are classified as Category III, but since these areas are primarily used for sidewalks and squares, they still meet the required standards. However, point 23 is classified as Category IV, primarily due to its proximity to a small isolated building and the narrow passage

**Table 6**  
Comfort levels of measuring points.

| Measuring point | Comfort categories | Measuring point | Comfort categories | Measuring point | Comfort categories |
|-----------------|--------------------|-----------------|--------------------|-----------------|--------------------|
| 1               | I                  | 11              | I                  | 21              | I                  |
| 2               | I                  | 12              | I                  | 22              | II                 |
| 3               | I                  | 13              | I                  | 23              | IV                 |
| 4               | II                 | 14              | I                  | 24              | II                 |
| 5               | II                 | 15              | II                 | 25              | III                |
| 6               | I                  | 16              | I                  | 26              | II                 |
| 7               | II                 | 17              | II                 | 27              | I                  |
| 8               | I                  | 18              | I                  | 28              | III                |
| 9               | I                  | 19              | I                  | 29              | II                 |
| 10              | I                  | 20              | I                  | 30              | I                  |

effect caused by parallel buildings on the left. Although the point is near a sidewalk and meets usage requirements, caution is still advised during daily use.

In conclusion, the wind environment comfort at pedestrian height in this area shows excellent performance, with most points classified as Category I and II, and only a few as Category III and IV. This demonstrates that the proposed building geometry generation framework not only produces highly detailed building geometries but also ensures accurate wind field characteristics in the field of wind environment studies.

#### 4.3. Visualization on WebGIS

In addition to analyzing pedestrian comfort, this study explores the framework's application in urban planning and disaster prevention. By combining the building geometry generation framework with CFD simulation results, a city wind field database was created. Models and wind fields were visualized using Cesium and displayed on a WebGIS platform. These outcomes can guide urban planning and disaster research. To visualize wind field data on WebGIS and minimize browser storage usage, spatial interpolation or subdivision is needed, creating continuous surfaces based on spatial point relationships (Dangermond & Goodchild, 2020). This study uses the Kriging method (Cressie, 1990) for point cloud data interpolation. Based on variogram theory and structural analysis, this method provides unbiased optimal estimates, predicting continuous value surfaces between known locations (Belkhiri, Tiri, & Mouni, 2020). The dataset selected includes wind speed results at a 30° wind direction angle, 10 m above the ground,

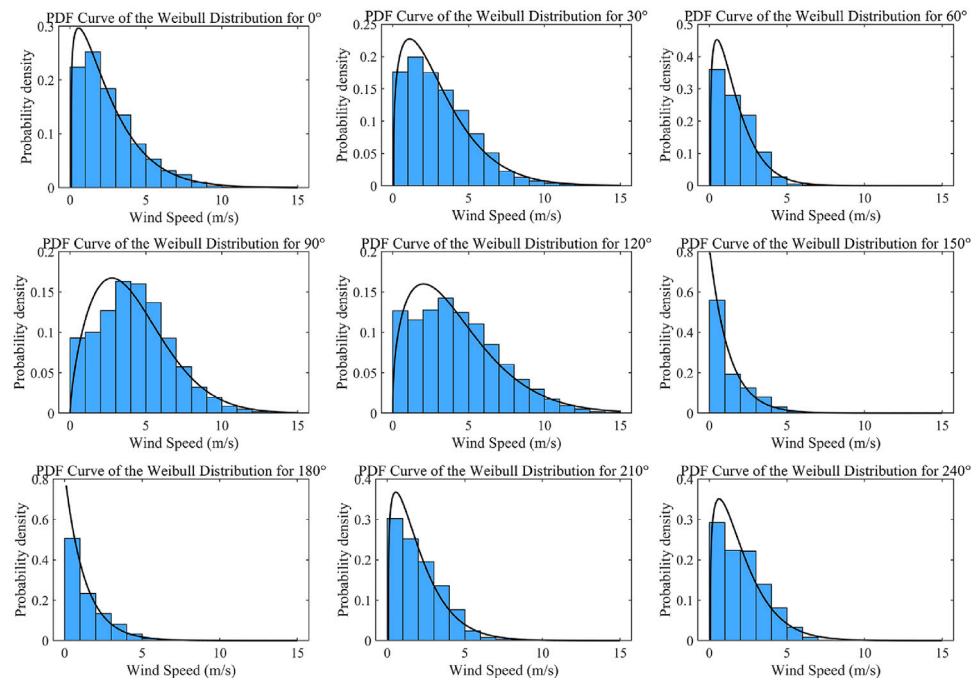
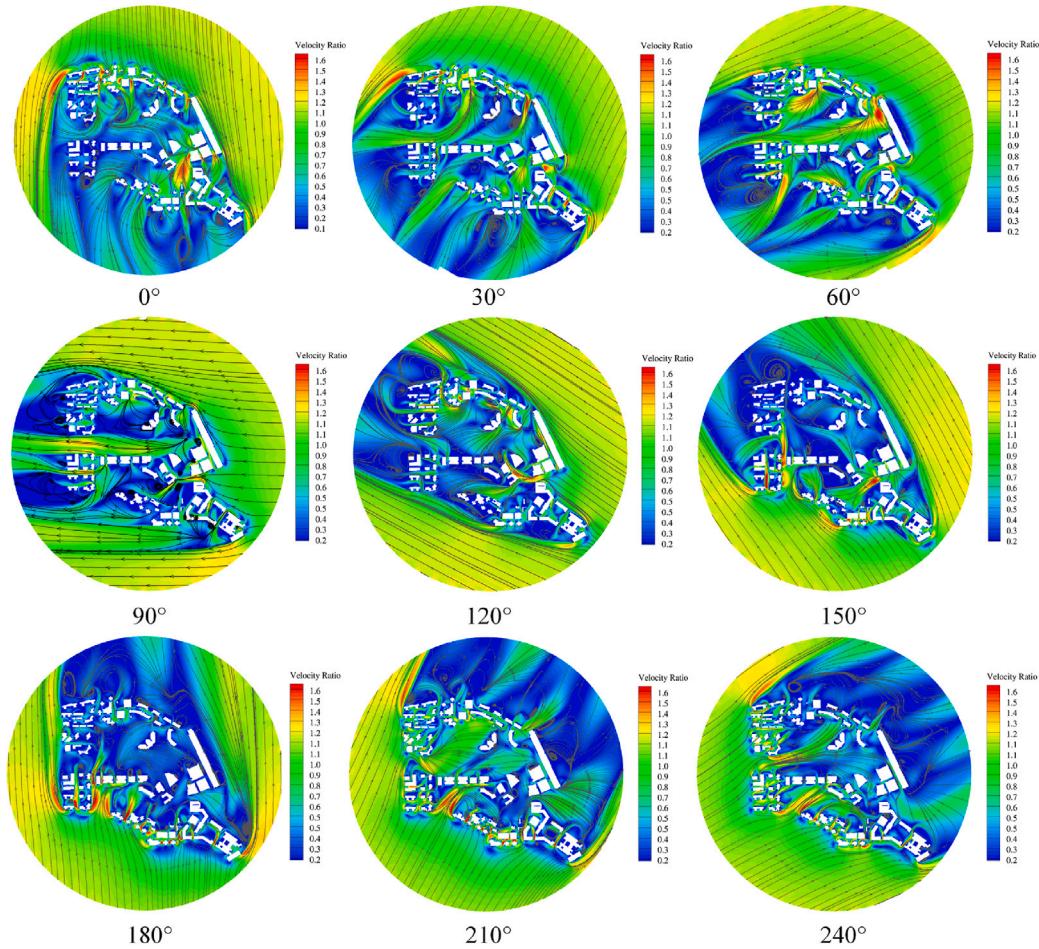


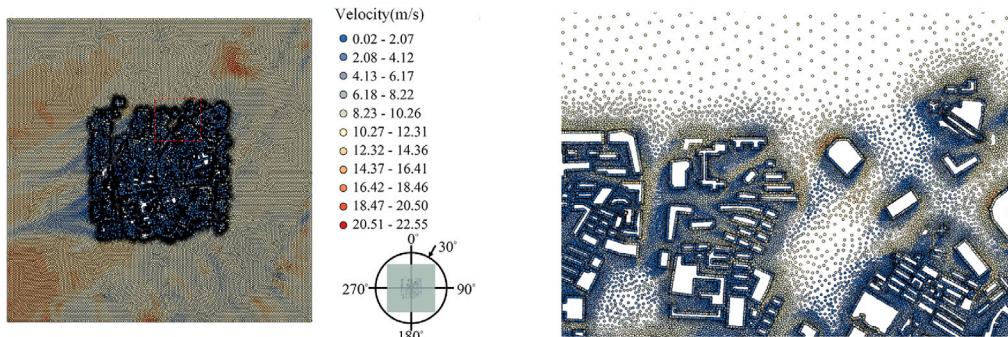
Fig. 29. PDF curves of the Weibull distribution for each wind direction.



Fig. 30. Distribution of monitoring points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 31.** Isograms and vector diagrams of wind speed ratio of pedestrian height section under different wind direction.



**Fig. 32.** The point cloud of the initial wind field. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

from CFD simulations, with approximately 711,000 point cloud data entries. The database storage fields include CELLNUMBER, X, Y, Z, and VELOCITY, totaling five columns.

Fig. 32 shows the initial wind field point cloud visualization. Varying point cloud density is due to grid size differences, with denser points near buildings. Redder point cloud color indicates higher wind speed. Areas with higher wind speeds mostly appear away from the core urban building area. Subsequently, using the empirical Bayesian kriging method (RBF-M) (Antal & Guerreiro, 2021) for interpolation, each subset simulates 100 semivariograms to balance computational resources and visualization effects, with a subset size of 100. The generated interpolation effect is shown in Fig. 33.

After completing the interpolation, the GeoServer service is started using Tomcat, which offers a stable platform for web applications (Vukotic & Goodwill, 2011). Data is then imported into the pre-configured workspace and storage within GeoServer, and the geographic data frame of the layer is set up before publishing it as a WMS layer. Through the Web interface provided by Cesium, the published WMS layer is accessed via a URL link and published in image/png format. This process enables the WebGIS visualization of the wind field data, as shown in Fig. 34. This result represents an initial exploration of integrating this framework with platforms like WebGIS. Further functionalities, such as pedestrian comfort assessment and wind disaster warning platforms, can be added as needed.

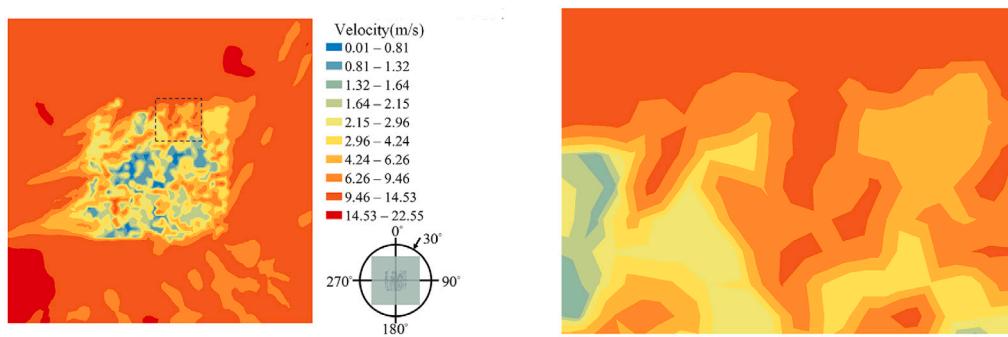


Fig. 33. The interpolation results of the wind field.

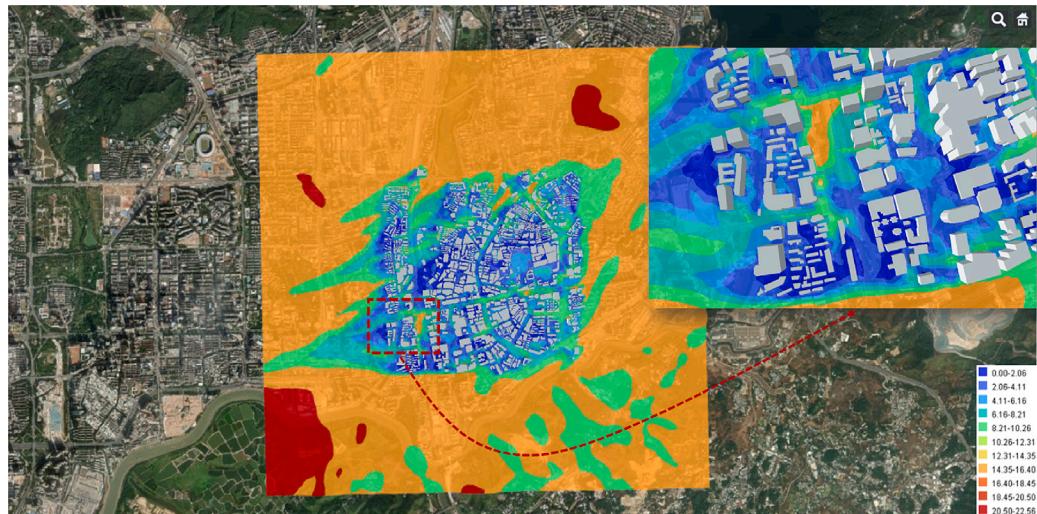


Fig. 34. The interpolation results of the wind field.

## 5. Conclusion and future work

This paper proposes an algorithm framework for generating building geometric models based on 3D Gaussian Splatting to rapidly produce accurate geometric models suitable for CFD calculations. To establish the test dataset, images of the study area were captured using a drone and downsampled. For the first time, 3D Gaussian splatting was introduced and improved, dividing the scene into multiple blocks of the same size based on the SfM sparse point cloud, addressing the issues of insufficient memory for large scene reconstruction and inadequate point cloud accuracy. An integrated algorithm was then designed to extract and refine building roof contours, generating regular geometric models with rich roof details. Results show that the generation speed of dense building point clouds is 2 to 3 times faster than traditional methods, with an average accuracy improvement of 12%. The generated mesh achieved satisfactory quality, and monotonic stable convergence was observed in the numerical simulation, with the mesh convergence index reaching 3.76% in the velocity and pressure fields. These results indicate that the algorithm framework significantly improves processing speed while maintaining high accuracy and generates high-quality building geometric models suitable for CFD simulations.

There are still some limitations in the adopted algorithm framework that need improvement. Firstly, our point cloud generation model may cause the loss of point cloud details when dealing with colors close to the background color and with insignificant variations. Secondly, the geometric model generation algorithm only focuses on the building parts, while the surrounding vegetation and terrain are also important in urban wind environment studies. Finally, the results of geometric reconstruction for complex buildings and residential areas were not as expected.

## CRediT authorship contribution statement

**Peisheng Zhao:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Chao Li:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Data curation, Conceptualization. **Jianxun Jiang:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Data curation. **Lingwei Chen:** Writing – review & editing, Writing – original draft, Validation, Resources, Project administration, Formal analysis, Conceptualization. **Xiaolu Wang:** Writing – review & editing, Writing – original draft, Validation, Resources, Investigation, Data curation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This research has received support from Shenzhen Basic Research Program, China (GXWD20231130143911001), GuangDong Basic and Applied Basic Research Foundation, China (2022A1515240062, 2022A1515240001, 2022A1515140136), and Shenzhen Science and Technology Program, China (KQTD20210811090112003), all of which are gratefully acknowledged.

## Data availability

Data will be made available on request.

## References

- Anaae s, H., Jensen, R. R., Vogiatzis, G., Tola, E., & Dahl, A. B. (2016). Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120, 153–168.
- Alemayehu, T. F., & Bitsuamlak, G. T. (2022). Autonomous urban topology generation for urban flow modelling. *Sustainable Cities and Society*, 87, Article 104181.
- Amorim, J. H., Rodrigues, V., Borrego, C., & Costa, A. M. (2010). A CFD analysis of the vegetative canopy effect on urban air pollutants dispersion. In *Proceedings of the CLIMAQS workshop 'local air quality and its interactions with vegetation' January (pp. 21–22).*
- Antal, A., & Guerreiro, P. M. (2021). A radial basis function approach to estimate precipitations in Brasov county, Romania. *Environmental Engineering & Management Journal (EEMJ)*, 20(8).
- Belkhiri, L., Tiri, A., & Mouni, L. (2020). Spatial distribution of the groundwater quality using kriging and co-kriging interpolations. *Groundwater for Sustainable Development*, 11, Article 100473.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 679–698.
- Chen, H., Li, C., & Lee, G. H. (2023). Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. arXiv preprint arXiv:2312.00846.
- Cheng, K., Long, X., Yang, K., Yao, Y., Yin, W., Ma, Y., et al. (2024). Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first international conference on machine learning*.
- Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, 22, 239–252.
- Dangermond, J., & Goodchild, M. F. (2020). Building geospatial infrastructure. *Geo-Spatial Information Science*, 23(1), 1–9.
- Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2), 112–122.
- Edelsbrunner, H., Kirkpatrick, D., & Seidel, R. (1983). On the shape of a set of points in the plane. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, 29(4), 551–559.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. vol. 96, In *Kdd* (pp. 226–231). 34.
- Fabbri, K., & Costanzo, V. (2020). Drone-assisted infrared thermography for calibration of outdoor microclimate simulation models. *Sustainable Cities and Society*, 52, Article 101855.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Frahah, M. M., & Arisha, A. M. G. (2023). CFD simulation of wind environment in high-rise buildings for wind energy acquisition. *Journal of Progress in Civil Engineering*, 5(11).
- Franke, J., Hellsten, A., Schlunzen, K. H., & Carissimo, B. (2011). The COST 732 best practice guideline for CFD simulation of flows in the urban environment: a summary. *International Journal of Environment and Pollution*, 44(1–4), 419–427.
- Fu, R., Padén, I., & García-Sánchez, C. (2024). Should we care about the level of detail in trees when running urban microscale simulations? *Sustainable Cities and Society*, 101, Article 105143.
- Frurukawa, Y., & Ponce, J. (2009). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8), 1362–1376.
- Galliani, S., Lasinger, K., & Schindler, K. (2015). Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE international conference on computer vision* (pp. 873–881).
- Gu, D., Zhang, N., Shuai, Q., Xu, Z., & Xu, Y. (2024). Drone photogrammetry-based wind field simulation for climate adaptation in urban environments. *Sustainable Cities and Society*, 117, Article 105989.
- Han, X., Leung, T., Jia, Y., Sukthankar, R., & Berg, A. C. (2015). Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3279–3286).
- Heidari, A., Navimipour, N. J., & Unal, M. (2022). Applications of ML/DL in the management of smart cities and societies based on new trends in information technologies: A systematic literature review. *Sustainable Cities and Society*, 85, Article 104089.
- Holmes, J. D., Paton, C., & Kerwin, R. (2007). *Wind loading of structures*. CRC Press.
- Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. vol. 7, In *Proceedings of the fourth eurographics symposium on geometry processing*. 4.
- Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 139–1.
- Kim, S., Lee, K., & Lee, Y. (2024). Color-cued efficient densification method for 3D Gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 775–783).
- Knapsch, A., Park, J., Zhou, Q.-Y., & Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4), 1–13.
- Kopanas, G., Philip, J., Leimkühler, T., & Drettakis, G. (2021). Point-based neural rendering with per-view optimization. vol. 40, In *Computer graphics forum* (pp. 29–43). Wiley Online Library, 4.
- Kwon, A., & Kim, J.-J. (2014). Study on detailed air flows in urban areas using GIS data in a vector format and a CFD model. *Korean Journal of Remote Sensing*, 30(6), 755–767.
- Liu, Y., Guan, H., Luo, C., Fan, L., Peng, J., & Zhang, Z. (2024). Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. arXiv preprint arXiv:2404.01133.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. vol. 2, In *Proceedings of the seventh IEEE international conference on computer vision* (pp. 1150–1157). IEEE.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 91–110.
- M, G. (2012). *Load code for the design of building structures*. Beijing, China: Ministry of Housing and Urban-Rural Construction of the People's Republic of China, Haidian District.
- Menze, M., & Geiger, A. (2015). Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3061–3070).
- Merrill, D. G., & Grimshaw, A. S. (2010). Revisiting sorting for GPGPU stream architectures. In *Proceedings of the 19th international conference on parallel architectures and compilation techniques* (pp. 545–546).
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 99–106.
- Ministry of Housing and Urban-Rural Development of the People's Republic of China (2014). Standard for wind tunnel test of buildings and structures. JSJ/T 338–2014.
- Mirzaei, P. A. (2021). CFD modeling of micro and urban climates: Problems to be solved in the new decade. *Sustainable Cities and Society*, 69, Article 102839.
- Oh, G., Yang, M., & Choi, J.-I. (2024). Large-eddy simulation-based wind and thermal comfort assessment in urban environments. *Journal of Wind Engineering and Industrial Aerodynamics*, 246, Article 105682.
- Paden, I., Peters, R., García-Sánchez, C., & Ledoux, H. (2024). Automatic high-detailed building reconstruction workflow for urban microscale simulations. *Building and Environment*, Article 111978.
- Qiu, Y., He, Y., Li, M., & Zhu, X. (2023). A generalization of building clusters in an urban wind field simulated by CFD. *Atmosphere*, 15(1), 9.
- van Rees, E. (2013). Open geospatial consortium (OGC). *Geoinformatics*, 16(8), 28.
- Ricci, A., Kalkman, I., Blocken, B., Burlando, M., Freda, A., & Repetto, M. (2017). Local-scale forcing effects on wind flows in an urban environment: Impact of geometrical simplifications. *Journal of Wind Engineering and Industrial Aerodynamics*, 170, 238–255.
- Roache, P. J. (1997). Quantification of uncertainty in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 29(1), 123–160.
- Rosnell, T., Honkavaara, E., & Nurminen, K. (2012). On geometric processing of multi-temporal image data collected by light UAV systems. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38, 63–68.
- Schönberger, J. L., Zheng, E., Frahm, J.-M., & Pollefeys, M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, the Netherlands, October 11–14, 2016, proceedings, part III* 14 (pp. 501–518). Springer.
- Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: exploring photo collections in 3D. In *ACM siggraph 2006 papers* (pp. 835–846).
- Stathopoulou, E. K., & Remondino, F. (2023). A survey on conventional and learning-based methods for multi-view stereo. *The Photogrammetric Record*, 38(183), 374–407.
- Sun, C., Zhang, F., Zhao, P., Zhao, X., Huang, Y., & Lu, X. (2021). Automated simulation framework for urban wind environments based on aerial point clouds and deep learning. *Remote Sensing*, 13(12), 2383.
- Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P. P., et al. (2022). Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8248–8258).
- Tominaga, Y., Mochida, A., Yoshie, R., Kataoka, H., Nozu, T., Yoshikawa, M., et al. (2008). AJJ guidelines for practical applications of CFD to pedestrian wind environment around buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(10–11), 1749–1761.
- Turki, H., Ramanan, D., & Satyanarayanan, M. (2022). Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12922–12931).
- Vukotic, A., & Goodwill, J. (2011). *Apache tomcat 7*. Springer.
- Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., & Wang, W. (2021). Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689.
- Wang, X., Wang, M., Wang, S., & Wu, Y. (2015). Extraction of vegetation information from visible unmanned aerial vehicle images. *Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE)*, 31(05), 152–157+159+158.

- Willemse, E., & Wisse, J. A. (2007). Design for wind comfort in The Netherlands: Procedures, criteria and open research issues. *Journal of Wind Engineering and Industrial Aerodynamics*, 95(9–11), 1541–1550.
- Xie, Z., Voke, P. R., Hayden, P., & Robins, A. G. (2004). Large-eddy simulation of turbulent flow over a rough surface. *Boundary-Layer Meteorology*, 111, 417–440.
- Xu, Q., & Tao, W. (2019). Multi-scale geometric consistency guided multi-view stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5483–5492).
- Yao, Y., Luo, Z., Li, S., Fang, T., & Quan, L. (2018). Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision* (pp. 767–783).
- Yi, X., & Zheng, L. (2021). Impact and assessment of urban wind environment change on architectural heritage protection. vol. 237, In *E3S web of conferences* (p. 03035). EDP Sciences.
- Younis, M., Bitsuamlak, G. T., & Sushama, L. (2024). High-resolution regional climate-CFD integrated modelling to inform climate responsive design of northern buildings in a changing climate. *Sustainable Cities and Society*, 115, Article 105773.
- Yu, Z., & Gao, S. (2020). Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1949–1958).
- Zhang, L., Tian, L., Shen, Q., Liu, F., Li, H., Dong, Z., et al. (2021). Study on the influence and optimization of the venturi effect on the natural ventilation of buildings in the Xichang area. *Energies*, 14(16), 5053.
- Zhang, K., Yan, J., & Chen, S.-C. (2006). Automatic construction of building footprints from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9), 2523–2533.
- Zhao, Q., Li, R., Cao, K., Yi, M., & Liu, H. (2024). Influence of building spatial patterns on wind environment and air pollution dispersion inside an industrial park based on CFD simulation. *Environmental Monitoring and Assessment*, 196(5), 427.
- Zhou, Y., Wang, L., Love, P. E., Ding, L., & Zhou, C. (2019). Three-dimensional (3D) reconstruction of structures and landscapes: a new point-and-line fusion method. *Advanced Engineering Informatics*, 42, Article 100961.
- Zwicker, M., Pfister, H., Van Baar, J., & Gross, M. (2001). Surface splatting. In *Proceedings of the 28th annual conference on computer graphics and interactive techniques* (pp. 371–378).