Urban 3D modeling using mobile laser scanning: a review

Cheng WANG*, Chenglu WEN, Yudi DAI, Shangshu YU, Minghao LIU

Fujian Key Lab. On Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen 361005, China

* Corresponding author: cwang@xmu.edu.cn
Received: 6 March 2020 Accepted: 8 May 2020

Abstract Mobile laser scanning (MLS) systems mainly comprise laser scanners and mobile mapping platforms. Typical MLS systems can acquire three-dimensional point clouds with 1–10 cm point spacings at a normal driving or walking speed in streets or indoor environments. The efficiency and stability of these systems make them extremely useful for application in three-dimensional urban modeling. This paper reviews the latest advances of the LiDAR-based mobile mapping system (MMS) point cloud in the field of 3D modeling, including LiDAR simultaneous localization and mapping, point cloud registration, feature extraction, object extraction, semantic segmentation, and processing using deep learning. Furthermore, typical urban modeling applications based on MMS are also discussed.

Keywords 3D Modeling; MMS; LiDAR; Urban

1 Introduction

Urban 3D modeling is used to establish a 2.5D or 3D digital representation of the earth's surface and the objects present on it, such as buildings, roads, vegetation, and other manmade attributes in urban areas. There are three major categories of this approach: (1) conventional geodetical mapping techniques, (2) approaches based on 2D image photogrammetry, and (3) approaches based on 3D measurements, such as laser scanning. Although the data acquired are dense, and precision is high, conventional geodetical mapping techniques are time-consuming and show poor mobility. Therefore, this method is not suitable for large-scale mobile mapping tasks. 2D image photogrammetry methods are easy to set up and low-cost, and various deep learning methods can be conveniently integrated with these methods to extract and visualize semantic information. However, these methods are overly sensitive to environmental changes, such as ambient light, weather, and darkness. Moreover, the 3D model built by only using images cannot be directly used for navigation. The modeling methods based on LiDAR are of high precision, exhibit high reliability and are not easily affected by changes in the environment. Unlike the 3D models built based on images, the 3D model built using LiDAR has applications in the field of autonomous driving. Therefore, the methods discussed in this paper are mainly based on LiDAR or other 3D measurement equipment.

The task of large urban area 3D modeling demands high efficiency in data acquisition. The MLS system comprises an MMS equipped with laser scanners. The technology of MMS facilitates efficient 3D modeling. Mobile mapping is a system technology that enables the installation of photogrammetry sensors...
on mobile platforms with high-precision, high-efficiency georeferencing capabilities. MMSs can efficiently collect georeferenced three-dimensional measurements of the environment when the platform is in motion. Successful MMSs include the VISAT\cite{1} system from the University of Calgary, Canada, the GPSVan\cite{2} developed by the Ohio State University, and the LD2000\cite{3} that is developed by the Wuhan University. At present, a typical MLS can collect 1 million points per second, which means that it can cover a road and its surrounding surface with a point density of 2000 points per square meter, 1–10 cm point spacings, and moving speed of 10–110 km/h.

MLS point clouds are large-volume and have heavy redundancy and irregular distributions\cite{4}. In addition, the quality of the point cloud is degraded if noise and occlusion are present. Consequently, MLS point cloud processing is a challenging task in urban 3D modeling. Standard point cloud processing involves aspects such as feature point extraction, matching, and registration, object detection, semantic segmentation, and simultaneous localization and mapping (SLAM).

This paper presents a review of the MLS solutions in urban 3D modeling, as depicted in Figure 1. The rest of the paper is organized as follows. Section 2 reviews the MLS technology. Section 3 provides the discussion on the processing of MLS point cloud, and Section 4 presents the typical urban modeling applications based on MLS.

![Figure 1](image-url)  
A introduction of the logic among all components of the urban 3D modeling using MLS.

2 MLS system

In this section, we first introduce the system design and the important sensors of the MLS. Among the MLS sensors, the global navigation satellite system (GNSS) and inertial measurement unit (IMU) are the key components of MLS for navigation. However, LiDAR plays a significant role in GNSS-denied environments.

2.1 System design

The MLS system is an MMS equipped with laser scanners. As shown in Figure 2, MLS systems usually consist of GNSS receivers, laser scanners, digital cameras, IMUs, and other devices. Synchronization of
the data from the abovementioned sensors to a time frame of reference is achieved via precise timestamping. Methods for calculating ground coordinates for objects from the laser scanning system have been well reported by various studies. One such method involves the combining of the measurements obtained from the integrated GNSS/INS navigation system, laser scanner, and sensor calibration parameters.

### 2.2 GNSS and IMU

The MLS systems perform the survey by the ground vehicles. In MLS, the navigation system, which includes a global navigation satellite system (GNSS) and an inertial measurement unit (IMU), provides the vehicle's trajectory and attitude for generating the georeferenced 3D point clouds. The relative precision of the point can be lower than in the order of a subcentimeter, and its absolute accuracy depends on the above GNSS-IMU-integrated navigation solution.

#### 2.2.1 GNSS/IMU integrated navigation

GNSS provides geographical position and velocity data of a GNSS receiver antenna by employing a constellation of orbiting satellites. The most popular GNSS systems include the global positioning system (GPS) (United States), global navigation satellite system (GLONASS) (Russia), COMPASS/BeiDou navigation system (BDS) (China), and Galileo (European Union).

The position measurement is computed by triangulating the satellite signals within a clear view of the receiver antenna. Generally, there must be four satellites visible for a positional fix, as shown in Figure 4, and the accuracy of the GNSS ideally increases as more satellites become available. However, there are some common error sources, for example, receiver noise, atmospheric delays, multipath, and satellite clock timing, which result in the GNSS receivers usually having a positioning accuracy of 1–2m. Obstructions such as buildings or trees can block the satellite signal, which results in unreliable navigation. Some methods such as post-processing, precise point positioning (PPP), and real-time kinematic (RTK) have been proposed to improve the accuracy of GNSS.

An inertial navigation system (INS) computes the relative position of an object over time using rotation
and acceleration measurements from an IMU, which can measure the relative movement in 3D space. An IMU contains six complementary sensors, which are arrayed on three orthogonal axes. An accelerometer and a gyroscope, which measure linear and rotational acceleration, respectively, are coupled on each of the three axes. Based on the linear acceleration and rotational acceleration measurements, the INS can calculate the position and velocity for all the three axes. In addition to this, the IMU can provide an angular solution, which can be translated into a local attitude (roll, pitch, and azimuth) solution in INS[8].

When using an IMU for navigating in a 3D space, hundreds/thousands of samples are acquired per second, and consequently, many errors get accumulated. Thus, without an external reference, an uncorrected INS system can quickly drift from the true position. The INS can estimate the error of the IMU measurements using a mathematical filter if an external reference is provided by the GNSS. The GNSS provides an absolute set of coordinates that are used as the starting points and continuous positions and velocities for updating the INS filter estimates. The integration of the GNSS and INS therefore enhances the overall performance in providing a more powerful navigation solution. For example, the INS system can be effectively used for navigating for longer periods when the GNSS is unreliable due to signal obstructions.

2.3 Laser scanner

In the MLS system, the point cloud is generated by the laser scanners, also known as LiDAR, which can estimate the distance of an object by emitting laser lights and measuring the time required by the light to returns to the sensor. 3D LiDARs can be used for plane mapping, obstacle avoidance during navigation, and urban area modeling. LiDARs are mainly used in outdoor environments, especially in fields such as geodesy, meteorology, geology, and military.

Usually, the optical pulse or the wave can only be used to measure the distance in a specific direction. LiDARs normally include oscillating mirrors, which can perform scanning in multiple directions. According to the specific oscillation mechanism, LiDARs can scan the surrounding environment in both 2D and 3D.

A rotating LiDAR has a 360° view. With each rotation, it can scan points along a cone originating from the sensor, thereby resulting in a single circular scan line. This cone angle is varied by a predefined amount after each full rotation, with a maximum absolute angle such that the sensor is unable to scan the area directly above or under it.

The Velodyne VLP-16 and HDL-32 are the most affordable commercial multi-beam sensors, and their main specifications are provided in Table 1. VLP-16 is more compact and lightweight as compared to HDL-32, which has a relatively higher cost and better scanning effect.

The main advantages of LiDAR are as follows:

(1) Different types of LiDARs can provide different measurement ranges, from a few centimeters to more than 100 meters. Therefore, it can be used in both indoor and outdoor environments.

(2) The horizontal aperture of LiDAR is usually between 90 and 360 degrees.

(3) The angle resolution of LiDAR is usually less than 1 degree.

(4) The measurement error of LiDAR is low and is a usually constant (for a short distance) or is linear with the measuring distance.

(5) LiDAR can provide both medium and high sampling rates, which is crucial for application in a dynamic environment; the sampling rate is usually adjustable from 10Hz to 20Hz.

The main disadvantage of LiDAR is that it is expensive. In addition, LiDAR’s power consumption is
high (more than ten times of the camera), and its scanning performance declines in the presence of fog, rainstorm, or dust.

2.3.1 SLAM-based navigation

The visibility of the ground receiver to the GNSS satellite is the main reason for the high accuracy achieved in GNSS positioning. However, GNSS signals are vulnerable to external interference and can lead to failure when the platform is in a complex environment such as in the case of high-rise buildings, steep slopes, or indoor environments. The accuracy of positioning will be also be consequently degraded. Therefore, alternative techniques must be developed to address the aforementioned issue. SLAM is arguably one of the most important algorithms used in robotics and 3D vision, which is also applicable in GNSS-denied environments.

LiDAR SLAM

LiDAR has been an important sensor in robot navigation for obstacle avoidance and path planning. Meanwhile, LiDAR-based SLAM methods, such as feature-based registration method\textsuperscript{10}, iterative closest point\textsuperscript{11} (ICP), and normal distribution transform\textsuperscript{12} (NDT), have been proposed for estimating the transformation between two sets of overlapped point clouds.

A feature-based registration method is commonly used for initial transformation estimation between the two point clouds. This type of method first finds the key features in the two point clouds. Next, it computes the descriptors for these key features to perform image matching. Finally, it calculates the transformation matrix between the corresponding key features.

ICP converges to a local minima by minimizing the squared error, and can therefore be categorized as: point-to-point, point-to-plane, and plane-to-plane ICP. For point-to-point ICP, the correspondence pairs are built by pairing each point in the first point cloud with the closest point in the second point cloud. Subsequently, in each correspondence pair, the transformation between the two point clouds is computed by minimizing the sum of the squared distance between the two points.

NDT employs statistical models of the points to estimate the possible alignment between the two point clouds.

LiDAR SLAM may fail to effectively work in situations where sparseness of LiDAR point clouds exists. The integration of camera and LiDAR can improve the performance\textsuperscript{10,13}. Camera-based visual odometry can provide initial estimation for ICP and correct the motion distortion of the point clouds caused by the different receiving times of the points. Scherer et al. estimated the ego-motion of the system by the integration of images and IMU data, and then refined the ego-motion estimation by LiDAR data\textsuperscript{14}.

Table 1 Manuf acturer specifications (VLP-16 and HDL-32 sensors)

<table>
<thead>
<tr>
<th></th>
<th>VLP-16</th>
<th>HDL-32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laser/detector pairs</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>Range</td>
<td>1m to 100m</td>
<td>1m to 70m</td>
</tr>
<tr>
<td>Accuracy</td>
<td>±3cm</td>
<td>±2cm</td>
</tr>
<tr>
<td>Data</td>
<td>Distance/calibrated reflectivity</td>
<td>Distance/calibrated reflectivity</td>
</tr>
<tr>
<td>Data rate</td>
<td>300000 points/s</td>
<td>700000 points/s</td>
</tr>
<tr>
<td>Vertical FOV</td>
<td>30°: [−15°, +15°]</td>
<td>41.3°: [−30.67°, +10.67°]</td>
</tr>
<tr>
<td>Vertical Resolution</td>
<td>2.0°</td>
<td>1.33°</td>
</tr>
<tr>
<td>Horizontal FOV</td>
<td>360°</td>
<td>360°</td>
</tr>
<tr>
<td>Horizontal Resolution</td>
<td>0.1° to 0.4° (programmable)</td>
<td>0.08° to 0.35° (programmable)</td>
</tr>
<tr>
<td>Size</td>
<td>103mm×72mm</td>
<td>85.3mm×149.9mm</td>
</tr>
<tr>
<td>Weight</td>
<td>0.83Kg</td>
<td>1.3Kg</td>
</tr>
</tbody>
</table>
Droeschel et al. developed a 3D multi-resolution map for robot navigation by fusing LiDAR data with a 3D map\(^{15}\).

As the sensor scans its surroundings, the platform may move and rotate. Let us consider an extremum example as follows: if the platform counter-rotates at the same angular velocity as its rotating-scanner, all the points will be located on the same vertical plane in the world frame. The full scan from the sensor frame can't be accurately map to the world frame with a single affine transform. This is because each point is taken at a different moment in time, and thus, each point has its own frame in relation to the world frame. Precise robot poses at various times during the scanning process allows for the correction of distortion by associating a different affine mapping from the sensor frame to world frame for each group of points acquired.

**Multi-sensor SLAM**

Monocular visual odometry (VO) has been well explored in this area for several years, and there are some robust and mature solutions that exist, such as MonoSLAM\(^{16}\), ORB-SLAM\(^{17}\), and SVO\(^{18}\). For example, graph-based optimization and loop closure can be applied for visual SLAM methods, such as RTAB-Map method\(^{19,20}\). However, visual SLAM methods are limited under dynamic weather and insufficient lighting conditions. To improve accuracy and robustness under such conditions, some studies have proposed the combination of additional cameras to this system. There are also some visual solutions that have been integrated with the inertial measurement unit (IMU), such as Geneva's work\(^{21}\), AbolDeepIO\(^{22}\), VINS\(^{23}\), as well as to its advanced version—VINS-Fusion\(^{24}\). VINS-Fusion fuses local states (camera, IMU, LiDAR, etc.) with global sensors (GPS, magnetometer, barometer, etc.) and achieves globally drift-free and locally accurate pose estimation. The fusion of local estimations from the existing VO/VIO approaches and global sensors is depicted in Figure 5.

![Figure 5 An illustration of VINS-Fusion\(^{24}\).](image)

Owing to the demands of high accuracy maps in autonomous driving, robustness of methods in dynamic environments, and dense point cloud data, LiDAR-based SLAM has always been a widely investigated technique in the field of autonomous driving. Furthermore, the price of multi-beam LiDARs has dropped significantly in the recent years. Several researchers have also investigated the integration of LiDARs with other sensors (for example, cameras, IMU, etc.) (Figure 6).

![Figure 6 Block diagram of the V-LOAM\(^{25}\) system.](image)
A straight-forward solution for the integration of lasers with cameras is the use of the VO result as an initial guess for the ICP or GICP pipeline, as was demonstrated in the work of Pandey and Zhang. Zhang combined visual odometry and LiDAR odometry for the mapping tasks. Some methods also exist that treat color information as the fourth channel of a 3D point for the subsequent ICP pipelines. Another way to fuse the information from cameras and lasers is by using the LiDAR information to enhance the visual features. Graeter et al. proposed LIMO, which can track camera features and estimate camera motion based on the LiDAR point clouds.

Furthermore, there are studies that have focused on LiDAR-IMU fusion, which is a topic that remains to be sufficiently investigated. Ye et al. introduced a tightly coupled LiDAR-IMU fusion method by jointly minimizing the cost derived from the LiDAR and IMU measurements. Geneva et al. presented the LIPS, which is a singularity free plane factor that leverages the closest point plane, by fusing with IMU in a graph-based optimization framework.

3 Processing of point cloud data

Several studies have focused on the processing of point cloud data. There are five categories of these methods, namely, feature extraction, registration, completion, semantic segmentation, and object/instance extraction. We will discuss these methods in detail in this section.

3.1 Point clouds completion

By increasing the popularity of data acquisition devices, such as laser scanners and RGB-D cameras, even complicated objects can be digitized with impressive accuracy. Given different digitizing technologies, there are still several limitations pertaining to environmental conditions, inter-object occlusion, and sensor capabilities that constrain the full effectiveness of scene depth captured by a mobile laser scanner. Incomplete data will bring uncertainty to subsequent processing. To avoid this, we must have a corresponding complete version of the data. For simple data acquisition, we can re-scan to obtain more data. However, sometimes, obtaining a full version of the 3D data by re-scanning can be challenging due to occlusion caused by objects or inaccessibility of the scanning to the observation area by the scanning device, and therefore, we need to complete the data manually or automatically.

This has created an area of completing the missing 3D information of an MLS data or other forms of 3D data. Existing methods for 3D data completion are categorized into geometry-based, data-driven, and learning-based approaches.

3.1.1 Geometry-based approaches

Geometry-based approaches estimate shapes using geometric cues from the input, where the missing regions are inferred from the observation areas. These approaches are effective in completing small holes and regular shapes within a reasonable time cost.

Surface reconstruction approaches

Many previous works on surface reconstruction have reported the generation of smooth interpolations to fill holes in locally incomplete scans. Their superior performance of surface reconstruction always relies on the type of environment the MLS data represents. The most common scenario is the traffic scene, which is also easy to reconstruct road surface.

A road surface reconstruction method was proposed to process the raw data and produce a 3D model while ensuring that the details are preserved. Another method was used to recognize the curbs while
reconstructing missing information caused by occlusion; it also reconstructed road surfaces and pavements with centimeter precision while reconstructing the missing information of the curbs. For indoor mapping, an incremental surface growing-based method\cite{35} was proposed to create the triangular mesh and fill the holes with sizeable noisy LiDAR data from an indoor environment.

Some other methods were proposed to reconstruct the surface in various ways; some reconstructed the operators for surface approximation\cite{36,37}; others provided algorithms to fill the holes on the surface\cite{38-40}. Road surface reconstruction approaches, however, fail when the surface of the object is severely damaged due to occlusion.

**Symmetry-based approaches**

Symmetry is a common characteristic of real-world objects like buildings. Symmetry is commonly exploited to analyze and process the computational representations of most 3D objects from the real world. Symmetry-based methods identify repeating structures and symmetry axes to duplicate parts to incomplete regions.

Some studies have focused on small objects, such as household objects\cite{41-43}, and others on large-sized objects, such as buildings\cite{44}. Most of these objects are not symmetrical, and only parts of these objects are symmetrical. For these kinds of 3D objects, some methods\cite{45-47} have been proposed to implement symmetry-based completion of the entire object. Thrun et al. described a technique for segmenting objects into parts characterized by different symmetries and used these parts to map the partial 3D shape model into an occluded space\cite{45}. Another general approach\cite{46} was proposed to efficiently extract a compact representation of the Euclidean symmetries of the object to capture essential and high-level information about the object, and in turn, enable further processing, including shape symmetrization and segmentation.

**Regularity-based approaches**

Regular geometric structures are ubiquitous in both natural and manmade worlds, in which repeated geometries play an essential role in the recognition and understanding of the world—as many objects are characterized by such patterns.

Regularity-based completion approaches are widely used for completing 3D building models\cite{48-50} as they are one of the most regular objects in the real world. These methods can complete data using various regularity principles, such as by performing Fourier analysis of each scanline to fill the holes and generating meshes\cite{49}, or merely exploiting the large-scale repetitions found in building scans and subsequently using it to remodel the input\cite{48}.

### 3.1.2 Data-driven approaches

Considering that the generation of perfectly precise and complete data could be challenging, data-driven approaches can handle complete shapes by matching the incomplete object with the template models present in template shape databases. The main rationale of this type of approaches is to retrieve a 3D model that is most similar to that of the input query, which can be done in the case of single objects, such as vehicles and furniture, but not when large objects such as buildings, are involved.

**Retrieval-based approaches of replacement with completed object**

Most of the retrieval-based methods can retrieve the complete shape from a database and use it to directly replace the incomplete one\cite{51-53}.

Two methods\cite{51,52} along with the datasets of thousands of models, are provided for 3D shape retrieval, where the defective scanned data are replaced with the retrieved model. The replacement is done via a 3D indoor reconstruction by classifying each object in the scene and replacing the mutilated object with a complete one from the dataset\cite{53}.
Approaches of assembling parts to obtain the complete shape

Some studies also deem that simply replacing the incomplete 3D object with a complete one can lead to inaccuracies in the final reconstructed model, and they suggest that the 3D shape be completed by retrieving and assembling all the object parts\cite{51,54-56}.

3.2 Feature extraction (line, plan, and supervoxel)

The method of efficiently processing massive and complex point cloud data is a challenge. There are two main methods for this purpose. The first method projects the high-density point cloud data into 2D images, and then implements the image processing techniques\cite{67-69}. In the other methods, the point cloud data are processed in the feature space. Line and plane features contain abundant geometric information of point clouds, especially in artificial environments. These features are generally parallel, orthogonal, or coplanar, which can effectively reduce the complexity of point clouds without losing their main geometric information. Therefore, line and plane extraction are widely used in target recognition\cite{60}, point cloud registration\cite{61}, reconstruction\cite{10,11}, and so on.

Line extraction can be classified into two categories. In the first category, the real-world object is projected into 2D images, then LSD\cite{62} or EDLines\cite{63} are used to extract lines from these images, and finally these lines are back-projected into a 3D space to obtain the 3D lines. Jain et al. extracted the straight lines of a scene from the multi-view images of the same scene, and then returned these lines to a 3D space based on the visual information, and finally obtained the 3D straight lines\cite{64}. Lin et al. proposed a line-half-planes (LHP) model to extract 2D lines by projecting 3D point clouds onto multi-view images and then obtaining 3D lines by projecting 2D lines back into the 3D space\cite{65}. The advantage of projecting point clouds into images is that the existing 2D line extraction algorithms can be fully utilized. Additionally, the disadvantage is that large-scale point clouds take a considerable amount of time to process. There are several studies that can directly extract line features on point clouds. Daniels et al. used robust moving least-squares method to fit the surface locally and then calculated a set of smooth curves aligned along the edge to identify the line features in the point cloud; finally, they were able to produce a set of complete smooth feature curves\cite{66}. Kim et al. used a moving least-squares approximation to estimate the local curvatures and their derivatives at a point using an approximating surface\cite{66}. Lin et al. presented a facet segmentation-based line segmentation method, which can be directly used on the point cloud\cite{67}. This method can extract complete and more precise line segments compared to the abovementioned methods\cite{68}.

Several different algorithms have been proposed for plane extraction from 3D point clouds. Traditional plane extraction techniques can be generally categorized into region-growing\cite{68-70}, Hough transform\cite{71,72}, and model-fitting methods\cite{73-75}. However, these methods do not fully employ the geometric constraints of the point clouds. Lin et al. proposed a method based on energy minimal to reconstruct the planes, thereby leveraging a constraint model that requires minimal prior knowledge to implicitly establish relationships among the planes\cite{69}. To balance between high-accuracy and high-efficiency, El-Sayed et al. proposed a plane detection method based on octree-balanced density down-sampling and adaptive plane extraction\cite{77}. Nguyen et al. utilized scan profile patterns and the planarity values between different neighboring scan profiles to detect and segment planar features in sparse and heterogeneous MLS point clouds\cite{78}. Kwon et al. proposed a plane extraction algorithm involving various stages including decomposition, expansion and merging; furthermore, the algorithm works effectively even in the case of low-density point clouds as the expansion stage is included between the conventional decomposing and merging stages\cite{79}.

Line and plane extraction are based on point-wise processing. To process point cloud faster, supervoxels
were proposed. Supervoxels, an analog of superpixels in the 3D domain, is a promising alternative by which redundancy in the information can be markedly reduced, thereby enabling computational efficiency for fully automatic operation, with a minimal loss of information. Using supervoxels, a point cloud is divided into several patches and then processed in a patch-wise manner, rather than a point-wise manner. Voxel cloud connectivity segmentation (VCCS) is a commonly used supervoxel generation method. Lin et al. formalized the supervoxel segmentation problem as a subset selection problem optimized efficiently by a heuristic method utilizing local information for each point. Zai et al. proposed an improved supervoxel algorithm to generate supervoxels with adaptive sizes inspired by the point cloud segmentation method. Wang et al. proposed an efficient 3D object detection method by integrating the supervoxel algorithm with a Hough forest framework.

3.3 Matching and registration

3D point cloud registration, a key issue in 3D data processing, is usually considered as rigid registration and urban 3D reconstruction, which can be solved using transforming parameters with six degrees of freedom (6DoF). To this end, numerous related methods have been proposed having a variety of applications.

The ICP algorithm alternates between the estimation of the point correspondence and that of the transformation matrix (Figure 7). Many variations of this method have been proposed in the literature. However, ICP suffers from certain limitations, such as (1) explicit estimation of closest point correspondences, which leads to quadratic complexity scaling of the points, (2) sensitivity to initialization, and (3) difficulty of integration with deep learning frameworks owing to the issue of differentiability. The abovementioned methods cannot guarantee the global optimality of the solutions. Therefore, several researchers have focused on optimization algorithms to estimate relative transformations.

Pioneering studies on handcrafted 3D feature descriptors were mostly inspired by their 2D counterparts. Many approaches including SHOT, RoPS, TOLDI, FPFH, and ACOV estimate the unique local reference frame (LRF), which is not robust to noise. Therefore, MLS large-scale point clouds are not ideal for adoption. With the development of deep-learning methods in the domain of geometric representation of 3D data, learned-based 3D local feature descriptors are being often applied to point cloud registration. Some studies have focused on learning local features with robustness and then extracting matching correspondences using strategies such as RANSAC; finally, the extracting correspondences are used to estimate the transformation matrix. Some other studies have focused on constructing a local feature learning method that is end-to-end and network based to achieve point cloud registration. Whereas, other studies have proposed the use of the global information to regress rotational transformation matrices and translation vectors.

Following RANSAC, Aiger et al. proposed a randomized alignment approach, which uses planar congruent sets to compute optimal global rigid transformation. However, these RANSAC-like methods are point-level operations, which may easily be sub-optimal when computing transformation.
3.4 Semantic labeling and segmentation

Semantic labeling and segmentation of point cloud entails understanding and recognizing the meaningful entities in a scene by assigning each point to an entity. Examples of entities in an urban scene may include sky, buildings, facades, roads, windows, doors, poles, and pedestrians. In this section, we review the classification and semantic segmentation methods that focus on terrestrial laser scanning (TLS) and MLS of the point cloud. Notably, comprehensive literature can be found on terrestrial mobile laser scan processing, covering semantic segmentation, feature extraction, and object recognition in Che's work\(^{[113]}\).

3.4.1 Feature-based methods

Feature-based methods label each point in the point cloud by extracting and joining the features to form a vector. A trained classifier is then employed to perform labelling. Hackel et al. reduced the computation time and also addressed the challenge of varying densities in a point cloud by handling the strong varying densities of TLS points\(^{[114]}\). The TLS and MLS point clouds are constituted of millions of points and as such, labeling each point is computationally intensive. Weinmann et al. improved classification results by using five different definitions of the neighborhood when selecting the optimized features in the feature extraction process\(^{[115]}\). Hu et al. used gridded segmentation to address the computational challenges and achieved good segmentation results without relying on computationally expensive representations of the scene\(^{[116]}\). Segmentation and classification were simultaneously conducted in Zhao's work\(^{[117]}\), where each segment was classified using its geometric properties and the homogeneity of each segment was evaluated using the object class. Spatial smoothing of neighboring elements can lead to improvements in the segmentation results. Probabilistic models, for example, the Markov random field (MRF) and conditional random field (CRF), are used for this purpose. Lu et al. assigned semantic labels to each point by calculating the node potentials and edge potentials using the distance between the points, and the contextual relationships between the points were given by the MRF\(^{[118]}\). Another network\(^{[119]}\) was proposed that employed CRFs to propagate contextual information between neighboring entities. They performed discrete, multi-label classification by learning high-dimensional parameters of CRFs, and the higher-order models were found to be robust in preserving salient labels.

Previously, handcrafted features were primarily used for visualization tasks. The handcrafted features were designed to be in variance to certain transformations; however, they are usually geared toward a specific task and require a considerable amount of human intervention. Feature-based methods in this category heavily relied on handcrafted features that have since been outperformed by semantic features.

3.4.2 Deep learning methods

Deep learning techniques learn features that can be applied in multiple tasks and the learning happens in an end-to-end manner, thereby requiring little human intervention. Convolutional neural networks (CNNs) have proven to be effective in data formats that have regular formats like the grid-like structure of pixels in 2D images. However, deploying CNNs directly on point clouds is challenging. Hence, it is an active and ongoing research area. Point clouds are irregular and as such, the segmentation of points has taken the following directions.

In general, deep learning methods in 3D can be categorized into Volumetric CNN, Multiview CNN, and Point-based methods corresponding to the popular 3D data representations of Volumetric, Multiview images, and Point Cloud, respectively.

**Volumetric CNNs**

Volumetric CNNs operate on volumetric data, which is often represented as a 3D binary voxel grid. 3D-
ShapeNets represent a 3D shapes on shapeNets as a probability distribution of binary variables on a 3D grid. The voxel grid makes it possible to apply 3D convolution operation. In Charles's work, they proposed a model for predicting objects from partial sub-volumes by addressing the issue of overfitting using auxiliary training tasks, and they also proposed another model for convolving the 3D shapes with the anisotropic probing kernel. In addition, the VoxelNet was used as a 3D CNN on voxels for real-time object recognition. VoxelNet incorporates normal vectors of the object surfaces to the voxels to improve the discrimination capability. Although techniques based on volumetric CNNs have reliable performances, they suffer from limitations such as the introduction of quantization artifacts, high memory consumption and computational cost owing to the sparsity of the occupancy grid.

**Multi-view CNNs**

Projection of 3D point cloud to the 2D grid is done to leverage the high performance of 2D segmentation algorithms by rendering the 3D data in 2D. These techniques are based on the traditional CNN that operates on 2D images. These techniques can map the 3D object into a collection of 2D images of the object taken from different angles. Compared to their volumetric counterparts, multi-view CNNs have superior performance as multi-view images contain richer information as compared to their 3D voxels counterparts. Su et al. conducted the first study on multi-view CNNs for object recognition and achieved state-of-the-art accuracy. Su et al. conducted the first study on multi-view CNNs for object recognition and achieved state-of-the-art accuracy. Leng et al. proposed a stacked local convolutional autoencoder (SLCAE) for the 3D object retrieval task. In Tostberg's work, 3D point clouds were projected onto a 2D image, and the image was semantically segmented using a 2D semantic classifier. This operation leads to a loss of valuable information in the transformation of 3D to 2D because the former is richer in content (or, depth information). In Wu's work, the spherical projection was used in a pipeline containing 2D CNNs and CRFs to project the point clouds into a 2D grid. The CNN of the pipeline performs segmentation and the CRF refines it. "Auto-labeling" is an approach of transferring high-quality image-based semantic segmentation from reference cameras to point clouds. A fully convolutional neural (FCN) network was used in pixel-wise semantic segmentation of roads from the top view images of the point cloud. Lawin et al. employed a similar approach and even went further to investigate the significance of the surface normal, depth, and color on the architecture. The main drawback of the aforementioned methods lies in the information loss that occurs during the 3D-to-2D projection process.

**Point-based**

Direct processing of the 3D point cloud is also very popular. Point-based methods were pioneered by PointNet. Because a point cloud is unstructured, irregular, and unordered, it is often converted into volumetric shapes and multi-view images, which are then processed using volumetric CNNs and multi-view CNNs, respectively. However, many methods exist that can be applied directly on the point clouds in an end-to-end manner using a combination of symmetric functions. These symmetric functions are composed of a multilayer perceptron that is shared by all the input points and the global feature is extracted using the maxpooling function, which is also a symmetric function. PointNet++ extended the PointNet to include local dependency by applying PointNet hierarchically on local regions. Several other methods were introduced to improve the local dependency computations. PointCNN applies X-transformations on local regions before applying PointNet-like MLPs. VoxelNet processed the point clouds directly to achieve object detection by dividing the provided input into voxels, and using the points in each voxel to compute the feature vectors for the voxels; this process is applied hierarchically in stacked voxel feature encoding layers. Notably, region proposals are also used for object detection. DGCNN presented a point cloud in the form of a graph where each point is represented as a node connected by a directed graph to its neighboring points and a convolution-like operation, EdgeConv, is implemented on.
the neighboring pairs of the points to exploit the local geometry. Huang et al. proposed a multi-scale feature extraction method that embeds local features into a low-dimensional and robust subspace\(^{137}\). SEGCloud\(^{138}\) transformed the point cloud to voxels because the former has a regular structure and therefore, the CNNs can be deployed on them. The architecture combines 3D-FCN, trilinear interpolation, and CRF to label the 3D point clouds. The processing of urban-scale voxels is computationally intensive. Semantic 3D net\(^{139}\) is a large-scale benchmark of labeled TLS points that is essential in urban-scale classification and segmentation tasks. OctNet\(^{140}\) trained a network on different resolutions of voxels to address resolution and computation challenges to segment 3D-colored point clouds in the RueMonge2014 dataset\(^{141}\) of Haussmanian-style facades into the window, wall, balcony, door, roof, sky, and shop. Engelmann et al.\(^{142}\) built its framework upon PointNet\(^{132}\) by enlarging its receptive field to cater for urban-scale scenes. Landrieu et al. presented an architecture that directly addresses the challenge of semantic segmentation of urban-scale scenes by encoding contextual relationships between object parts in the 3D point cloud\(^{143}\). The network first partitions the point cloud into simple shapes called "superpoints" that are then embedded using PointNet\(^{132}\) for onward segmentation. The superpoints enable the segmentation of large-scale scenes. Xu et al. presented a supervised classification method for LiDAR point cloud semantic labeling\(^{144}\).

There are few annotated large-scale datasets because the manual point-wise labeling of points is time-consuming and demands great effort. This is the major challenge in large-scale classification and semantic segmentation of point clouds because the tasks are mostly supervised. This is an active and ongoing research field. The task of labeling and segmenting urban scenes is an active research area, especially with the advent of deep learning technologies. Its major challenges are already scaling the existing algorithms or generating novel pipelines to cater to large-scale scenes and lack of detailed annotated datasets to serve as benchmarks for classification and segmentation tasks. Currently, deep learning techniques on point cloud are becoming increasingly popular owing to an increase in the popularity of laser scanners and the fact that they require less preprocessing as compared to both multi-view and volumetric CNNs. Point-based 3D deep learning methods and other deep learning methods applicable on other unstructured data such as social networks, are becoming increasingly popular under the term 'Geometric deep learning' introduced in LeCun's work\(^{145}\).

### 3.5 Object/instance extraction

3D object detection is crucial for several real-world applications, such as robotics, autonomous driving, and augmented/virtual Reality. It locates and recognizes objects in 3D scenes by estimating oriented 3D bounding boxes and semantic labels of the objects from their point clouds.

Range scans involve the use of the spatial coordinates of the 3D point cloud, and thus, they have an advantage over camera images in locating the detected objects. Furthermore, point clouds are robust to changes in illumination. In addition, compared with image detection, object detection in point cloud naturally locates an object in 3D and provides crucial information for use in subsequent tasks, such as in navigation. However, unlike images, 3D point clouds are sparse and have inconsistent point densities owing to the non-uniform sampling in 3D space, limited sensor ranges, and presence of occlusions. Thus, detecting objects from their point clouds continues to be a huge challenge.

Existing object detection methods for point clouds are mainly divided into three categories as follows:

1. Projection-based methods, which project the point clouds into multiple perspective views, and then apply image-based object detection methods.
2. Voxelization-based methods, which rasterize the point...
clouds into a 3D voxel grid and then transform them into regular tensors. (3) Direct methods, which project the point clouds and predict the bounding boxes directly without further processing.

**Projection-based methods**

Projection-based methods project point clouds into perspective views and apply image-based techniques, which may sacrifice critical geometric details\(^\text{146}\). Alejandro et al. developed a multi-cue, multimodal, and multi-view framework for pedestrian detection with handcrafted features and a random forest classifier, which increases the accuracy by a comparatively large margin\(^\text{147}\). Li et al. presented 3D point clouds in a 2D point map and then used a fully convolutional network to simultaneously predict the confidence of objects detected and bounding boxes\(^\text{148}\). Chen et al. formulated an object detection problem as minimizing an energy function encoding object size prior, ground plane, and several depths informed features such as point cloud densities, and distance to the ground etc\(^\text{149}\). Yang et al. proposed a proposal-free, single-stage 3D object detector, called PIXOR, that estimates the oriented 3D objects from pixel-wise neural network predictions on point clouds\(^\text{150}\).

**Voxelization-based methods**

Voxelization-based methods grid irregular point clouds to 3D voxels, and then apply 3D CNN for object detection. These methods fail to leverage data sparsity and suffer from high time cost due to 3D convolution operations. Dominic et al. proposed an efficient and effective framework to apply the sliding window approach on a 3D point cloud for object detection\(^\text{151}\). They demonstrated that exhaustive window searching in 3D can efficiently exploit the sparsity problem. They proved the mathematical equivalence between sparse convolution and voting. Martin et al. detected 3D objects in point clouds using CNNs constructed from sparse convolutional layers\(^\text{152}\). Chen et al. proposed multi-view 3D networks (MV3D) by using both LiDAR point clouds and images to predict oriented 3D bounding boxes\(^\text{153}\). Li et al. proposed a 3D fully convolutional network for object detection in a point cloud\(^\text{154}\). Zhou et al. proposed a 3D detection network, called VoxelNet, by integrating feature extraction and bounding box prediction into an end-to-end deep network\(^\text{155}\). Daniel et al. presented a method for detecting small and potentially obscure obstacles in vegetated terrain\(^\text{156}\). The novelty of this method is the coupling of a volumetric occupancy map with a 3D CNN, which allows for the training of an efficient and highly accurate framework for detection tasks from raw occupancy data.

**Direct methods**

Recently, many approaches have been designed to operate on raw point clouds and predict bounding box directly without other processing. Shi et al. proposed PointRCNN for 3D object detection from a point cloud by using the bottom-up 3D proposal generation and refinement in canonical coordinates\(^\text{156}\). Charles et al. introduced VoteNet, which "votes" for object centroids directly from point clouds and aggregates votes to generate high-quality object proposals by local geometry\(^\text{157}\). Alex et al. proposed PointPillars, a method for object detection in 3D that enables end-to-end learning with only 2D convolutional layers\(^\text{158}\). PointPillars uses a novel encoder that learns features on vertical columns (pillars) of the point cloud to predict 3D-oriented boxes for objects.

In summary, with the evolution of deep learning architectures suited for point clouds, 3D object detection plays a key role in point cloud processing. However, direct detection of 3D objects in the raw point cloud is still a problematic issue and worthy of future research.

4 Typical urban modeling applications based on MLS

MLS technology has greatly facilitated urban 3D modeling of both indoor and outdoor environments.
Nowadays, more and more applications based on MLS have been proposed. In this section, we introduce four major applications based on MLS: (1) Building facet modeling, (2) high-definition (HD) map, (3) building information models, and (4) traffic visibility evaluation.

### 4.1 Building facet modeling

Recently, 3D modeling of large-scale urban buildings and reconstruction of indoor scenes has attracted increasing attention. Urban buildings are usually composed of complex primitives that may be difficult to model, as shown in Figure 8.

![Example of urban building modeling with complex structures](image)

*Figure 8  Example of urban building modeling with complex structures. Enlarged areas give more details of the model.*

The rapid developments in LiDAR technology, however, have greatly facilitated the acquisition of 3D model data for indoor and large-scale urban scenes. The captured point cloud is inherently capable of representing the physical geometry of real scenes, which facilitates modeling. However, in city scenes, there is a large number of urban objects with a great variety of shapes, and thus, it is difficult and time-consuming to carry out manual modeling of urban buildings from raw point clouds.

Automatic reconstruction of refined 3D models of large-scale urban buildings from raw point clouds is still a big challenge for researchers. The main difficulty is the data quality of raw point clouds from urban buildings. LiDAR point clouds are often contaminated by noise and outliers; they may also be influenced by point density, coverage, and occlusions.

Zhou et al. proposed a novel building segmentation and damage detection to realize automated component-level damage assessment of major building envelop elements including walls, roofs, balconies, columns, and handrails. Goebbels et al. used airborne LiDAR point clouds and true orthophotos to get better building model edges. Zhang et al. constructed a Delaunay triangulated irregular network (TIN) model and an edge length ratio based trace algorithm to refine the building’s boundary; they then used clusters from the same plane point set to determine the roof structures. Chen et al. integrated the LiDAR point cloud and large-scale vector map to model buildings. They preprocessed LiDAR point cloud and vector maps, roof analysis, and building reconstruction in three steps to get the building models. Yi et al. used the divide-and-conquer strategy to decompose the entire point cloud into a number of individual building subsets and then extracted the primitive elements through a novel algorithm called spectral residual clustering. The final 3D building model was generated by applying the union Boolean
operations over the block models.

Xiong et al. analyzed the topology graphs of building model surfaces and found the three basic primitives of roof topology graphs. Wang et al. combined the advantages of point clouds and optical images to accurately define building facade features. Zhang et al. proposed a novel framework for urban point cloud classification and reconstruction. They presented an activation function that rectified linear units’ neural networks (ReLU-NNs) to speed up convergence of the rectified linear units (ReLU). Diaz et al. detected doors and analyzed the visibility issue of indoor environments. Stambler et al. introduced room-, floor-, and building-level reasoning, and built highly accurate models by performing modeling and recognition simultaneously over the entire building.

Javanmardi et al. proposed an automatic and accurate 3D building model reconstruction technique that integrates an airborne LiDAR point cloud with a 2D boundary map. Zhang et al. proposed a deep neural network that integrates a 3D convolution, a deep Q-network, and a residual recurrent neural network to acquire semantic labels for large-scale point cloud data. They then used classification results and an edge-aware resampling algorithm to generate urban building models. López et al. utilized historical and bibliographical data to obtain graphic and semantic information of the point cloud, and used BIM software to create a library of parametric elements. Ochmann et al. developed a parametric building model that incorporates contextual information such as global wall connectivity. Xiong et al. proposed a parameter-free algorithm to robustly and precisely construct roof structures and building models.

Hojebrí et al. proposed a method based on the fusion of a structure’s point cloud and image to obtain accurate modeling results. Hron et al. reviewed auto-generation of 3D building models from the point cloud. Based on the concept of data reuse, Chen et al. proposed a building modeling method that has physical geometric shapes similar to a user-specified point cloud query; the shapes can be retrieved and reused for data extraction and modeling of buildings. Zhang et al. used the Canny and Hough transform operator to extract the edges of the building, and the E3De software to obtain the 3D building model. Chen et al. introduced a novel encoding scheme based on low-frequency spherical harmonic basis functions for 3D building model retrieval.

In contrast to previous studies, Demir et al. proposed an approach that can operate directly on the raw point cloud. Their approach consists of semi-automatic segmentation, a consensus-based voting schema, a pattern extraction algorithm, and an interactive editing tool. Wu et al. proposed a fast and easy algorithm of plane segmentation based on cross-line element growth (CLEG) for 3D building modeling. Chen et al. integrated point cloud and large-scale vector maps to get the 3D building model. Wang et al. proposed a novel semantic line framework–based modeling building method based on the backpacked point cloud. The proposed method can effectively perform line framework extraction and output results for building modeling.

Table 2 presents a quantitative evaluation of various building models.

### 4.2 High-definition (HD) map

HD map is a crucial technology for autonomous driving, especially for vehicle localization and motion planning for cars.

There are plenty of related works for constructing HD maps. Zhang et al. built an HD map system and described the components of an HD map according to embodiments. Siam et al. proposed a semantic segmentation method to construct HD maps from images. Barsi et al. created HD maps using the TLS system.
HD maps can be classified into two types: Dense semantic point cloud maps and maps based on the landmark. Dense semantic point cloud maps are constructed by using a laser scanning point cloud with semantics. Such HD maps include the road surface, road markings, road boundaries, traffic signs, etc. Hence, many technology-oriented companies use this type of map due to its highly accurate and integrated road information.

However, it is difficult to build HD maps directly from the collected LiDAR point cloud. Generally, the collected point cloud consists of buildings, roads, parking lots, vegetation, and other points that are not relevant for generating the map. Therefore, there are many studies focused on how to extract each component of HD maps separately.

### 4.2.1 Road surface

The road surface is one of the primary components of the HD map, and also one of the essential parts of a road structure. In general, the raw data collected from the laser scanning system contain many irrelevant points and noise; separating the on-road points and off-road points from raw data is a key step for HD map component extraction. Many methods have been proposed for road surface detection and extraction from the point cloud and these are mainly categorized into two basic methods: (1) 3D-based methods, and (2) georeferenced feature (GRF)-based methods\(^{189}\).

To decrease computation complexity, trajectory information was used in road surface extraction. Wu et al. vertically partitioned raw point clouds using trajectory information; they then used the random sample consensus (RANSAC) method to extract ground points by calculating the average height of ground points\(^{190}\). Based on point cloud features, Hata et al. extracted ground surfaces by using different filters, including differential filters and regression filters, on the point cloud\(^{191}\). There are some curb-based road surface extraction methods, for example, Guan et al. assumed that road curbstones can represent the boundaries of the pavement and extracted the road surface by separating pavement surfaces from roadsides\(^{192}\).

Many methods convert the point cloud into a 2D GRF map; road surfaces are then efficiently detected and extracted based on existing computer vision technologies. To minimize computation complexity,
Riveiro et al. projected the point cloud onto a 2D space and then detected the road by using principal component analysis (PCA)\(^{193}\). Yang et al. extracted road surfaces by generating GRF images to filter out off-ground objects\(^{194}\).

### 4.2.2 Road boundary

Road boundary, also called road edge, road curb, or curbstone, is an essential part of HD maps. It is mostly extracted by determining the height variance between sidewalks and driveways. To reduce computational complexity, some trajectory-based methods have been proposed. Wang et al. first divided the point cloud into several parts along the trajectory\(^{195}\); the road boundary was then extracted and refined from each part. Wang et al. extracted the road boundary from the point cloud by assuming that the altitude of the road boundary points varies considerably from the altitude of road surface points\(^{196}\). Zai et al. detected the rough road boundary via super voxels and the alpha-shape algorithm, and then extracted the curb by applying graph cuts on the trajectory and rough boundary\(^{197}\). Since road edge extraction can be regarded as a classification problem, Rachmadi et al. detected the road edge from the 3D point cloud using an encoder-decoder convolutional network\(^{198}\). Based on the 3D local feature, Yang et al. proposed a new binary kernel descriptor (BKD) to detect road curbs and markings\(^{199}\).

### 4.2.3 Road markings

Road markings are an important part of the HD map for self-driving; they form a key component to achieve accurate navigation. Many related studies extract road markings from the laser scanning point cloud. Road markings consist of different marking types, such as lane lines, zebra crossings, arrows, and texts. Therefore, much research also studied the classification of road markings. The related studies can be mainly divided into two categories: (1) methods based on the 3D method, and (2) methods based on GRF projection.

3D-based methods extract road markings directly from the road surface based on the distinct intensity difference between markings and other points. The trajectory of a vehicle can be used to locate the positions of road markings. Hence, Chen et al. proposed a profile-based intensity analysis by partitioning the point cloud into slices along the trajectory of the vehicle and then extracted the road markings via analyzing the peak value of intensity in each scan line\(^{198}\). Yu et al. extracted road markings by using a multi-segment thresholding strategy and spatial density filtering from the point cloud; they then extracted and classified small-sized road markings via deep Boltzmann machine (DBM)-based neural networks\(^{199}\).

Jung et al. rasterized the point cloud into the x-y plane; the lane markings were then extracted by intensity contrast\(^{200}\). As road marking extraction can be treated as semantic segmentation, neural networks can be applied to the extraction problem. With the emergence of many image classification networks, road markings can be classified efficiently by such networks. Wen et al. proposed a deep learning framework to extract, classify, and complete road markings. A modified U-net was first used to extract road markings from the projected intensity image, and then a multi-scale clustering algorithm and a CNN classifier was applied to classify them\(^{201}\). Finally, they completed the classified markings using a conditional generative adversarial network and a context-based method.

### 4.2.4 Traffic signs

Traffic signs are also an important part of HD maps; they provide critical information about roads for traffic safety in navigation during autonomous driving. Generally, traffic signs are part of pole-like objects in a raw point cloud. Therefore, in most studies, pole-like object extraction was performed first. Then, the pole-like objects were classified into different categories that contain different types of traffic signs. Most
researches were performed by analyzing the position, continuity, verticality, shape, size, and intensity of the pole-like objects\cite{193}.

Based on the size and intensity difference, Wen et al. set a minimum threshold in clusters to remove small objects and filtered out non-sign objects\cite{201}. By using the traffic sign attributes, Arcos-García et al. developed height and planar filters to eliminate small parts and non-planar parts\cite{202}. Huang et al. first detected traffic signs from the point cloud based on high intensity and position; then performed occlusion detection to analyze traffic sign occlusion by observing the relationship between viewpoint and traffic signs\cite{203}.

The detected traffic signs are usually classified into different types by analyzing the features of the point clouds and images. Wen et al. extracted integral features consisting of a histogram of gradients (HOG) and color descriptors; they used the support vector machine (SVM) to train the classification model\cite{201}. Some deep learning-based methods have been also been proposed for traffic sign recognition\cite{204}. Yu et al. projected the point cloud into a 2D image and applied the Gaussian-Bernoulli deep Boltzmann machine model for traffic sign recognition (TSR)\cite{205}.

The main issue in building an HD map using the LiDAR point cloud is how to accurately extract each component of the HD map from raw data. However, as of now, there is no method to extract all the HD components simultaneously. With the development of point cloud semantic segmentation, this goal may be achieved in the future.

### 4.3 Building information models (BIM)

The indoor building model is the data source of BIM, which plays a vital role in building maintenance, disaster rescue, and building renewal planning. However, it is a time-consuming and labor-consuming process to generate indoor three-dimensional models artificially. To generate three-dimensional models more efficiently, many studies develop indoor models from the original point clouds automatically. For example, Previtali et al. proposed a method based on optimization to detect the indoor characteristics of buildings\cite{206}. Wang et al. proposed a new method of realizing line frame-based semantic indoor modeling. Tran et al. proposed a novel shape grammar method which can effectively generate three-dimensional models\cite{207}. Shi et al. presented a method capable of automatically reconstructing 3D building models with semantic information from the unstructured 3D point cloud of indoor scenes\cite{208}. Xiao et al. presented a framework that recovers missing points and estimates connectivity relations between planar and non-planar surfaces to obtain complete and high-quality 3D models\cite{209}.

Existing approaches to realize indoor modeling can be classified as linear-primitive, planar-primitive, and volumetric-primitive types\cite{210}.

#### 4.3.1 Linear-primitive

The line-primitive indoor modeling method assumes that the wall is plane and vertical to the ground, and the indoor model is built based on the plane map. Oesau et al. presented a graph-cut-based indoor-reconstruction method to solve an inside/outside labeling of a space partitioning based on the raw point cloud\cite{211}. Ochmann et al. proposed a parametric modeling method for reconstructing parametric three-dimensional building models from indoor point clouds and automatically reconstructing structural models containing multiple indoor scenes\cite{212}. Ochmann et al. also presented a novel method of tackling the indoor building reconstruction problem from point clouds using integer linear programming\cite{213}. Li et al. presented a segmentation method for the reconstruction of 3D indoor interiors\cite{214}. This method overcomes the over-segmentation of graph-cut operations for long corridors and removes shared surfaces to reconstruct
connected areas across multiple floors. The line-primitive indoor modeling method deals with indoor point clouds from a two-dimensional perspective, which is usually only applicable under ground independence and no clutter conditions.

### 4.3.2 Planar-primitive

The planar-primitive methods mainly involve two steps. First, the plane is extracted by classification, and then, the plane model is built based on the classification result. Sanchez et al. used random sample consensus (RANSAC) for plane fitting and the alpha shape for calculating their ranges and extracting large-scale plane structures such as the ground, ceiling, and wall from indoor point cloud data\(^{(213)}\). Similarly, Budroni et al. used plane scanning to extract ceilings, floors, and walls\(^{(214)}\). These methods can extract the plane very well, but they do not consider occlusion. Moreover, these methods typically use "context-based" reasoning to distinguish building elements before plane fitting and intersection. These methods are not suitable for complex indoor scenes or serious data-missing situations. Wang et al. proposed a method of first semantically classifying the 3D point clouds into different categories and then extracting the line structures from the labeled points separately\(^{(181)}\).

### 4.3.3 Volumetric-primitive

The volumetric-primitive methods have strong regularity. These methods generally satisfy the Manhattan world hypothesis that only vertical and horizontal environments can be included. Furukawa et al. proposed an inverse solid geometry algorithm that detects walls in 2D and then combines them into cubes\(^{(215)}\). Khoshelham et al. proposed a grammar-based approach to reconstruct the indoor space that satisfies the Manhattan world hypothesis by iteratively placing, connecting, and merging the cubes\(^{(216)}\). Previtali et al. transformed the indoor reconstruction problem into a labeling problem of result units in a two-dimensional plane under the condition that the Manhattan world hypothesis is satisfied\(^{(217)}\). Kim et al. presented a geometry and camera pose reconstruction algorithm from image sequences for indoor Manhattan scenes\(^{(218)}\).

### 4.3.4 Door and window detection

Indoor building models generally include the main structures of the buildings, such as ceilings, floors, walls, doors, windows, and other immovable objects, excluding the furniture and other movable objects. The detection of doors and windows is also a necessary part of indoor building models. Michailidis et al. focused on the wall and the extraction of the structures of doors and windows by detecting the holes in the wall\(^{(219)}\). However, this method can only operate on a single wall, and cannot be directly implemented on all indoor point cloud data. Wang et al. determined the most peripheral boundary line of the wall through point clouds on the ground and ceiling, and then, only the internal line structure was retained when extracting the wall line structure that is used to detect the locations of doors and windows\(^{(190)}\). Jung et al. first divided the point cloud into several separate rooms and then modeled the wall; finally, they projected the point on the wall onto a reverse binary and then detected the doors and windows\(^{(209)}\). Quintana et al. proposed a method of detecting doors and windows in three-dimensional color point clouds\(^{(220)}\). The method detects open doors based on rectangular data on the wall, and detects closed doors by identifying the rectangular areas that do not correspond to the actual wall area in subsequent processing. Previtali et al. proposed a voxel-based marking method based on visibility analysis\(^{(222)}\). Diaz-Vilariño et al. applied the generalized Hough transform to wall orthophoto images generated by color point clouds to detect closed doors\(^{(223)}\). Previtali et al. detected occluded doors and windows by implementing a ray-tracing algorithm after extracting the wall\(^{(224)}\). Doors and windows are modeled by obtaining parametrized rectangular shapes in images using the generalized Hough transform (Diaz-Vilariño et al.\(^{(224)}\). Nikoohemat et al. presented several algorithms for
the interpretation of the interior space using MLS point clouds, in combination with the trajectory of the acquisition system\cite{225}.

### 4.4 Traffic visibility evaluation

Maintaining high visibility of traffic signs is crucial for traffic safety. Research on the visibility of traffic signs has provided the following categories of methods: simulation-based, image-based, naturalistic driving experimentation-based, and point clouds-based methods.

Simulation-based methods gather statistics based on visual or cognitive information collected from volunteers, and output evaluation results by performing simulation. Some researchers use the simulation platform to investigate the cognition time\cite{226}, driver behavior associated with visual distractions\cite{227}, and cognitive workload\cite{228}. Motamedi et al. analyzed traffic-sign visibility in BIM-enabled virtual reality (VR) environments\cite{229}. Eye-tracker equipment\cite{230,231} has also been used to determine the visual cognition of traffic signs under simulated driving conditions. Simulation-based methods cannot provide a quantitative evaluation of visibility and visual or cognitive information for real roads.

Image-based methods compute the visibility of a traffic sign based on different contrast ratios and numbers of pixels in the occluded area of an image\cite{232-235}. These methods cannot continuously evaluate visibility over an entire road surface because of viewpoint-position limitations. Meanwhile, image-based methods are not robust to lighting conditions and do not consider the current geometric properties of the road and traffic signs.

Naturalistic driving experimentation-based methods recognize driving modes by observing a driver's behavior over a prolonged period under natural conditions\cite{236-238}. Because humans require time for cognition, a driver has to stop to obtain visibility from a given viewpoint when driving in natural conditions. This drawback renders obtaining the visibility distribution of traffic signs difficult.

Point cloud-based methods study the visibility of traffic signs using point clouds\cite{203,239,240}. Mobile laser scanning (MLS) systems provide an efficient 3D measurement over large-scale traffic environments. Zhang et al. proposed the concept of the visibility field of a traffic sign and considered the geometric, occlusion, and sightline-deviation factors to build a model for evaluating the visibility distribution of traffic signs\cite{241}. Their algorithm is, to date, the only automated algorithm that can test the visibility field on real roads on a large scale. The experimental results are presented in Figure 9.

![Figure 9](image)

**Figure 9** Large-scale applications of visibility-field calculation on a real road\cite{240}. In this Figure, the detected traffic signs are presented in yellow color, occluding point clouds are in red color, and visibility-field results are presented as mesh planes. The box marked with a cross (“X”) represents the occluded traffic sign that is observed in the area. The color change of the mesh planes from green to red indicates that visibility values change from large to small.
5 Future work

Despite it achieving success, 3D modeling based on MLS scanning encounters many challenges. First, the next-generation MLS must have a new FOG-based IMU and a multi-GNSS constellation receiver to promote more reliable positioning. It will also integrate a smaller laser scan head to achieve a higher scanning frequency and easier operation. At the same time, owing to the rapid development of the present hardware technology, the expected cost of the MLS system will continue to fall, which will result in its more widespread use. More research and applications are required to explore the full potential of MLS in the future, combining LiDAR data and UAV images. According to the collected data, automatic algorithms, such as terrain extraction, urban 3D modeling, and vegetation analysis, require further development, and semi-automatic change-detection mapping also requires further development.

At present, deep learning on point cloud is in its early stages. Current work should not only focus on improving the accuracy and performance of the dataset but also ensure robustness and portability of the methods. More sophisticated deep-learning architectures need to be developed to handle the challenge of uneven distribution and possible insufficient sampling in the point cloud from the real world. Few datasets capture the complexity of real-world urban scenes, and comprehensive semantic understanding of the complex urban street is a challenge encountered by artificial intelligence (AI). The rapidly growing urban MLS point cloud data will give rise to a new category of geo-big data, and it provides more opportunities to develop better AI on point clouds. Finally, changes in downtown buildings, roads, and vegetation never stop, together with the dynamic scenery of traffic and pedestrians. Most methods only focus on how an accurate 3D city model can be developed by scanning the city only once; there is a lack of abundant dynamic information of the real world. Dynamic 3D modeling, combing the point cloud with other sensors such as cameras, is more challenging and worth studying.

Future applications of the MLS system will play an important role in various detection and modeling tasks in various civil fields such as transportation, civil engineering, forestry and agriculture, and in-process monitoring, and in understanding some natural sciences such as archaeology and geosciences.

6 Conclusion

Large-area urban 3D modeling has evolved rapidly in the past few years. The current development of MLS-based urban 3D modeling includes two parts: development of the hardware MLS system and the processing of point clouds, including LiDAR SLAM, point-cloud registration, feature extraction, object extraction, semantic segmentation, and deep point cloud processing. The current development of MLS brings together various levels of innovation from deep point cloud processing to high-level applications such as BIM, HD map, and traffic monitoring. In this paper, we reviewed the research that has been conducted on mobile mapping and urban 3D modeling using laser scanning.

References

1 El-Sheimy N. The development of VISAT: a mobile survey system for GIS applications. 1996
   DOI:10.3390/s120911712
5 Olsen M J. Guidelines for the use of mobile LIDAR in transportation applications. Transportation Research Board, 2013


24 Qin T, Cao S Z, Pan J, Shen S J. A general optimization-based framework for global pose estimation with multiple sensors. 2019


DOI:10.1007/s00371-007-0167-y

DOI:10.1109/tdpt.2002.1024098

DOI:10.1109/humanoids.2012.6651593

DOI:10.5220/0006170005650570

DOI:10.1111/cgf.12481

DOI:10.1109/iros.2005.1545152

DOI:10.1109/iccv.2005.221

DOI:10.1145/1141911.1141924

DOI:10.1145/1618452.1618484

DOI:10.1145/1778765.1778831

DOI:10.1109/cvprw.2012.6238908

DOI:10.1145/1399504.1360642

DOI:10.1111/cgf.12573


53 Nan L L, Xie K, Sharf A. A search-classify approach for cluttered indoor scene understanding. ACM Transactions on
DOI:10.1145/2185520.2185551

DOI:10.1007/978-3-319-46466-4_29

DOI:10.1109/tgrs.2014.2344714

DOI:10.1016/j.isprsjr.2014.12.027

DOI:10.1109/tgrs.2012.2205003

DOI:10.1109/tgrs.2014.2359951

DOI:10.1109/tro.2010.2042989

DOI:10.1109/tpami.2008.300

DOI:10.1016/j.patrec.2011.06.001

DOI:10.1109/cvpr.2010.5539781

DOI:10.1109/smai.2007.32

DOI:10.1007/s11042-012-0999-y

DOI:10.1109/tgrs.2016.2639025

DOI:10.1109/34.3881

68 Pu S, Vosselman G. Automatic extraction of building features from terrestrial laser scanning. International Archives of
DOI:10.1016/j.isprsjprs.2007.05.012

DOI:10.1016/j.isprsjprs.2013.02.019

Stechschulte J, Heckman C. Hidden Markov random field iterative closest point. 2017

DOI:10.1109/tpami.2015.2513405

DOI:10.1109/cvpr.2016.613

DOI:10.1109/cvpr.2017.258

DOI:10.1007/978-3-642-15558-1_26

DOI:10.1007/s11263-013-0627-y

DOI:10.1016/j.patcog.2016.11.019

DOI:10.1109/iros.2008.4650967

DOI:10.1016/j.isprsjprs.2017.10.001

DOI:10.1109/cvpr.2017.29

DOI:10.1145/3137609

DOI:10.1109/cvpr.2017.265


110 Sarode V, Li X Q, Goforth H, Aoki Y, Choset H. PCRNet: point cloud registration network using PointNet encoding. 2019


DOI:10.1109/icra.2013.6630888

DOI:10.1109/robot.2010.5509169

DOI:10.1109/iros.2012.6386039

DOI:10.1109/cvpr.2009.5206590

DOI:10.1109/cvpr.2015.7298801

DOI:10.1109/cvpr.2016.609

DOI:10.1109/iros.2015.7353481

DOI:10.1109/iccv.2015.114

DOI:10.1016/j.sigpro.2014.09.005

125 Tosbeberg P. Semantic segmentation of point clouds using deep learning. 2017

DOI:10.1109/icra.2018.8462926

DOI:10.1007/978-3-030-11024-6_39

DOI:10.1109/ivs.2017.7995848

DOI:10.1007/978-3-319-64689-3_8


204


Learning where to classify in multi-view semantic classification and segmentation. 


DOI:10.1109/ivs.2015.7225711

DOI:10.15607/rss.2016.xii.035


DOI:10.1109/cvpr.2018.00798

DOI:10.15607/rss.2015.xii.035

DOI:10.1109/icra.2017.7989161

DOI:10.1109/cvpr.2017.691

DOI:10.1109/ijrobonics.2017.8205955

DOI:10.1109/icra.2015.7139679

DOI:10.1109/cvpr.2019.00086

DOI:10.1109/iccv.2019.00937

DOI:10.1109/cvpr.2019.01298

DOI:10.1016/j.cad.2017.07.005

DOI:10.1111/mice.12345

162 Zhang D, Du P. 3D building reconstruction from lidar data based on Delaunay TIN approach. SPIE, 2011


DOI:10.14358/pers.73.9.1147


DOI:10.1016/j.isprsjprs.2015.01.002


DOI:10.1016/j.isprsjprs.2017.11.015


DOI:10.1016/j.isprsjprs.2018.02.008


DOI:10.3390/s150203491


DOI:10.1109/3dv.2014.65


DOI:10.1109/icves.2015.7396906


DOI:10.1109/tgrs.2017.2769120


DOI:10.1080/15583058.2017.1325541


DOI:10.1016/j.cag.2015.07.008


174 Hojebri B, Samadzadegan F, Arefi H. Building reconstruction based on the data fusion of lidar point cloud and aerial imagery. 2014, 103–121


DOI:10.1007/978-3-319-11463-7_8


DOI:10.3390/ijgi6090269


DOI:10.1088/1757-899x/301/1/012037

178 Chen J Y, Lin C H, Hsu P C, Chen C H. Point cloud encoding for 3D building model retrieval. IEEE Transactions on
DOI:10.1109/iccv.2015.248

DOI:10.3390/rs8050383

DOI:10.1016/j.isprsjprs.2018.03.025

DOI:10.1109/jstares.2014.2349003

DOI:10.1016/j.eng.2016.02.010

DOI:10.1109/itsc.2016.7795600

DOI:10.1109/cvpr.2019.00886

Zhang R, Chen C, Di Z, Wheeler M D. Visual odometry and pairwise alignment for high definition map creation. 2019

DOI:10.1109/itsc.2017.8317714

DOI:10.30657/pea.2017.16.09

DOI:10.3390/rs10101531

DOI:10.1109/tits.2016.2565698

DOI:10.1109/ivs.2014.6856405

DOI:10.1016/j.isprsjprs.2013.11.005
DOI:10.1016/j.optlastec.2015.01.011

DOI:10.1016/j.isprsjprs.2017.02.014

DOI:10.1109/lgrs.2015.2449074

DOI:10.1109/cvrs.2012.6421248

DOI:10.1109/kcic.2017.8228570

DOI:10.1145/1653771.1653851

DOI:10.1109/jstars.2014.2347276

DOI:10.1016/j.isprsjprs.2018.11.012

DOI:10.1109/tits.2015.2418214

DOI:10.1016/j.eswa.2017.07.042

DOI:10.1109/tits.2016.2639582

DOI:10.1016/j.isprsjprs.2016.01.005

205 Previtali M, Díaz-Vilariño L, Scaioni M. Towards automatic reconstruction of indoor scenes from incomplete point clouds: door and window detection and regularization. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2018, XLII-4: 507–514
DOI:10.5194/isprs-archives-xlili-4-507-2018

206 Tran H, Khoshelham K, Kealy A, Díaz-Vilariño L. Shape grammar approach to 3D modeling of indoor environments
DOI:10.1061/(asce)cp.1943-5487.0000800

DOI:10.3390/ijgi8010009

DOI:10.1061/(asce)cp.1943-5487.0000776

DOI:10.5194/isprs-archives-xlii-2-w7-345-2017

DOI:10.1016/j.isprsjprs.2014.02.004

DOI:10.1016/j.isprsjprs.2019.03.017

DOI:10.3390/rs10081281

DOI:10.1109/icip.2012.6467225

DOI:10.1260/1478-0771.8.1.55

DOI:10.1109/iccv.2009.5459145

DOI:10.5194/isprsarchives-xl-5-321-2014

217 Previtali M, Díaz-Vilaríño L, Scaioni M. Indoor building reconstruction from occluded point clouds using graph-cut and ray-tracing. Applied Sciences, 2018, 8(9): 1529
DOI:10.3390/app8091529

DOI:10.1109/3dv.2018.00076

DOI:10.1007/s00371-016-1230-3

DOI:10.1016/j.aei.2018.10.007

DOI:10.1016/j.autcon.2017.10.016


227 Li N X, Busso C. Predicting perceived visual and cognitive distractions of drivers with multimodal features. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(1): 51–65. DOI: 10.1109/tits.2014.2324414


236 Balsa-Barreiro J, Valero-Mora P M, Berné-Valero J L, Varela-García F A. GIS mapping of driving behavior based on
naturalistic driving data. ISPRS International Journal of Geo-Information, 2019, 8(5): 226
DOI:10.3390/ijgi8050226

DOI:10.1049/iet-its.2012.0152

DOI:10.1007/s12239-020-0070-3

DOI:10.1145/1275808.1276407

DOI:10.1109/cvpr.2013.23

DOI:10.3390/rs11121453