Sensors and Materials, Vol. 35, No. 1 (2023) 247–268
MYU Tokyo

S & M 3166

A Review of Indoor Automation Modeling Based on Light Detection and Ranging Point Clouds

Yang Cui,[1,2] Bogang Yang,[1,2*] Peng Liu,[1,2] and Lingyan Kong[1,2]

[1]Beijing Institute of Surveying and Mapping, 15 Yangfangdian Road, Haidian, Beijing 100038, China
[2]Beijing Key Laboratory of Urban Spatial Information Engineering,
15 Yangfangdian Road, Haidian, Beijing 100038, China

(Received October 30, 2022; accepted January 10, 2023)

Keywords: 3D indoor modeling, laser scanning sensor, standards, point cloud acquisition and characteristics, object classification, room segmentation, model reconstruction

3D modeling of the indoor environment is essential for urban applications such as indoor navigation, emergency simulations, floor planning, and building construction. With the development of laser scanning sensors, 3D laser scanners can quickly obtain high-density, high-precision 3D coordinates and attribute information, which brings significant advantages in collecting 3D information on indoor scenes. Many studies have been published on the fast reconstruction of 3D models based on point cloud data obtained by various types of laser scanning sensors. In this paper, we review state-of-the-art automated 3D indoor modeling technologies. The 3D modeling standards for indoor environments are introduced, and data acquisition based on laser scanning sensors and characteristics of point clouds are discussed. Indoor object classification and indoor room segmentation are also examined in detail. The 3D indoor reconstruction methods (i.e., line-based, plane-based, and volume-based) are systematically introduced and the advantages and disadvantages of these methods are presented. Future research directions in this field are discussed and summarized. This review can help researchers improve current approaches or develop new techniques for 3D indoor reconstruction.

1. Introduction

With rapid developments in communication and computer technology, urban informatization has gradually developed from a digital city to a smart city. However, realizing a smart city requires the use of digital spatial information to express the real world, making urban spatial structures more digital, transparent, and convenient to serve social needs.(1) Accurate, detailed, and structured building models are crucial in developing smart cities. Today, more than 80% of human activities (e.g., office work, commercial activities, learning, and living) are conducted indoors, significantly increasing demands for spatial information about indoor environments.

*Corresponding author: e-mail: bogangy@126.com
https://doi.org/10.18494/SAM4211

ISSN 0914-4935
https://myukk.org/

As abstract plane expressions of geographical space objectives, 2D maps are traditionally the source of indoor space analysis. Such maps are not intuitive enough to express complex indoor environments and lack spatial structure relations, and they are unable to provide indoor 3D space services, such as indoor navigation,[2,3] emergency services,[4,5] architectural design,[6] virtual reality,[7] energy consumption estimation,[8] signal simulation,[9] and other requirements of a smart city. In recent years, increased attention has been given to acquiring indoor 3D spatial data, and numerous modeling approaches have been developed for reconstructing high-precision 3D indoor models. The most traditional is the manual reconstruction of high-precision models with commercial software (e.g., CAD, Revit and 3DReshaper, SketchUp, and 3DMax), which requires a significant amount of manual processing, has high labor costs and low efficiency, and is difficult to adapt to large-scale indoor mapping and modeling needs.

With the development of laser scanning sensors in the past decade, 3D laser scanners can quickly obtain high-density, high-precision 3D coordinates and attribute information on objects sitting on surfaces, which are not affected by light and have strong penetrability.[10] As a highly effective means to obtain spatial data for real-scene 3D reconstruction, numerous studies have explored using various types of laser scanning equipment and optical imagers for indoor spatial data acquisition.

Sensors can be divided into three categories according to the data source (see Fig. 1): red green blue depth (RGBD) sensors, terrestrial laser scanners, and mobile laser scanners (e.g., handheld, backpack, or push-cart). An RGBD panorama may be acquired by a camera and a depth sensor,[11] and this method has added advantages of affordability and convenience. However, image acquisition is affected by lighting conditions, and an image may have both distortion and noise, which can cause difficulties in building accurate large-scale models. Terrestrial laser scanning (TLS) has been used to collect 3D geometric information with high density and accuracy. Owing to the need for laborious scan station resetting, protracted registration procedures, and high costs, TLS often suffers from low mapping efficiency and is often excluded in large-scale indoor data acquisition. With developments in simultaneous localization and mapping (SLAM), the indoor mobile light detection and ranging (LiDAR) measurement system integrates an inertial measurement unit (IMU), laser scanner, digital camera, and other instruments and equipment to obtain high-density and high-precision 3D laser point clouds continuously and to reduce occlusion effects. The easy-to-use, low-cost mobile laser scanning (MLS) equipment has been widely used for data acquisition for large indoor scenes.[12] Owing to the complexities of indoor environments, data collection is affected considerably by numerous factors, such as moving objects and multiple reflections. These issues may result in large deviations in point clouds, huge amounts of noise, and uneven density distribution and therefore create challenges for the automatic reconstruction of indoor models.

Integration of LiDAR into smartphones opens up a whole new world of possibilities for 3D indoor/outdoor mapping since the iPhone 12 Pro and iPhone 12 Pro Max became available. Although these systems offer new opportunities for democratization in the use of scanning technology, data quality is lower than data captured from high-end LiDAR sensors. In large indoor scenes, the point clouds are clearly incorrect: walls may not be perpendicular, and they

(a) (b) (c)

(d) (e) (f)

Fig. 1. (Color online) Different types of data acquisition equipment: (a) RGB-D (Kinect), (b) TLS, (c) MLS (handheld), (d) MLS (backpack), (e) push-cart (MLS), and (f) iPhone 12 Pro Max.

may be represented by partially overlapped surfaces. The detection range of the device is only 5 meters, and it is only stable for mapping small environments.[13,14]

Point clouds collected by various laser scanning sensors provide robust data for indoor model reconstruction. Furthermore, the reconstructed indoor model may be useful for applications involving urban spaces. Therefore, the amount of research on the reconstruction of indoor models using point clouds has substantially increased in recent years, and many methods that address some of the limitations have been proposed. In this paper, we comprehensively review and discuss the current literature, techniques, and issues of indoor model reconstruction. In Sect. 2, we introduce the open model standard, LiDAR measurement system for indoor scenes, and the characteristics of the collected data, while in Sect. 3, we explain the classification and extraction of indoor scenes (including indoor object classification and room segmentation). In Sect. 4, we introduce the technologies related to the 3D reconstruction of indoor environments based on line, plane, and volume elements according to different types of vectors. Additional discussion of the advantages and disadvantages of existing methods and future research directions are provided in Sect. 5, after which we present conclusion in Sect. 6.

2. Preliminaries

2.1 Indoor modeling standard

As the foundation of modeling, the building model standard has been improved over years of development. It includes the City geography markup language (CityGML)[15] formulated by the Open Geospatial Consortium (OGC), the Industry Foundation Classes (IFC) formulated by buildingSMART,[16] and indoor geography markup language (indoorGML).[17]

Published by OGC in 2012, CityGML 2.0 defines detailed standards for level of detail (LOD)0–4 in describing building models. Their CityGML3.0, released in 2018, integrates the LOD4 indoor model version 2.0 into LOD 0–3, which is more compatible with indoorGML and IFC standards. CityGML is a format used for data exchange and storage of virtual 3D city models and is a general data model used to express 3D city templates. It has geometric, topological, semantic, and appearance information on targets.

BuildingSMART defines the conceptual data model and exchange file format of building information modeling (BIM) data[18] and develops global BIM technologies, which are accepted by International Organization for Standardization (ISO) 16739 standards. BuildingSMART develops and maintains IFC standards that include indoor spaces and outdoor spaces. For indoor space models, IFC mainly defines characteristics for indoor structural models, such as walls, doors, panels, and windows.

Developed by OGC in 2012, the standard data model IndoorGML is based on Extensible Markup Language (XML) to exchange modeling data formats. IndoorGML is not an independent standard but is derived from CityGML LOD0–4 and IFC.[19] IndoorGML contains the definition of indoor space, which has a semantic, topological, and geometric model framework[20] and is intended to meet the application requirements of indoor space.

2.1.1 CityGML model standard

Organized and compiled in 2002, CityGML is committed to the development, commercialization, and spatial visualization of exchange 3D models and realizes the data storage and exchange of urban 3D models through XML.[15] The CityGML standard model covers indoor and outdoor spaces. In CityGML 2.0, the building space is represented by five consecutive LOD models (LOD0–LOD4), as shown in Fig. 2.

Fig. 2. (Color online) Building hierarchy model in CityGML2.0 (LOD0-LOD4). (https://www.citygml.org/ongoingdev/tudelft-lods/)

LOD0 is the 2.5D terrain model composed of triangular irregular networks.
LOD1 is a simple cube that represents the target location of the urban scene and does not have structural characteristics.
LOD2 is a 3D model composed of boundary surfaces with a detailed roof structure.
LOD3 is the volume model composed of a detailed roof, floor, ceiling, wall, door, window, and other body elements.
LOD4 is the expression of the detailed information of the indoor room and building furniture.

In CityGML 3.0, the building space is represented by four consecutive LOD models (LOD0–3), integrating the indoor grade model LOD4 in CityGML into LOD0–3 (Fig. 3).

2.1.2 IFC model standard

The main contribution of the IFC standard is the research and development of global BIM technology, which defines the conceptual data model and exchange file format of BIM data.[18] BIM is increasingly used for comprehensive construction planning and turning facility management into a common digital foundation. IFC defines BIM as a wide range of geometric and abstract entities with spatial relationships. It differs from a single geometric representation, such as unordered point clouds, unconnected surface features, and unstructured boundary grids.

The BIM/IFC model is very similar to the actual physical structure of a building that has the building entity model (wall, floor, ceiling) with semantic, geometric, structural, and attribute information and volume elements that have interconnection information.[21] Because BIM models have increasingly been applied in many fields, such as difference detection between BIM models and actual completion point clouds,[22] building management, and energy simulation,[23] the automatic reconstruction of the BIM model has become a research topic of considerable current interest.

2.1.3 IndoorGML model standard

The IndoorGML standard model expresses, stores, and exchanges indoor space information, in support of the indoor location service system.[17] IndoorGML provides a standard framework to express the geometric, topological, and semantic features of indoor space cells and uses the XML application architecture extension language. The space cell expressions are as follows:

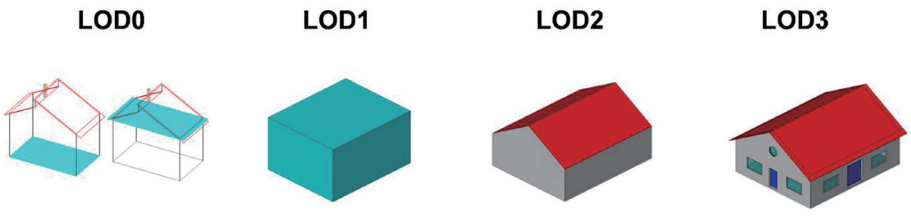

Fig. 3. (Color online) Building hierarchy model in CityGML3.0 (LOD0-LOD3).

(1) geometric information: it contains the 2D surface model and 3D volume model.
(2) topological graph: The core module of indoorGML is the spatial structure model. A topological graph is formed by the geometric units of 2D surfaces and 3D volumes; the spatial objects are the graph nodes, and the adjacent objects are the edges of the graph. The specific contents include: 1) an ID for each cell belonging to the index number of the room; 2) a common boundary for each cell with other cells that do not cover each other; and 3) location information for each cell. The diagram for this is shown in Fig 4.
(3) semantic information: both indoor units and boundaries have attribute information, such as corridors, stairs, elevators, rooms, doors, and windows.

The IndoorGML standard has been applied to a variety of indoor location services, such as blind navigation,[24] indoor photo location,[25] and other smart city applications.[26]

2.2 Mobile LiDAR measurement system and data characteristics

According to analyses of human activity trajectories, more than 80% of activities are conducted indoors. Thus, indoor data collection and 3D model reconstruction are important tasks in urban construction. Compared with outdoor locations, the indoor environment is more complex, has serious occlusions, and has no global navigation satellite system (GNSS) signals; the model accuracy requirements are also higher (about 0.1 m).[27]

With developments in sensor technology, various types of laser scanning and optical imaging instruments have been developed for obtaining indoor 3D spatial data. The main instrument types are RGBD depth sensor, TLS, and MLS. The easy-to-use, low-cost MLS equipment has been widely used for data acquisition for large indoor scenes. We focus herein on the MLS measurement system and discuss the characteristics of collected data.

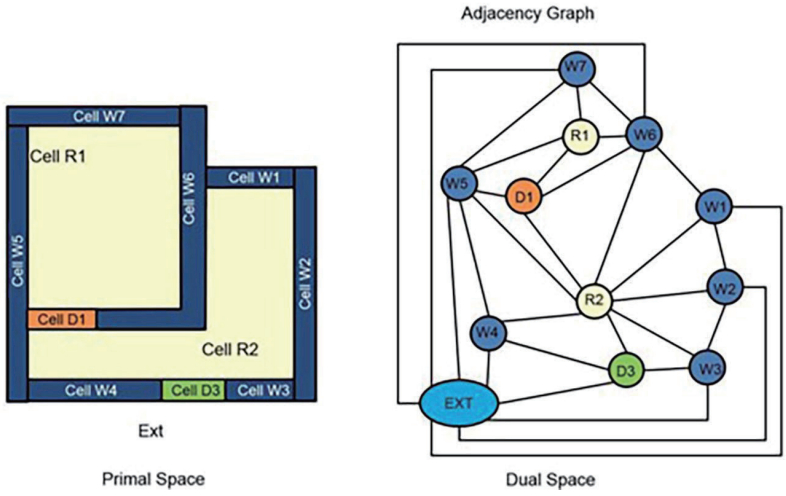

Fig. 4. (Color online) Topological graph of indoor space.[17]

2.2.1 Mobile LiDAR measurement system

In the past decade, mobile measurement systems have been widely studied by industry practitioners and academic researchers, mainly for surveying and mapping, to improve the acquisition of indoor 3D laser point clouds. The indoor mobile 3D mapping hardware system consists of a mobile platform and with one or more sensors installed on it. Mobile platforms generally include a handheld, a backpack, and a push-cart. The sensors can be divided into two categories: sensors that perceive the environment, and sensors that sense the state of motion (position, speed, and posture). The sensors that perceive the environment include cameras, laser radar, and RGBD cameras, while the sensors that sense the state of motion mainly include inertial navigation sensors, odometers, and magnetometers.[28]

For complex indoor environments without GNSS, the mobile measurement system adopts the SLAM technology consisting of two main types: visual SLAM (VSLAM), based on a binocular camera or depth camera integrated with IMU; and a LiDAR-based SLAM (LSLAM) integrated with IMU. The hardware system can synchronously collect inertial measurement data, laser radar, and image texture data. Inertial measurements, LiDAR data, and other data files stream into the software system, and the laser-assisted inertial navigation mapping system obtains centimeter-level 3D trajectories and point clouds. Compared with the traditional methods of mobile measurement, the mobile measurement system is independent of GNSS, flexible, and convenient, greatly improving the efficiency of data collection.[28]

Owing to advancements in SLAM technology, the price of mobile measure equipment has declined, its volume and weight have decreased, and its performance has improved. Given the progress achieved in practical SLAM navigation, the technology has been applied to many tasks and devices, including robot positioning and navigation, augmented reality and virtual reality (AR/VR), unmanned aerial vehicles (UAV), and unmanned driving.[29]

2.2.2 Mobile LiDAR point cloud characteristics

The SLAM mobile measurement system adopts a closed-loop route in acquiring data; the starting and ending points are the same, so the 3D laser point clouds and trajectories can be obtained simultaneously. However, owing to the complexities of the indoor environment, abundant targets, moving objects, and multiple reflections, point clouds generated from mobile LiDAR measurements may have significant shortcomings that seriously impede the reconstruction of high-quality indoor models.

The data characteristics of mobile LiDAR systems are as follows:

(1) 3D coordinates with high density and precision

The measurement area is not affected by GNSS signal limitations, lighting, or weather conditions. The indoor equipment for mobile LiDAR measurements is self-positioned through position estimation and feature matching and has stable ranging performance. Therefore, the point cloud data collected can achieve centimeter-level accuracy and support 3D target extraction and model reconstruction.[30] Laser point clouds are also characterized by high density that can

reach hundreds or even thousands of points per square meter, which is useful for calculating local features.

(2) Discreteness

The mobile LiDAR system obtains the 3D coordinates of a target by emitting a pulsed laser beam and measuring the time and distance traveled. The distance between the target points is uneven and discrete. Unlike the features of 2D images stored in a plane array, the 3D laser point cloud is discrete and unstructured, causing difficulties in managing the space of massive data, displaying 3D in real time, extracting target elements, and reconstructing models. Currently, spatial indexes are usually constructed using kdtree, quadtree, or octree.[30]

(3) Data loss

In a complex indoor environment, incomplete laser point cloud data is an unavoidable problem. For example, objects with different materials absorb signals; the mirror reflection of smooth object surfaces may cause echo information to be unreceivable, resulting in data loss. Stairs, glass, and moving objects can result in occlusion and drift that may make the target expression incomplete and weaken spatial relationships, further affecting the object extraction and classification of point clouds.[30]

(4) Uneven density distribution

The data collected by the mobile LiDAR measurement system usually has a large number of redundancies in the local area, while the local area itself is relatively sparse.[30] Before data processing, the original point clouds are usually sampled down using a uniform grid to reduce the problem of uneven density without affecting the point cloud characteristics.

(5) Geometric position distortion

The point cloud data acquired by the SLAM mobile measurement system is solved through a graph optimization algorithm. However, the algorithm has numerous limitations, such as geometric structure deformation, data drift, local structure layering, and overlapping of point clouds.[30] These limitations can result in substantial gaps between point cloud-generated images and the real indoor structure. Some examples include a wall not being vertical to the ground, the ceiling not being parallel to the floor, and other geometric structural deviations.

(6) Short effective observation distance

The mobile LiDAR measurement system is generally equipped with a Velodyne VLP-16 or Velodyne VLP-32 laser scanner. The effective distance of the collected point clouds is only 30–100 meters, which is only applicable to limited indoor spaces and not suitable for large-scale outdoor measurements.[30]

3. Classification and extraction of indoor scenes

Indoor scenes include offices, residential houses, parking lots, and shopping areas. These scenes are composed of rigid structural elements such as ceilings, floors, walls, windows, doors, columns, individual room spaces, and indoor furniture. The extraction and classification of indoor objects and room segmentation provide semantic information for the reconstruction of an indoor scene.

3.1 Indoor object classification

The two main types of indoor point cloud classification method are: (1) classification of indoor structures based on geometric and semantic information on segmented planes and (2) classification of indoor objects by machine learning or deep learning.[31,32]

3.1.1 Classification of indoor structures based on geometric and semantic information on segmented planes

The point clouds of indoor structural elements (e.g., floors, ceilings, and walls) are smooth and interconnected, so the direction of normal vectors is consistent. The common algorithms for extracting structural elements are region growing, random sample consensus (RANSAC),[33] and Hough transform.[34] These algorithms cluster point clouds with similar geometric characteristics into plane elements or fit them into line elements. The differences between indoor structures can be better expressed through geometric features (e.g., normal vector, point number, shape, angle deviation) or using semantic information planes or lines.

For instance, Sanchez and Zakhor[35] divided point clouds into four categories (i.e., floor, ceiling, wall, and other objects) according to normal vectors and adjacency of planes. Previtali *et al.*[36] proposed using the ray tracing method to detect openings and classify openings into windows and doors by regularization. Díaz-Vilariño *et al.*[37] proposed combining orthoimages and laser point clouds to extract doors. In their approach, the boundary of the candidate doors was extracted using the image. The doors and other objects were then classified according to the size and shape of the extracted boundaries and the geometric information from the point cloud.

3.1.2 Classification of indoor objects based on machine learning and deep learning

Many studies have classified point clouds using traditional machine learning or increasingly popular deep learning techniques. The statistics of features in the sample data are calculated according to the neighborhood information from the point clouds. On the basis of these features, classifiers are established to support semantic information extraction and feature classification of point clouds, which provide classification labels for each point or voxel in the indoor scene.

Descriptions of features by point clouds are mainly based on two methods: point cloud geometric constraint features and deep network learning. Such descriptions provide a morphological structure and serve as the basis of point cloud segmentation and classification. The feature vectors of point cloud geometric constraints include descriptors of the values of features,[38] fast point feature histograms,[39] rotation projection statistical feature description,[40] and binary shape context.[41] The feature vectors of the point cloud depend primarily on prior knowledge, are trained using the original point cloud samples, and are classified point by point. Common classifiers include the random forest,[42,43] support vector machine,[44] JoinBoost,[45] Expectation Maximum,[46] conditional random field,[47] neural oscillator network,[48] deep learning,[49] and adaptive boosting (AdaBoost).[50] Random forests[43,51,52] and JoinBoost[45] are also often used for feature selection.

For instance, Armeni *et al.*[53] proposed a semantic classification method in which a logarithmic, linear model is established to realize the classification of indoor elements by maximizing the energy function. The weight of the energy function is determined by learning from a structured support vector machine.[54] Rottmann *et al.*[55] proposed a supervised learning method to perform semantic classification for objects in different positions. This method first integrates image and laser data to extract features, then applies the AdaBoost algorithm[56] to form a strong classifier; finally, an implicit Markov model is used to improve the final classification accuracy.

Parameter design for machine learning depends primarily on prior knowledge, which is not suitable for complex environments. With the development of deep learning, feature expression can automatically be learned from a large amount of training data, and many parameters can be used to express scene features, thereby improving classification accuracy and algorithm adaptability.[57] Deep learning technology provides new possibilities for point cloud classification.

Deep learning models can be classified into three categories: voxel-based, multi-view-based, and irregular-point-based. 1) Voxel-based model. VoxNet[58] divides space into regular 3D voxels and then uses a 3D convolutional neural network to obtain descriptions of features of the point cloud. 2) Multi-view-based model. This method mainly projects 3D data from different perspectives to obtain 2D rendered images and uses an image convolution network to learn features of 3D data.[59] However, the results of this model depend on the method and perspective of the rendering. 3) Irregular-point-based model. PointNet[60] directly processes point cloud data using depth learning for the first time and uses multi-layer perception (MLP) to obtain point-by-point features. The largest feature pooling method is adopted to achieve invariance in order to replace and describe the global features of the point set, generating better results for the expression of features. PointNet++[61] adds multi-level information and uses PointNet for local feature learning to achieve the fusion of global and local features. The biggest bottleneck for deep learning is the need for a large amount of training sample data and the generalization ability of the learning network.[62–64] Compared with deep learning of 2D images, owing to the large amount of data and complex features, deep learning for 3D point clouds still requires substantial improvements to overcome the limitations in training efficiency and network architecture design.

3.2 Indoor room segmentation

3.2.1 Room segmentation based on TLS point clouds

Room segmentation based on TLS point clouds uses the point clouds collected by each station as initial information. Oesau *et al.*[65] first proposed using a binary graph cut algorithm to segment space; however, the method only distinguishes the indoor and outdoor areas, and a single room could not be semantically divided. Ochmann *et al.*[66] proposed a class-conditional probability iterative clustering method for room segmentation. This method usually had over-segmentation in a long space, which required manual interaction to merge the over-segmented

space. Mura *et al.*[67] constructed a diffusion map to spread the similarity between plane units and completed space division by iteratively clustering 2D units. Ikehata *et al.*[11] used k-medoids clustering, which involves combining the regions whose distance from neighboring cluster centers is less than a certain threshold to complete room segmentation, and in which the distance measure of clustering depends on the binary visual vector of the scanning center. In general, room segmentation algorithms using point clouds obtained by TLS have fewer iterations and high time efficiency, but due to limited TLS data, this method is not convenient for large scenes.

3.2.2 Room segmentation based on MLS point clouds

Turner *et al.*[68,69] proposed an approach in which the triangle with the larger circumscribed circle is selected as the seed point for room estimation in the plane of the building model. A dual graph of all triangles in the plane is then formed, and the minimum cut method is used to segment the rooms. However, knowing the number of rooms is a precondition for seed point clustering, which limits the practicability of the algorithm. Wang *et al.*[70] proposed using the morphological clustering method to segment rooms, which was similar to Mura *et al.*[67] in that it built diffusion maps to cluster units in the same room. Díaz-Vilariño *et al.*[71] used timestamp information in determining the point clouds of each trajectory point and built the minimization energy function for global space optimization to complete the segmentation of a single room. Ochmann *et al.*[72] proposed a fully automatic room segmentation, which used ray tracing for visibility analysis of point patches in planes and thereby constructed visibility graphs. Then Markov clustering was used to cluster the nodes of the graph to complete the room segmentation. Bormann *et al.*[73] introduced morphological and distance transformation-based segmentation methods. Morphological segmentation converts the entire point cloud into a binary image, carries out morphological corrosion transformation, and determines the divided space through connectivity analysis. If the divided space is between the minimum and maximum area thresholds, marked as a room label, the process is repeated until the labels of all 2D pixels are determined. The room segmentation based on distance transformation converts the binary graph of the indoor scene into different regions. The local maximum of the distance transformation is always at the center of the room. Therefore, if the distance transformation result is properly thresholding, the center of the room can be obtained. The key principle of the algorithm is to cycle all possible thresholds in descending order to determine the center label of the room. The wavefront propagation strategy is then used on the unlabeled space to complete the room segmentation. This method is similar to morphological segmentation, which is prone to over-segmentation or under-segmentation. For example, Li *et al.*[74] proposed a comprehensive segmentation method created using morphological erosion and connectivity analysis methods on floor space, which can overcome over-segmentation in long corridors. Zhao *et al.*[75] proposed an iterative Gaussian mapping-based segmentation strategy that has been discussed in this article, which goes from a rough segmentation to a refined one iteratively to decompose the indoor scene into detectable point cloud clusters layer by layer. The method retains sharp structures and details of the indoor scene.

The classification and extraction of indoor objects give semantic information on unordered point clouds. However, the target point clouds are highly redundant and contain large amounts of data. They do not have topological relationships and spatial structure information, so they are not convenient for 3D adaptive expression, spatial analysis, and location services. Therefore, it is necessary to build 3D vector models with topological relationships and spatial structures based on the extracted geometric primitives. Owing to the complex structure of buildings, serious occlusion of objects, and large noise, automatic reconstruction of high-quality 3D models is a challenging task.

4. Research Status of 3D Indoor Model Reconstruction

Developments in science and technology, coupled with the increasing demands for social services and natural resources, have promoted the growth in the construction of 3D real scenes in China. The 3D reconstruction of real scenes can be divided into terrain level, city level, and component level according to the content and level of expression. Indoor model reconstruction is one of the main tasks of component-level construction.

Unlike digital surface models that have redundant mesh grid reconstruction, the key to a reconstructed structural model is accurately extracting the building's structural elements and converting them into line, surface, and volume vector models with topological connections. The reconstruction of 3D models from point clouds can be classified into three types: data-driven,[76,77] model-driven,[78–80] and hybrid-driven.[81,82] Model-driven methods fit the building point clouds to the most appropriate models from a pre-defined model library. However, this method is limited by the primitive types in the model library and cannot reconstruct complex structures. Data-driven methods can reconstruct building models with multi-levels but are sensitive to data quality and depend mainly on the accuracy of extracted geometric primitives. Therefore, high-precision models are usually constructed using hybrid-driven methods that combine the advantages of data-driven and model-driven approaches.

On the basis of the types of vector, reconstruction methods for indoor spaces can be divided in terms of three strategies: line-based, plane-based, and volume-based. Different types of vector data represent models with different levels of detail. Models reconstructed based on line features mainly represent scene details with high modeling accuracy. Plane-based reconstruction primarily expresses the main contour information of a scene. Models based on volume element reconstruction typically represent the real 3D shape of scenes and the structural components of the buildings. This approach is commonly used to visualize 3D solid models and is applied to spatial analysis and other applications.

At present, to promote research on indoor modeling methods and the evaluation of reconstructed models for accuracy, the International Society of Photogrammetry and Remote Sensing (ISPRS) had released the WG IV/5 "3D indoor modeling" dataset.[83] Owing to the complex structure of indoor scenes, serious occlusion between objects, and the considerable noise associated with local point clouds, automatic reconstruction of high-precision 3D models remains a challenging problem.

4.1 Line-based reconstruction

The extraction of line elements can be divided into 2D image-based extraction and 3D laser point cloud extraction according to the different types of data involved. The traditional Hough transform [84] maps each data point to a discrete parameter space and selects the parameter space with the largest vote to fit the lines. Similar methods include the Roberts Cross, Sobel, or Canny edge detectors.[85] These methods have considerable limitations, producing many erroneous detections in high-density texture regions and resulting in line segments with abnormal directions. In addition, the algorithm is not robust, because it sets a fixed threshold to extract the whole line segment.

Burns *et al.*[86] proposed a method of clustering image gradient orientations in extracting line elements, ignoring boundary points and gradient levels. The extracted features are not regular lines but detailed, textured features and local objects. Matas *et al.*[87] proposed the progressive probabilistic Hough transform (PPHT) to extract the lines, which accelerated the computational efficiency of randomly selected edge points, the effect of which was to significantly improve results compared with those from with the standard Hough model. While PPHT has many advantages over the standard HT in many circumstances, the method only detects the entire line and does not retain scene details. Desolneux *et al.*[88,89] proposed determining abnormal lines by calculating the number of points in the gradient direction and using the non-structural inverse model, which controls the false extraction of line segments and solves the threshold problem of line extraction parameters. Combining the standard line segment test from Desolneux *et al.* and the improved Burns algorithm, Rafael *et al.*[90] proposed the line segment detector (LSD) algorithm, which can automatically and accurately detect line segments from images without requiring manual adjustment of the parameters. On the basis of the previous line segment extraction algorithm, Bauchet and Lafarge[91] proposed a kinetic approach that constructs polygonal contours to express the detailed geometric structure of a 2D image. The core idea of the algorithm is to extend the pre-detected line segments until they intersect the neighboring line segments to better express the geometric information in the image with fewer polygons. Liu *et al.*[92] proposed a novel deep neural architecture, FloorNet, to process data using three neural network branches. PointNet has been adopted to process 3D point cloud data, and the CNN frame can be used in processing 2D point density images to enhance local spatial reasoning. The CNN model then processes the RGB image with full image information, and the neural architecture reconstructs a high-precision vector-graphics floorplan based on low-cost RGBD video data.

The uneven density and high noise in 3D laser point clouds generate considerable challenges for 3D line extraction. In recent years, numerous studies have explored the extraction of line segments using 3D laser point clouds. The general method was first to segment surfaces from point clouds. In it, the intersection area of adjacent planes is fitted to 3D lines.[93] However, the extracted line elements are completely independent, without topological relationships and spatial connections. Xia and Wang[94] developed a similar approach. Others have suggested reconstructing topological 3D models based on extracted 3D lines. Jung *et al.*[95] proposed implicit regularization for reconstructing 3D building models. The building point clouds are first

clustered into planes, and the outer boundaries, intersection lines, and step lines are extracted on the basis of the planes. The binary space partitioning (BSP) technique is then used to build the space topology, and the building rooftop model is regularized using an implicit regularization process. Sui *et al.*[96] proposed an approach that automatically recovers the geometry of urban buildings. The dominant floorplans are extracted by correcting the normal vector and position of the edge points, and each floorplan is then propagated to similar floors. Wang *et al.*[62] developed a novel semantic line framework-based modeling approach using point clouds. The detected line structures are optimized by constructing a conditional generative adversarial net (cGAN) deep learning model.

The 3D indoor model reconstruction based on lines can express indoor scene details but lacks semantic information and spatial relationships. Reconstructed indoor models based on plane and volume features have spatial topology and adjacency, making up for the shortcomings of line-based reconstructions. This approach can generate 3D visualizations and perform spatial analysis and location services.

4.2 Plane-based reconstruction

The main problem of plane-based reconstruction is how to extract high-precision planes to robustly describe indoor scene geometry. The traditional RANSAC algorithm[33] extracts planes by repeatedly selecting random data subsets. Three points are randomly sampled from the point clouds as seed points to fit the initial plane. Under the conditions of fixed interior points and fitting thresholds, it requires abundant samples, from which it selects the largest point set as the best model. The process is continued repeatedly until all point clouds are clustered into respective planes. Although the traditional RANSAC algorithm has a good segmentation effect, several modified approaches have been introduced to improve the segmentation algorithm, such as the efficient RANSAC,[97] the new weighted RANSAC algorithm,[98] and the enhanced RANSAC algorithms.[99] However, these segmentation algorithms still have some shortcomings. For instance, they depend heavily on parameter selection and are considerably affected by point cloud quality. Different point cloud features (e.g., point density, quality, and point spacing) also determine the selection of parameters, and the order of surface detection affects the segmentation results. If the first surface extracted is inaccurate, the subsequent extractions may be seriously affected.

Many research algorithms have focused on the intersection of planar primitives to find appropriate combinations to express building models, which break through the traditional geometric primitive extraction. For example, Monszpart *et al.*[100] developed an effective approach to extract regular arrangements of planes (RAP) from unstructured point clouds in rebuilding anthropogenic scenes. However, the method requires long computing times for the reconstruction of large scenes. Hu *et al.*[101] designed and built a multi-label graph cut model to extract building roof planes. The initial candidate plane set is obtained using superpixel over-segmentation and random sampling technology, and then the best candidate plane set is optimized by combining it with the scene constraint relationship. Finally, the hypervoxel and candidate planes are globally optimized to achieve high-precision plane extraction. Lin *et al.*[102]

proposed a method based on energy minimization to reconstruct the planes consistent with the constraint model. The method implicitly establishes the relationship between planes with the least prior knowledge, which is efficient in plane extraction. Because the extracted high-precision planes have no topological relationship and are only used for visualization, the vector model with topology has to be reconstructed on the basis of the extracted planes and would be visualized using either a regular grid, an irregular triangular network (TIN), or the hybrid model.

Many scholars have studied various methods of reconstructing 3D models. In the common method,[103,104] the roof surface topological relationship is first generated, and the building's 3D model is reconstructed by combining intersection lines and contour lines of surface primitives based on the topological relationship. Lafarge and Alliez[105] proposed to restore the detailed structure of a scene by a hybrid modeling method using surfaces and planes. The final surface model is reconstructed by solving a graph-cut formulated on the 3D Delaunay triangulation of the structured point set in which the tetrahedra are labeled as inside or outside cells. The reconstructed model has rich sharp features, while the number of surface triangles is reduced. Boulch *et al.*[106] provided a method for constructing piecewise planes to reconstruct a 3D model. This method expressed the visible area of the indoor scene with a water-tight polygon mesh, which was conducive to the reconstruction of the indoor occluded area. Chauve *et al.*[107] proposed an approach that employs the adaptive decomposition of 3D space produced from planar primitives. The algorithm uses an optimized water-tight polygon mesh grid to replace the redundant point cloud data, which retains local details, simplifies redundant mesh grids, and is not affected by point cloud noise. Huang *et al.*[108] proposed a new method that inferred the vertical walls directly from the data. With the planar segments of both roofs and walls in place, the faces of the building are hypothesized, and the final model is obtained using an extended hypothesis-and-selection-based polygonal plane reconstruction framework. Bauchet and Lafarge[109] proposed a shape assembling mechanism that was designed from a kinetic data structure for partitioning the space into convex polyhedra. The method can reconstruct a variety of objects and scenes in terms of complexity, size, and acquisition characteristics. While the 3D mesh model provides spatial, semantic, and geometric information, it lacks topological connections for adjacent spaces. The IndoorGML standard model defines a building framework as a solid geometric model expressed by 3D cell elements with rich geometric attributes, topological structure, and semantic information. The reconstruction models are used to express, store, and exchange indoor space information, which is convenient for space analysis, model management, and applications.

4.3 Volume-based reconstruction

Volumetric reconstruction of indoor environments divides a scene into 3D voxels using extracted planes and splits and merges volumetric primitives based on predefined rules. Oesau *et al.*[65] proposed for the first time to reconstruct the indoor model using volume elements. The final model is reconstructed from 3D cell decomposition by categorizing the cells as either empty or solid spaces using a graph-cut algorithm. However, the method can only build inside

and outside partitions, not single rooms. Nan *et al.*[112] and Li *et al.*[113] proposed similar modeling methods. Their methods break through the extraction of traditional geometric primitives, focusing on the intersection of plane primitives to form 3D volume elements to find suitable combinations to express the structural model of a building. In the past five years, many researchers have expanded volumetric reconstruction approaches. Xiao and Furukawa[114] proposed the inverse constructive solid geometry (CSG) method, which uses volume elements to construct geometric solids, adopts regularization constraints to complete structure regularization, and finally completes the reconstruction of large-scale museum models. Ikehata *et al.*[11] proposed constructing indoor structured models guided by structure diagrams in which the nodes are represented as structural elements (e.g., rooms, walls, and objects) and edges represent geometric relationships. This method relies on local geometric relationships and is limited by the assumption of the Manhattan world. Ochmann *et al.*[72] developed an approach that uses RANSAC to extract pieces of planar surfaces; the scanning point cloud of each TLS station is regarded as the room's initial label. Global optimization by multi-label graph cuts is then utilized to reconstruct multi-room indoor model. Given that this method relies on the semantic knowledge of scanning stations, over-segmentation occurs in long corridors. In addition, owing to occlusion and noise, automatic reconstruction is difficult for high-precision indoor models, and manual interaction is still required. Similar to Ochmann's method, Mura *et al.*[115] proposed an effective technique that extracts planes, dividing a scene into 3D polyhedral cells and establishing a BSP tree. The polyhedral cells are then split and merged according to attribute information in the visible point cloud. The final 3D indoor model is reconstructed, overcoming the Manhattan assumption.[11,65,72,113,114] Cui *et al.*[116] developed a method using high-precision line elements and segmented individual rooms as semantic constraints. Global optimization is then carried out to obtain the planar model, and the reconstructed model is generated using the 3D geometric information of the surface elements. This approach can be used to build a complete vector model without local point clouds while ensuring accuracy and model efficiency. Ochmann *et al.*[117] extracted walls and formulated the optimization method to arrange wall entities in reconstructing the structural model, adding manual interactions to improve model accuracy. This model can meet the requirements of BIM, although its main limitations include slanted walls, ceilings, and floors. Li and Wu[118] developed a robust method that embeds multiple relationships into procedural modeling for reconstructing 3D CityCML building models. The hybrid tree of constructive solid geometry and boundary representation (CSG-BRep) was built to divide a building into polyhedral cells based on geometric constraints. The shapes of buildings were selected on the basis of topological-relation constraints. The building models were built using a reconstructing CSG-BRep tree. The specific surfaces of building models are converted into the CityGML format. Chen *et al.*[119] provided a novel framework that reconstructed urban models by exploiting a learned implicit representation as an occupancy indicator for the extraction of an explicit geometry. This report is the first work in which an implicit field is explored for building reconstruction.

Because point clouds are easily affected by many factors, it is still difficult to reconstruct high-precision 3D models automatically. At present, the 3D reconstruction of complex buildings requires manual editing; thus, 3D modeling research has been geared towards reducing human interactions to enable full automation.

5. Discussion

In this paper, we provide a comprehensive literature review of the state-of-the-art techniques for the 3D reconstruction of indoor environments based on laser point clouds. While most of the algorithms generate high model accuracy, several issues still need to be addressed.

(1) Adaptive filtering for indoor point clouds

For complex indoor environments, the existing methods[72–76,116,117] filtered out the clutter of indoor objects using the geometric constraints of point clouds. However, outdoor trees and ground-cluttered points are often included when collecting indoor point cloud data. Relying solely on geometric constraints does not filter out all the noise and still requires manual editing. Therefore, the adaptive point cloud filtering algorithm must be improved to increase point cloud accuracy.

(2) Real 3D model reconstruction

While state-of-the-art techniques can automatically reconstruct regular indoor models, they are not always suitable for the complex structures of indoor environments. For example, the modeling algorithms[110–117] can only express ridge roof structures and cured surfaces with horizontal planes. Future research may involve reconstructing real 3D models using arbitrary geometric shapes (e.g., curved surfaces, planes).

Previous work has primarily focused on reconstructing the vector structure model without texture information. Extracting structural features from images and automatically matching them with vector models should be explored in subsequent studies to improve the creation of real 3D indoor scenes and aid in the construction of realistic 3D China. The various reconstruction algorithms and approaches should be applied and tested in more indoor locations, such as commercial booths, large airports, shopping malls, and subway stations.

(3) Indoor model reconstruction with multi-level details

The major limitation in existing reconstruction methods is that they only reconstruct indoor main frame models (e.g., floors, ceilings, windows, and doors) and lack 3D models for indoor furniture (e.g., tables, chairs, and cupboards). To perform better in indoor location applications, indoor main frame and furniture models with multi-level details should be reconstructed on the basis of point cloud semantic classification results.

6. Conclusions

With developments in sensor technology, various types of laser scanning and optical imaging instruments may now be used for data collection of indoor scenes. The main instrument types are RGBD sensors, TLS, and MLS. Easy-to-use, low-cost MLS equipment has been widely used for data acquisition for large indoor scenes. Therefore, many researchers work on object extraction and 3D model reconstruction based on collected point clouds to carry out urban spatial analysis and other applications. In this paper, we comprehensively analyzed and summarized the current literature on indoor model reconstruction based on point clouds. We first described the indoor modeling standard, which is the foundation of modeling; we introduced the MLS measurement system and discussed the characteristics of collected data. In the second part of the paper, we discussed the classification and extraction of indoor scenes to give

unordered point clouds semantic information, which are used to build 3D vector models with topological relationships and spatial structures. We also comprehensively summarized the existing literature on 3D indoor model reconstruction, which can be divided into line-based, plane-based, and volume-based methods. Line-based reconstruction can express the local details of indoor scenes; however, it lacks semantic information and spatial relationships. Plane-based reconstruction can describe the geometric structure with a redundant 3D mesh grid. Volume-based reconstruction expresses geometric shapes using 3D water-tight volumetric primitives and establishes topological relationships between entities with geometric, semantic, and structural information; however, most existing methods focus on reconstructing main architectural structures, which may be unsuitable for complex indoor scenes.

The last part of the paper presented the issues and trends in indoor model reconstruction. For example, adaptive filtering for indoor point clouds can improve the accuracy of point clouds. Real 3D model reconstruction can be applied to all indoor complex structures and provides texture information. The reconstruction of indoor main frame and furniture models with multi-level details were discussed and summarized.

Acknowledgments

We thank the fund supported by the Beijing Key Laboratory of Urban Spatial Information Engineering (Project No. 20220106) and the Beijing Natural Science Foundation (Project No. 4214069).

References

1 D. R. Li, H. Y. Zhang, and W. J. Jin: Geomatics Inf. Sci. Wuhan Univ. **47** (2022) 1515. http://ch.whu.edu.cn/cn/article/doi/10.13203/j.whugis20220078
2 U. Isikdag, S. Zlatanova, and J. Underwood: Comput. Environ. Urban Syst. **41** (2013) 112. https://doi.org/10.1016/j.compenvurbsys.2013.05.001
3 X. Q. Teng, D. Guo, Y. Guo, and X. L. Zhou: ACM Trans. Sens. Netw. **15** (2019) 1. https://doi.org/10.1145/3216722
4 H. Tashakkori, A. Rajabifard, and M. Kalantari: Build. Environ. **89** (2015) 170. https://doi.org/10.1016/j.buildenv.2015.02.036
5 M. Hashemi and H. A. Karimi: J. Comput. Civ. Eng. **30** (2016). http://d-scholarship.pitt.edu/id/eprint/28202
6 T. Cerovsek: Adv. Eng. Inf. **25** (2011) 224. https://doi.org/10.1016/j.aei.2010.06.003
7 E. Turner, P. Cheng, and A. Zakhor: IEEE J. Sel. Top. Signal Process. **9** (2015) 409. https://doi.org/10.1109/JSTSP.2014.2381153
8 M. M. Mozumdar, R. Tomasi, T. Riccardo, and C. Pastrone: IEEE Sens J. (2015) 5988. https://iris.polito.it/handle/11583/2616954
9 A. Rafiee A, E. Dias, S. Fruijtier, and H. Scholten: Procedia Environ. Sci. **22** (2014) 397. https://doi.org/10.1016/j.proenv.2014.11.037
10 B. S. Yang and Z. Dong: ISPRS J. Photogramm. Remote Sens. **81** (2013) 19. https://doi.org/10.1016/j.isprsjprs.2013.04.002
11 S. Ikehata, H. Yang, and Y. Furukawa: Proc. 2015 IEEE Int. Conf. Computer Vision (IEEE ICCV, 2015) 1323–1324. https://doi.org/10.1109/ICCV.2015.156
12 J. Wang, K. Xu, L. G. Liu, J. J. Cao, S. J. Liu, Z. Y. Z. Yu, and X. F. David Gu: Comput. Graphics Forum **32** (2013) 207. https://doi.org/10.1111/cgf.12187
13 L. D. Vilariño, H. Tran, E, Frías, J. B. Frias, and K. Khoshelham: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLIII-B**4** (2022) 303. https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-303-2022
14 L. T. Losè, A. Spreafico, F. Chiabrando, and F. G. Tonolo: Remote Sens. **14** (2022) 4157. https://doi.org/10.3390/rs14174157
15 OGC, OGC CityGML Encoding Standard, Document No. 12-019, 2012, http://www.opengeospatial.org/standards/citygml (accessed 12 April 2017).

16 BuildingSMART, IFC Standard: http://www.buildingsmart-tech.org/specifications/ifc-overview (accessed 12 April 2017).
17 OGC, OGC IndoorGML, Document No. 14-005r4, 2014, http://www.opengeospatial.org/standards/indoorgml (accessed 12 April 2017).
18 ISO/TC184, Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries; ISO 16739:2013; ISO: Geneva, Switzerland (2013). https://www.iso.org/standard/51622.html
19 J. S. Kim, S. J. Yoo, and K. J. Li: W2GIS. **8470** (2014) 184. https://doi.org/10.1007/978-3-642-55334-9_12
20 OGC, OGC IndoorGML, Document No. 14-005r4, 2014, http://www.opengeospatial.org/standards/indoorgml (accessed 1 April 2019).
21 S. Becker, M. Peter, and D. Fritsch. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XL-3/W**2** (2015) 17. https://doi.org/10.5194/isprsannals-II-3-W4-17-2015
22 O. S. Keeffe, N. Hyland, C. Dore, and S. Brodie: CITA BIM Gathering (2017). https://www.researchgate.net/publication/315786941
23 T. L. Garwood, B. R. Hughes, D. O'Connor, J. K. Calautit, M. R. Oates, and T. Hodgson: Appl. Energy **224** (2018) 527. https://doi.org/10.1016/j.apenergy.2018.04.046
24 K.-J. Li, S.-J. Yoo, and Y. Han: Proc. 21st ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems. (2013) 424–427. https://doi.org/10.1145/2525314.2525451
25 H. G. Ryu, T. Kim, and K. J. Li: ACM (2014) 32. https://doi.org/10.1145/2676528.2676533
26 H. Wang: Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity) 11 (IEEE, 2015) 40. https://doi.org/10.1109/SmartCity.2015.44
27 X. L. Liu: Sci. Surv. Mapp. **44** (2019) 1. https://doi.org/10.16251/j.cnki.1009-2307.2019.06.001
28 L. X. Lin: Research and Implementation of SLAM System for Mobile Robot Based on Fusion IMU (Fuzhou University, 2017) p. 24. https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD201901&filename=1018079795.nh
29 X. Y. Sun: Research on Traversability of Mobile Robot in Indoor Environment Based on SLAM Technology (Harbin Engineering University, 2019) p. 33. https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD201902&filename=1019146764.nh
30 R. G. Huang: Objects Extraction and Level-of-details Representation from Airborne Laser Scanning Point Clouds (Wuhan University, 2017) p. 36. https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CDFDLAST2020&filename=1017192102.nh
31 M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet: ISPRS J. Photogramm. Remote Sens. **105** (2015) 286. https://doi.org/10.1016/j.isprsjprs.2015.01.016
32 R. Zhang. G. Y. Li, M. L. Li, and L. Wang: ISPRS J. Photogramm. Remote Sens. **143** (2018) 85. https://doi.org/10.1016/j.isprsjprs.2018.04.022
33 M. A. Fischler and R. C. Bolles: Commun. ACM **24** (1981) 381. https://doi.org/10.1016/B978-0-08-051581-6.50070-2
34 D. H. Ballard: Pattern Recognit. **13** (1981) 111. https://doi.org/10.1016/0031-3203(81)90009-1
35 V. Sanchez and A. Zakhor: Proc. Int. Conf. Image Processing (2012) 1777. http://www-video.eecs.berkeley.edu/papers/victors/Sanchez_Zakhor_ICIP2012.pdf
36 M. Previtali, L. Díaz-Vilariño, and M. Scaioni: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-**4** (2018) 507. https://doi.org/10.5194/isprs-archives-XLII-4-507-2018
37 L. Díaz-Vilariño, J. M. Sanchez, S. Lagüela, J. Armesto, and K. Khoshelham: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XL-**5** (2014) 203–209. https://doi.org/10.5194/isprsarchives-XL-5-203-2014
38 L. Jean-François, N. Vandapel, D. F. Huber, and M. Hebert: J. Field Rob. **23** (2006) 839. https://doi.org/10.1002/rob.20134
39 R. B. Rusu, N. Blodow, and M. Beetz: Proc. 2009 IEEE Int. Conf. Robotics and Automation (IEEE, 2009). https://doi.org/10.1109/ROBOT.2009.5152473
40 Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. W. Wan: Int. J. Remote Sens. **105** (2013) 63. https://doi.org/10.48550/arXiv.1304.3192
41 Z. Dong, B. S. Yang, Y. Liu, B. J. Li, and Y. F. Zang: ISPRS J. Photogramm. Remote Sens. **130** (2017) 431. https://doi.org/10.1016/j.isprsjprs.2017.06.012
42 H. B. Kim and G. Sohn: Photogramm. Eng. Remote Sens. **79** (2013) 821. https://doi.org/10.14358/PERS.79.9.821
43 H. Guan, J. Li, and M. Chapman: Int. J. Remote Sens. **34** (2013) 5166. https://doi.org/10.1080/01431161.2013.788261
44 J. X. Zhang, X. G. Lin, and X. G. Ning: Remote Sens. **5** (2013) 3749. https://doi.org/10.3390/rs5083749
45 B. Guo, X. F. Huang, F. Zhang, and Y. M. Wang: Acta Geod. Cartogr. Sinica **42** (2013) 715.
46 S. K. Lodha, D. M. Fitzpatrick, and D. P. Helmbold: J. Electron. Imaging (2007). https://doi.org/10.1117/12.714713
47 J. Niemeyer, F. Rottensteiner, and U. Soergel: ISPRS J. Photogramm. Remote Sens. **87** (2014) 152. https://doi.org/10.1016/j.isprsjprs.2013.11.001

48 C. Liu, B. Q. Shi, X. Yang, N. Li, and H. B. Wu: EEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **6** (2013) 2008. https://doi.org/10.1109/JSTARS.2012.2234726
49 Y. T. Yu, J. Li, C. L. Wen, H. Y. Guan, H. Luo, and C. Wang: ISPRS J. Photogramm. Remote Sens. **113** (2016) 106. https://doi.org/10.1016/j.isprsjprs.2016.01.005
50 Z. Zhang, L. Zhang, X. Tong, and W. Zhen: IEEE Trans. Geosci. Remote Sens. **54** (2016) 3309. https://doi.org/10.1109/TGRS.2016.2514508
51 N. Huan, X. G. Lin, and J. X. Zhang: Remote Sens. **9** (2017) 288. https://doi.org/10.3390/rs9030288
52 S. J. Fan, A. W. Zhang, S. X. Hu, and W. D. Sun: Chin. J. Lasers **40** (2013) 221. https://kns.cnki.net/kcms2/article/abstract?v=3uoqIhG8C44YLTlOAiTRKjw8pKedNdX5_mkCYmAjR9xEONtAYoerLXJ6MkOIpdq33HxUuezCUPTLX6Hfe14_ms33m6ZTWi2b&uniplatform=NZKPT
53 I. Armeni, O. Sener, A. R. Zamir, H. L. Jiang, L. Brilakis, M. Fischer, and S. Savarese: Proc.2016 IEEE Conf. Computer Vision and Pattern Recognition (IEEE CVPR, 2016) 1534–1535. https://doi.org/10.1109/CVPR.2016.170
54 G. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan: Support Vector Machine Learning for Interdependent and Structured Output Spaces (Brown University, 2005) p. 71.
55 A. Rottmann, O. M. Mozos, C. Stachniss, and W. Burgard: National Conf. Artificial Intelligence **3** (2005) 130–1311. http://dl.acm.org/doi/10.5555/1619499.1619543
56 R. E. Schapire and Y. Singer: Mach. Learn. **37** (1999) 297. https://doi.org/10.1023/A:1007614523901
57 J. X. Zhang, X. G. Lin, and X. G. Ning: Remote Sens. **5** (2013) 8. https://doi.org/10.3390/rs5083749
58 D. Maturana and S. Scherer: Proc. 2015 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IEEE IROS, 2015) 922. https://doi.org/10.1109/IROS.2015.7353481
59 H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller: Proc. 2015 IEEE Inf. Conf. Computer Vision (IEEE ICCV, 2015) 945. https://doi.org/10.1109/ICCV.2015.114
60 R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas: Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (IEEE CVPR, 2017) 21–26. https://doi.org/10.1109/CVPR.2017.16
61 C. R. Qi, L. Yi, H. Su, and L. J. Guibas: Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (IEEE CVPR, 2017). https://doi.org/10.48550/arXiv.1706.02413
62 C. Wang, S. Hou, C. L. Wen, Z. G, Q. Li, X. T. Sun, and J. Li: ISPRS J. Photogramm. Remote Sens. **143** (2018) 150. https://doi.org/10.1016/j.isprsjprs.2018.03.025
63 L. Q. Zhang and L. Zhang: IEEE Trans. Geosci. Remote Sens. **56** (2018) 1887. https://doi.org/10.1109/TGRS.2017.2769120
64 C. L. Wen, X. T. Sun, J. Li, C. Wang, Y. Guo, and A. Habib: ISPRS J. Photogramm. Remote Sens. **147** (2019) 178. https://doi.org/10.1016/j.isprsjprs.2018.10.007
65 S. Oesau, F. Lafarge, and P. Alliez: ISPRS J. Photogramm. Remote Sens. **90** (2014) 68. https://doi.org/10.1016/j.isprsjprs.2014.02.004
66 S. Ochmann, R. Vock, R. Wessel, M. Tamke, and R. Klein: Inf. Conf. Computer Graphics Theory and Applications (2014) 120. https://ieeexplore.ieee.org/document/7296039/
67 C. Mura, O. Mattausch, A. J. Villanueva, E. Gobbetti, and R. Pajarola: Comput Graph. **44** (2014) 20. https://doi.org/10.1016/j.cag.2014.07.005
68 E. Turner, P. Cheng, and A. Zakhor: IEEE J. Sel. Top. Signal Process. **9** (2015) 409. https://doi.org/10.1109/JSTSP.2014.2381153
69 E. Turner and A. Zakhor: Proc. 2014 IEEE Inf. Conf. Computer Graphics Theory and Applications (IEEE GRAPP, 2014) 22. https://ieeexplore.ieee.org/document/7296026
70 R. S. Wang, L. Xie, and D. Chen: Photogramm. Eng. Remote Sens. **83** (2017) 827. https://doi.org/10.14358/PERS.83.12.827
71 L.D. Vilariño, E. Verbree, S. Zlatanova, and A. Diakité: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-2/W7 (2017) 345. https://doi.org/10.5194/isprs-archives-XLII-2-W7-345-2017
72 S. Ochmann, R. Vock, R. Wessel, M. Tamke, and R. Klein: Comput. Graph. **54** (2016) 94. https://doi.org/10.1016/j.cag.2015.07.008
73 R. Bormann, F. Jordan, W. Z. Li, J. Hampp, and M. Hägele: Proc. 2016 IEEE Inf. Conf. Robotics and Automation (IEEE ICRA, 2016) 1019. https://doi.org/10.1109/ICRA.2016.7487234
74 L. Li, F. Su, F, Yang, H. H. Zhu, D. L. Li, X. K. Zuo, F. Li, Y. Liu, and S. Ying: Remote Sens. **10** (2018) 1281. https://doi.org/10.3390/rs10081281
75 B. F. Zhao, X. H. Hua, K. G. Yu, W. Xuan, X. J. Chen, and W. Y. Tao: Int. J. Comput. Vision **58** (2020) 7890. https://doi.org/10.1109/TGRS.2020.2984943
76 B. Xiong, S. J. Elberink, and G. Vosselman: ISPRS J. Photogramm. Remote Sens. **93** (2014) 227. https://doi.org/10.1016/j.isprsjprs.2014.01.007
77 M. Jarząbekrychard and A. Borkowski: ISPRS J. Photogramm. Remote Sens. **118** (2016) 1. https://doi.org/10.1016/j.isprsjprs.2016.04.005

78 S. Xia and R. Wang: ISPRS J. Photogramm. Remote Sens.**144** (2018) 453. https://doi.org/10.1016/j.isprsjprs.2018.08.009
79 L.Q. Zhang, Z. Q. Li, A. J. Li, and F. Y. Liu: ISPRS J. Photogramm. Remote Sens. **138** (2018) 86. https://doi.org/10.1016/j.isprsjprs.2018.02.008
80 B. S. Yang, R. G. Huang, J. P. Li, M. Tian, W. X. Dai, and R. F. Zhong: Remote Sens. **9** (2017) 14. https://doi.org/10.3390/rs9010014
81 F. Lafarge and C. Mallet: Int. J. Comput. Vision **99** (2012) 69. https://doi.org/10.1007/s11263-012-0517-8
82 K. Karantzalos and N. Paragios: IEEE Trans. Geosci. Remote Sens. **48** (2010) 2283. https://doi.org/10.1109/TGRS.2009.2039220
83 K. Karantzalos, L. D. Vilariño, M. Peter, Z. Z. Kang, and D. Acharya: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-2/W7 (2017) 367. https://doi.org/10.5194/isprs-archives-XLII-2-W7-367-2017
84 D. H. Ballard: Pattern Recognit. **13** (1987) 714. https://doi.org/10.1016/0031-3203(81)90009-1
85 J. Canny: IEEE Trans. Pattern Anal. Mach. Intell. **8** (1986) 679. https://doi.org/10.1109/TPAMI.1986.4767851
86 J. B. Burns, A. R. Hanson, and E. M. Riseman: Comput. Vision **8** (1986) 180. https://doi.org/10.1016/B978-0-08-051581-6.50023-4
87 J. Matas, C. Galambos, and J. Kittler: Comput. Vision Image Understanding **78** (2000) 119. https://doi.org/10.1006/cviu.1999.0831
88 A. Desolneux, L. Moisan, and J. M. Morel: Int. J. Comput. Vision **40** (2000) 7. https://doi.org/10.1023/A:1026593302236
89 A. Desolneux, L. Moisan, and J.M. Morel: From Gestalt Theory to Image Analysis (Springer, New York, 2008) p. 34.
90 R. G. Von Gioi, J. Jakubowicz, J. Morel, and G. Randall: IEEE Trans. Pattern Anal. Mach. Intell. **32** (2010) 722. https://doi.org/10.1109/TPAMI.2008.300
91 J. Bauchet and F. Lafarge: Proc. 2018 IEEE/CVF Conf. Computer Vision Pattern Recognition (IEEE CVPR, 2018) 3146–3154. https://doi.org/10.1109/CVPR.2018.00332
92 C. Liu, J. Wu, and Y. Furukawa: Comput. Vision Pattern Recognit. (CVPR, 2018) 203. https://doi.org/10.48550/arXiv.1804.00090
93 Y. B. Lin, C. Wang, B. L. Chen, D. W. Zai, and J. Li: IEEE Trans. Geosci. Remote Sens. **55** (2017) 4839. https://doi.org/10.1109/TGRS.2016.2639025
94 S. B. Xia and R. S. Wang: IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **12** (2019) 1041. http://dx.doi.org/10.1109/JSTARS.2019.2897987
95 J. Jung, Y. Jwa, and G. Sohn: Sensors **17** (2017) 621. https://doi.org/10.3390/s17030621
96 W. Sui, L. f. Wang, B. Fan, H. F. Xiao, H. Y. Wu, and C. H. Pan : IEEE Trans. Geosci. Remote Sens. **22** (2016) 1261. https://doi.org/10.1109/TVCG.2015.2505296
97 Schnabel, R. Wahl, and R. Klein: Comput. Graphics Forum **26** (2007) 214. https://doi.org/10.1111/j.1467-8659.2007.01016.x
98 B. Xu, W. S. Jiang, J. Shan, J. Zhang, and L. L. Li: Remote Sens. **8** (2016) 5. https://doi.org/10.3390/rs8010005
99 D. Chen, L. Zhang, P. T. Mathiopoulos, and X. F. Huang: IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **7** (2014) 4199. https://doi.org/10.1109/JSTARS.2014.2349003
100 A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra: ACM Trans. Graphics **34** (2015) 1. https://doi.org/10.1145/2766995
101 P. B. Hu, B. S. Yang, Z. Dong, P. F. Yuan, R. G. Huang, H. C. Fan, and X. Sun: Remote Sens. **10** (2018) 10. https://doi.org/10.3390/rs10071127
102 Y.B. Lin, J. L. Li, C. Wang, and Z. G. Chen: Comput. Vision Pattern Recognit. (CVPR 2019). https://doi.org/10.13140/RG.2.2.16048.00007
103 V. Verma, R. Kumar, and S. C. Hsu: Proc. 2006 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE CVPR, 2006) 2213. https://doi.org/10.1109/CVPR.2006.12
104 S. O. Elberink and G. Vosselman: ISPRS J. Photogramm. Remote Sens. **66** (2011) 157. https://doi.org/10.1016/j.isprsjprs.2010.09.009
105 F. Lafarge and P. Alliez: Comput. Graphics Forum **32** (2013) 225. https://doi.org/10.1111/cgf.12042
106 A. Boulch, M. D. La Gorce, and R. Marlet: Comput. Graphics Forum **33** (2014) 55. https://doi.org/10.1111/cgf.12431
107 A. Chauve, P. Labatut, and J. Pons: Proc. 2010 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE CVPR, 2010) 1261. https://doi.org/10.1109/CVPR.2010.5539824
108 J. Huang, J. Stoter, R. Peters, and L. L. Nang: Remote Sens. **14** (2022) 2254. https://doi.org/10.3390/rs14092254
109 J. P. Bauchet and F. Lafarge: ACM Trans. Graphics (TOG) **39** (2020) 1. http://dx.doi.org/10.1145/3376918
110 J. L. Han, M. Q. Rong, H. Q. Jiang, H. M. Liu, and S. H. Shen: ISPRS J. Photogramm. Remote Sens. **177** (2021) 57. https://doi.org/10.1016/j.isprsjprs.2021.04.019

111 T. F. Wang, Q. D. Wang, H. B. Ai, and L. Zhang: Remote Sens. **14** (2022) 4820. https://doi.org/10.3390/rs14194820
112 L. L. Nan and P. Wonka: Inf. Conf. Computer Vision (ICCV, 2017) 2371. https://doi.org/10.1109/ICCV.2017.258
113 M. L. Li, P. Wonka, and L. L. Nan: European Conf. Computer Vision (ECCV, 2016) 54. https://doi.org/10.1007/978-3-319-46493-0_4
114 J. X. Xiao and Y. Furukawa: Int. J. Comput. Vision **110** (2014) 243. https://doi.org/10.1007/s11263-014-0711-y
115 C. Mura, O. Mattausch, and R. Pajarola: Comput. Graphics Forum **35** (2016) 179. https://doi.org/10.1111/cgf.13015
116 Y. Cui, Q. Q. Li, and Z. Dong: Remote Sens. **11** (2019) 2262. https://doi.org/10.3390/rs11192262
117 S. Ochmann, R. Vock, R. Wessel, and R. Klein: ISPRS J. Photogramm. Remote Sens. **54** (2019) 251. https://doi.org/10.1016/j.cag.2015.07.008
118 Y. Li and B. Wu: Remote Sens. **13** (2021) 129. https://doi.org/10.3390/rs13010129
119 Z. Y. Chen, H. Ledoux, S. Khademi, and L. L. Nan: ISPRS J. Photogramm. Remote Sens. **194** (2022) 58. https://doi.org/10.1016/j.isprsjprs.2022.09.017

About the Authors

Yang Cui received her B.S. and M.S. degrees from Liaoning Technical University, China, in 2013 and 2016, respectively, and her Ph.D. degree from Shenzhen University, China, in 2020. Since 2020, she has been working at the Beijing Institute of Survey and Mapping. Since 2022, she has been a senior engineer. Her research interests include laser scanning sensors, computer graphics, 3D automatic modeling, and their applications in urban environments. (cuiyangmailbox@163.com)

Bogang Yang received his Ph.D. degree from Beijing Forestry University, China, in 2006, and he was engaged in postdoctoral work at Peking University from 2007 to 2009. Since 1983, he has been working at Beijing Institute of Survey and Mapping, and since 2004, he has been a professor-level senior engineer. He is the Director of the Beijing Key Laboratory of Urban Spatial Information Engineering. His research interests are in surveying and mapping technologies, smart city construction, and urban spatial analysis. (bogangy@126.com)

Peng Liu received his B.S. and M.S. degrees from Liaoning Technical University, China, in 1998 and 2001, respectively. Since 2001, he has been working at the Beijing Institute of Survey and Mapping. Since 2011, he has been a professor-level senior engineer. His research interests are in sensors, urban informatics, and smart cities. (liupeng@bism.cn)

Lingyan Kong received his B.S. and M.S. degrees from Beijing Normal University, China, in 2000 and 2003, respectively. Since 2003, he has been working at Beijing Institute of Survey and Mapping. Since 2016, he has been a professor-level senior engineer. His research interests are in geographic information systems, smart city, and urban governance. (kongly@163.com)