

## Hierarchical extraction of urban objects from mobile laser scanning data

Bisheng Yang\*, Zhen Dong\*, Gang Zhao, Wenxia Dai

*State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China*



### ARTICLE INFO

#### Article history:

Received 16 August 2014

Received in revised form 24 October 2014

Accepted 27 October 2014

Available online 20 November 2014

#### Keywords:

Mobile laser scanning

Multi-scale supervoxels

Segmentation

Object extraction

Classification

Filtering

### ABSTRACT

Point clouds collected in urban scenes contain a huge number of points (e.g., billions), numerous objects with significant size variability, complex and incomplete structures, and variable point densities, raising great challenges for the automated extraction of urban objects in the field of photogrammetry, computer vision, and robotics. This paper addresses these challenges by proposing an automated method to extract urban objects robustly and efficiently. The proposed method generates multi-scale supervoxels from 3D point clouds using the point attributes (e.g., colors, intensities) and spatial distances between points, and then segments the supervoxels rather than individual points by combining graph based segmentation with multiple cues (e.g., principal direction, colors) of the supervoxels. The proposed method defines a set of rules for merging segments into meaningful units according to types of urban objects and forms the semantic knowledge of urban objects for the classification of objects. Finally, the proposed method extracts and classifies urban objects in a hierarchical order ranked by the saliency of the segments. Experiments show that the proposed method is efficient and robust for extracting buildings, streetlamps, trees, telegraph poles, traffic signs, cars, and enclosures from mobile laser scanning (MLS) point clouds, with an overall accuracy of 92.3%.

© 2014 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

### 1. Introduction

Mobile laser scanning (MLS) systems capture three-dimensional (3D) point clouds with high flexibility and precision, thus are widely used for various applications (e.g., transportation, forestry). However, MLS systems, unlike traditional surveying equipment, collect huge data volumes, resulting in urgent demands for efficient and effective processing of the point clouds, or the productivity gained in the data collection phase may be lost during processing. Points collected by MLS systems from urban scenes contain numerous objects with significant disparities in size, complicated and incomplete structures, holes, varied point densities, and huge data volumes, raising great challenges for automated point segmentation and object extraction. In recent years, there have been many scientific contributions aiming to process mobile laser scanning point clouds from urban scenes, focusing on segmentation (Aijazi et al., 2013; Barnea and Filin, 2013; Lari and Habib, 2013; Serna and Marcotegui, 2014; Yang and Dong, 2013; Yao et al., 2009), roads extraction and modelling (Boyko and Funkhouser, 2011; Guan et al., 2014; Hernández and Matcotegui,

2009a,b; Yang et al., 2012, 2013a; Zhu and Hyppa, 2014), pole-like objects extraction (Cabo et al., 2014; Lehtomäki et al., 2010; Li and Elberink, 2013; Monnier et al., 2012; Pu et al., 2011; Yang and Dong, 2013), and building extraction and reconstruction (Jochem et al., 2011; Pu and Vosselman, 2009; Yang et al., 2013b).

Segmentation is the fundamental step for extracting objects from MLS point clouds. Barnea and Filin (2013) proposed a segmentation method for terrestrial laser scanning data by integrating ranges, normals, and colors in a panoramic representation. The proposed segmentation method yielded more physically meaningful segments. Objects were extracted by forming the segments as meaningful units according to predefined rules (Pu et al., 2011; Yang and Dong, 2013). Pu et al. (2011) presented a framework for structured recognition from MLS point clouds and classified the objects as traffic signs, trees, building walls, and barriers based on the characteristics of points segments, such as size, shape, orientation, and topological relationship. Yang and Dong (2013) proposed a shape-based segmentation method that classified points according to the geometric features derived from support vector machines (SVMs) for objects extraction. All of the above methods have difficulties extracting individual objects from areas of dense mixed objects, and have heavy computing costs because the local geometric features of each point must be calculated.

\* Corresponding authors.

E-mail addresses: [bshyang@whu.edu.cn](mailto:bshyang@whu.edu.cn) (B. Yang), [\(Z. Dong\).](mailto:dongzhenwhu@whu.edu.cn)

[Lehtomäki et al. \(2010\)](#) developed an automated method for the detection of vertical pole-like structures (e.g., traffic signs, lamp posts) in road environments with the support of profile information from the scanner. [Yang et al. \(2013a\)](#) presented a method for extracting and delineating roads from large-scale MLS point clouds by detecting curb points from a set of consecutive scanning lines. These methods are suitable for the rapid classification of MLS data and the extraction of specific kinds of objects, but they are difficult to deal with unordered laser scanning data (particularly when point clouds from more than two laser scanners are mixed).

[Hernández and Matcotegui \(2009a\)](#) presented a method to generate range images from point clouds for extracting features using morphological operators. [Serna and Marcotegui \(2014\)](#) proposed an automatic and robust approach to classify objects from the points of urban scenes using elevation images. The result is re-projected onto the 3D point cloud, to detect, segment and classify urban objects. These methods accelerate the speed of operation and improve the efficiency, but at the cost of loss of accuracy in the process of generating images. In addition, pixel size requires careful tuning to extract different kinds of objects. The quality of objects extraction depends on the generation of images from the points.

To reduce the computing cost of point clouds, the voxels based segmentation and classification method was developed ([Aijazi et al., 2013; Lim and Suter, 2008, 2009](#)). [Lim and Suter \(2008, 2009\)](#) first over-segmented the points into 3D voxels, then calculated the local and regional features of voxels, and finally used multi-scale Conditional Random Fields to classify 3D outdoor terrestrial laser scanning data. [Aijazi et al. \(2013\)](#) presented a method to classify urban scenes based on voxels segmentation of sparse 3D data obtained from LiDAR sensors. The method first partitioned 3D point cloud into voxels, then joined voxels by a link-chain method to create objects, and finally classified these objects using geometrical models and local descriptors. These voxels based methods accelerate the computing speed. Nevertheless, the segmentation quality is subject to the size of the voxels. Voxels of a fixed size lead incorrect results, particularly in the areas of dense mixed objects.

Although the reported methods are generally able to extract specific kinds of objects based on segmentation or with the support of scanning lines, they suffer from the quality of segmentation and there remains much room for improvement. On the one hand, the semantic knowledge of urban objects should be formed as rules for extracting and classifying urban objects. On the other hand, the time efficiency of extracting objects should be improved. This paper proposes an automated method to extract and classify urban objects following the pipeline of segmentation.

The contributions of the proposed method have three aspects:

- Generate multi-scale supervoxels from scattered MLS points to improve the estimation of local geometric structures of neighboring points and the time efficiencies of segmentation.
- Form semantic knowledge of urban objects into rules for merging adjacent segments into meaningful units, resulting in the meaningful units having better consistency with physical objects.
- Define a hierarchical strategy to extract and classify objects from urban scenes in the order of the saliency of the segments, resulting in the improved accuracy of object extraction, especially in the cluttered situation of occlusion and overlapping between closely neighboring objects.

Following the introduction to the subject, we elaborate the proposed method and test and validate the method with two MLS datasets before drawing our conclusions.

## 2. Methodology

The point clouds of urban scenes usually contain a huge number of points with varied point densities and occlusions, including ground points and non-ground points. The proposed method aims to extract urban objects from non-ground points of urban scenes robustly and efficiently. It removes ground points using the approach of [Hernández and Matcotegui \(2009b\)](#) before extraction of urban objects. The proposed method firstly generates the supervoxels of two different sizes according to the attributes (e.g., colors, intensities) of non-ground points and spatial relations between the points, resulting in multi-scale supervoxels by integrating the generated supervoxels. Each supervoxel contains the points of a certain kind of objects and has unique geometric property, and then the proposed method segments the supervoxels according to the principles of graph-based segmentation. Secondly, the proposed method defines the formula of the saliency of the segments using several factors and models the semantic knowledge of urban objects as formal representation. Finally, the proposed method forms the segments as physical urban objects according to a set of predefined rules in a hierarchical order ranked by the saliency of the segments, and classifies the objects according to the formal representation of urban objects, resulting in robust and efficient extraction and classification of urban objects.

### 2.1. Generating multi-scale supervoxels of non-ground points

Point-wise segmentation processing has a heavy computing cost and may lead to over or under segmentation. To reduce the computing costs of the point clouds of large scale urban scenes, the scene space is partitioned into 3D voxels and the points are allocated into the corresponding 3D voxels according to their coordinates. Then, we utilize a weighted distance measurement to reallocate the voxels of each point by its color or intensity and the spatial distance between itself and other close points. Inspired by the work of [Achanta et al. \(2012\)](#) for 2D image segmentation, we construct supervoxels of the point clouds by the weighted distance. The key steps of generating 3D supervoxels are:

*Step 1:* Initialize the size,  $S$ , of the 3D voxels, and partition the scene of point clouds into 3D voxels according to their associated coordinates.

*Step 2:* Set the centroid of each 3D voxel  $k$ , with the attribute  $C_k = (x_k, y_k, z_k, L_k, a_k, b_k)$ , where  $x_k, y_k, z_k, L_k, a_k, b_k$  are the coordinates and the color values in CIE Lab, respectively, of the closest point to the centroid of the 3D voxel. The mathematical formulations of the descriptive about transforming RGB values to CIE Lab colors are given in [Appendix A](#).

*Step 3:* Initialize the distances between each point  $p_i(x_i, y_i, z_i, L_i, a_i, b_i)$  in each 3D voxel and its associated centroid as infinitely large  $d_i = \infty$ .

*Step 4:* Traverse the 3D voxels one by one, search the neighboring points of the centroid  $C_k$  of each voxel within a sphere radius  $S$ , the voxel size, and calculate the associated distances between each neighboring point and the centroid by:

$$d_{ik} = \sqrt{\left(\frac{d_{ik}^c}{N_c}\right)^2 + \left(\frac{d_{ik}^s}{N_s}\right)^2} \quad (1)$$

$$d_{ik}^s = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2}$$

$$d_{ik}^c = \sqrt{(L_i - L_k)^2 + (a_i - a_k)^2 + (b_i - b_k)^2}$$

where  $N_c$  and  $N_s$  are the weights associated with color proximity  $d_{ik}^c$  and spatial distance proximity  $d_{ik}^s$ , respectively, controlling the trade-off between compactness and boundary adherence of the 3D voxel. See Achanta et al. (2012) for a detailed explanation of  $N_s$  and  $N_c$ .

*Step 5:* Suppose that  $d_{ik} < d_i$ , we replace  $d_i$  as  $d_{ik}$  and allocate  $p_i$  into the associated 3D voxel  $k$ . Repeat steps 4 and 5 until all the 3D voxels are traversed.

*Step 6:* Update the centroid of each 3D voxel by averaging the (modified) associated points. The 3-D voxel is considered completed when either: the distance between the calculated centroid and its current centroid is less than a specified threshold ( $S/5$ ); or the number of iterations is more than a specified threshold (we have found that 10 iterations suffices for most cases). Otherwise, iterate from step 4.

Once all 3-D voxels are completed, all the points are allocated into their corresponding 3D voxels, called supervoxels. Each supervoxel has an irregular shape rather than a fixed size. As the similarity between the points in each supervoxel is measured by considering the attributes of points and spatial distances in a local area, most points in each supervoxel belong to an identical object or part-object, laying a good foundation for segmentation.

The geometric structures of each supervoxel are calculated as:

Let  $p_i (i = 1, 2, \dots, k)$  be the points in one supervoxel. The covariance matrix,  $M$ , of the points may be written as

$$M_{3 \times 3} = \frac{1}{k} \sum_{i=1}^k (\vec{P}_i - \vec{P}_c)(\vec{P}_i - \vec{P}_c)^T, \quad (2)$$

$$\text{where } \vec{P}_c = \frac{1}{k} \sum_{i=1}^k p_i.$$

The eigenvalues  $\lambda_1, \lambda_2, \lambda_3$ ; ( $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$ ) of  $M$  can be determined, and the geometric structures of the supervoxel are

$$a_{1D} = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}}, \quad a_{2D} = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}}, \quad a_{3D} = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}}; \quad (3)$$

$$V_L = \arg \max_{d \in [1,3]} (a_{dD}); \quad (4)$$

where  $a_{1D}, a_{2D}, a_{3D}$  are the linear, planar, and volumetric geometric features, respectively; and  $V_L$  is the geometric structure (i.e., linear, planar, volumetric), of the supervoxel.

Simultaneously, the normal direction,  $V_N$ , and the principal direction,  $V_P$ , of the supervoxel are obtained, corresponding to the eigenvectors of the smallest eigenvalue and the largest eigenvalue, respectively. The color,  $V_C$ , and the intensity,  $V_I$ , of the supervoxel are calculated by the sum of the colors and intensities of the points in the supervoxel divided by the number of points.

As urban objects have complicated and incomplete structures with significant disparities in size, using fixed size voxels can lead to incorrect estimation of the geometric structures of the points in

a local area, resulting in bad segmentation. For example, the points of a small supervoxel may estimate the geometric structure of the points corresponding to thin pole-like objects, such as a linear structure, correctly, but would estimate the geometric structure of the points corresponding to thick pole-like objects incorrectly, and vice versa. To overcome this, we generate the supervoxels of the point clouds with voxels at two different sizes, and then integrate the supervoxels from the different sized original voxel sizes as multi-scale supervoxels. This contradiction is not uncommon for supervoxels generated by different sized original voxels, because the points in the supervoxels of different sizes may be classified into inconsistent geometric structures. Hence, we have developed optimized strategies, as listed in Table 1, to address the contradiction.

Integrating the results from the supervoxels with two different scales using the optimizing strategies in Table 1, the points in each multi-scale supervoxel have similar geometric structures and attributes (e.g., colors, intensities), and we achieve accurate estimation of geometric structures corresponding to the points within the area with mixed object sizes and shapes.

A multi-scale supervoxel is somewhat different from a regular voxel. As illustrated in Fig. 1 (each voxel is dotted in one color), the points in the area of the pole and the tree are incorrectly allocated into one regular voxel (Fig. 1b), but are correctly separated and allocated into different supervoxels (Fig. 1c).

## 2.2. Segmenting multi-scale supervoxels

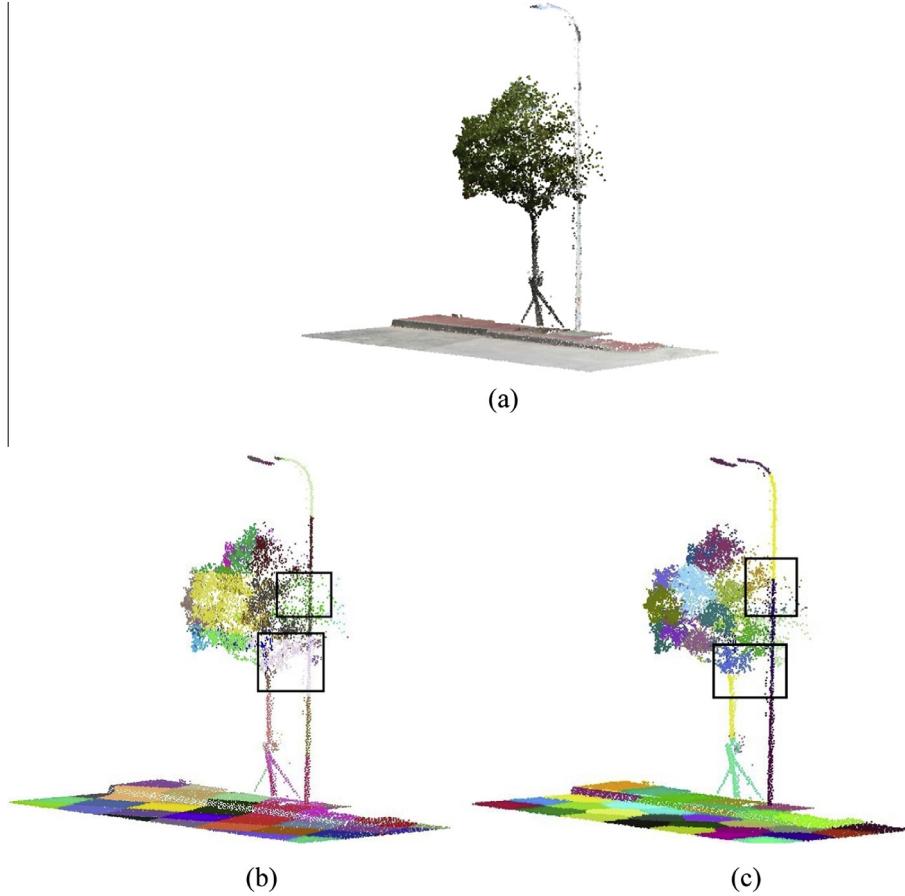
Each supervoxel is characterized by its individual properties in terms of normal vector, principal direction, color, intensity, and geometric structures (e.g., linear/planar/volumetric). All points in each supervoxel may be treated as a unit rather than individual points, making the supervoxels robust to noise and outliers, and reducing the computing cost of segmentation. Segmentation of supervoxels aims to seek the consistent property (e.g., color, normal vector, principal direction) between supervoxels. However, approaching the segmentation of the supervoxels by seeking consistency around a single cue or parameter is likely to generate partial results (Barnea and Filin, 2013). We thus extended the graph-based 2D images segmentation method (Felzenszwalb and Huttenlocher, 2004) by considering the 3D neighboring supervoxels to segment the supervoxels according to color, intensity, normal vector, and principal direction. Following the principle of 2D graph-based segmentation methods, the weights of the edges in one region is the key to linking adjacent vertices (or in the 3D case: segments).

As the supervoxels are classified by their geometric structures of their linear, planar, and/or volumetric properties, the three graphs were derived for  $G_{linear}(V, E, W)$ ,  $G_{planar}(V, E, W)$ , and  $G_{volumetric}(V, E, W)$ , corresponding to the three geometric structures,

**Table 1**  
Optimization strategy for multi-scale supervoxels.

Cases	Strategy	Description
$V_L^S = V_L^L$	Both sizes have identical geometric structure	The local area is homogeneous
$V_L^S = \text{linear}$	The geometric structure of the small sized voxel is selected	The neighborhood of large size voxels may contain points of adjacent objects
$V_L^L \neq \text{linear}$	The geometric structure of the large sized voxel is selected	Points on thick pole-like objects are misclassified as planar points when the small size neighborhood is chosen
$V_L^S = \text{planar}$	The geometric structure of the small sized voxel is selected	The neighborhood of large size may contain points of adjacent objects
$V_L^L = \text{linear}$	The geometric structure of the large sized voxel is selected	The neighborhood of large size voxels is robust to noise
$V_L^S = \text{planar}$	The geometric structure of the small sized voxel is selected	
$V_L^L = \text{volumetric}$	The geometric structure of the large sized voxel is selected	
$V_L^S = \text{volumetric}$	The geometric structure of the large sized voxel is selected	
$V_L^L \neq \text{volumetric}$		

Note:  $V_L^S, V_L^L$  are the geometric structure type of a local area at the supervoxels of small and large sizes, respectively.



**Fig. 1.** Comparison of supervoxels and regular voxels for point clouds: (a) point clouds; (b) regular voxelization; and (c) supervoxelization.

respectively; where  $V$ ,  $E$ , and  $W$  represent the vertices (supervoxels), the edges, and the weights of each edge in the graph, respectively. The weights were calculated by the angle between the principal directions, the angle between normal vectors, and the intensity or colors difference between supervoxels impinging upon a given edge, respectively. To segment one graph, the edges were ranked in increasing order of their corresponding weights, then traversed in order. Suppose the weight of one edge is less than the standard deviation of the weights between within the supervoxels impinging on that edge. In that case, the supervoxels are merged and the corresponding edge removed. Each edge in the graph is processed until all the edges (of the modified supervoxels) are traversed, resulting in the final segmentation.

The segmentation result of each graph corresponds to each type of supervoxel (linear/planar/volumetric). The final segmentation is obtained by combining the segmentation result from each graph to overcome the partial segmentation based on one cue. The properties (linear/planar/volumetric geometric structure, normal vector, principal direction, color, and intensity) associated with each final segment are calculated according to Eqs. (2)–(4). Each segment formed by the supervoxels may be a complete object or part-object. The segments need to be further merged as meaningful geometric abstractions of objects such as buildings, traffic signs, street lamps, trees and so forth. Such abstraction is beneficial and meaningful for the detection of objects from the point clouds of urban scenes (Yang and Dong, 2013). However, merging adjacent segments without considering the semantic knowledge of the objects for the abstraction results in incorrect object detection, because of the dense object mix in a local urban scene. For example, tree crowns and street lamps are often interlaced. We thus propose a

hierarchical order ranked by the saliency of the segments to merge adjacent segments into meaningful units.

### 2.3. Calculating the saliency of the segments

The non-ground points collected by MLS systems usually distribute along the vertical direction in the space. Hence, for example, vertically associated segments are considered as to have higher saliency. The saliency any segment is determined by its geometric properties as follows:

- The height of the segment,  $S_H$ , which indicates the size of the segment.
- The angle,  $S_\alpha$ , between the vertical direction and the normal vector of the segment.
- The angle,  $S_\beta$ , between the principal direction of the segment and the vertical direction.
- The number,  $S_{Num}$ , of segments neighboring the segment.

The saliency of a given segment,  $S_{Sa}$ , is calculated by

$$S_{Sa} = \frac{S_H}{\max_{k=(1,2,\dots,n)}(S_H^k)} + \left(1 - \frac{S_{Num}}{\max_{k=(1,2,\dots,n)}(S_{Num}^k)}\right) + \frac{S_\alpha}{\pi/2} + \left(1 - \frac{S_\beta}{\pi/2}\right); 2(\partial, \beta \in [0, \pi/2]), \quad (5)$$

where  $n$  is the total number of segments in the point clouds. The saliency of each segment is ranked in decreasing order, as one input to the decision for merging adjacent segments.

**Table 2**

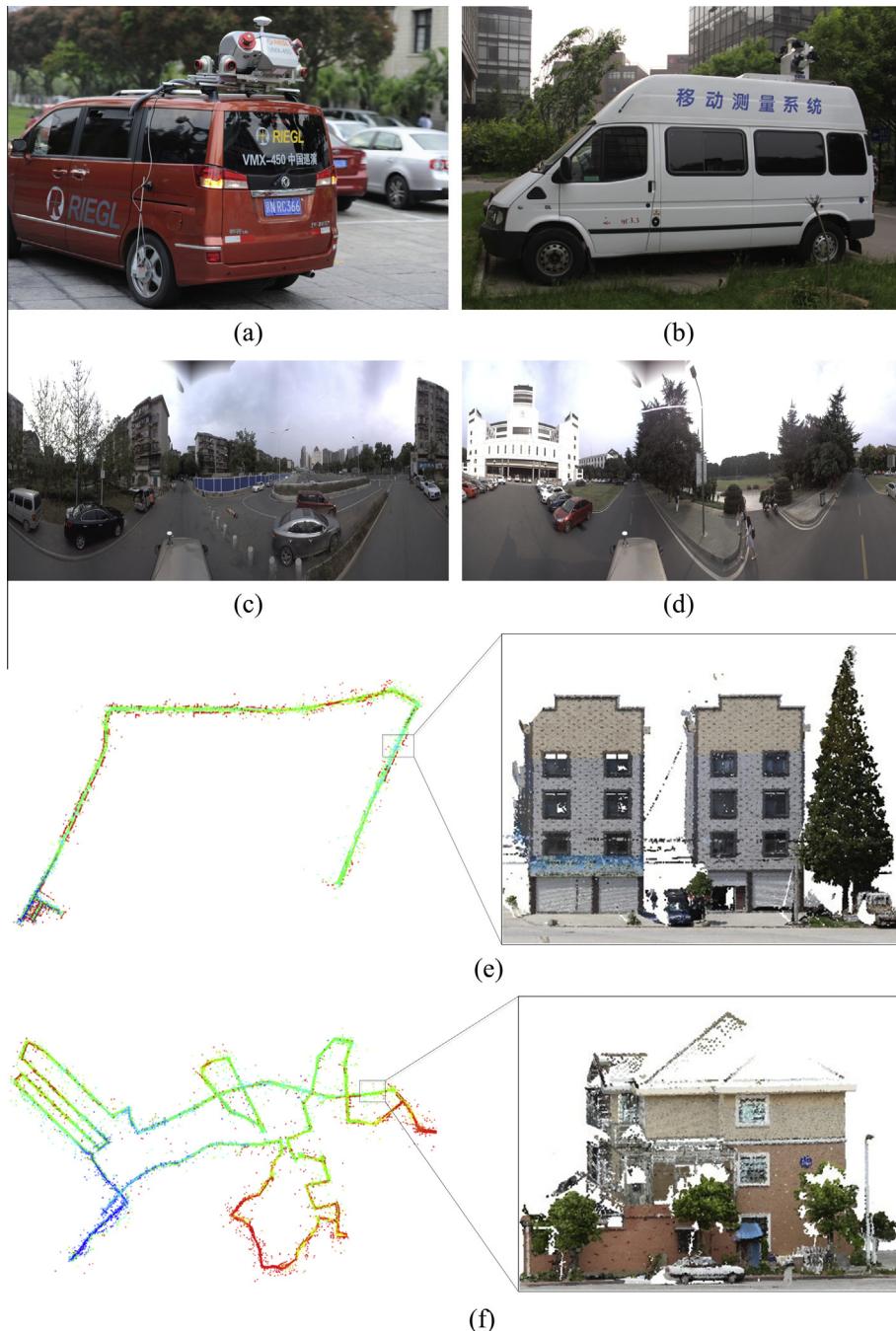
Formal representations of seven common types of urban objects.

Urban objects	Formal representation	Description	Threshold values
Buildings	$(\exists S_L^i = planar \& \& S_L^i \perp \vec{z} \& \& S_L^i > T_{S_H} \& \& S_W^i > T_{S_W})$ $\& \& (B_H \geq T_{B_H} \& \& B_W \geq T_{B_W}) \& \& B_{NOS} \geq T_{B_{NOS}}$ $\vec{z} = (0, 0, 1); S_L^i, S_N^i, S_H^i, \text{and } S_W^i$ are the geometric structure (e.g., linear, planar, volumetric), normal vector, and the height and width of the $i$ -th segment, respectively; $B_H, B_W$ are the height and width of the building, respectively; and $B_{NOS}$ is the number of segments comprising the building	A building object should have one planar segment, $i$ , perpendicular to the ground; the height and width of the $i$ -th segment, are more than $T_{S_H}$ and $T_{S_W}$ , respectively; the height and width of the building should be more than $T_{B_H}$ and $T_{B_W}$ , respectively; it should have at least $T_{B_{NOS}}$ planar segments	$T_{S_H} = 3.0 \text{ m}$ $T_{S_W} = 3.0 \text{ m}$ $T_{B_H} = 5.0 \text{ m}$ $T_{B_W} = 3.0 \text{ m}$ $T_{B_{NOS}} = 2$
Utility poles	$(\exists S_L^i = linear \& \& S_P^i // \vec{z} \& \& S_H^i > T_{S_H})$ $\& \& (\exists S_L^j = linear \& \& S_P^j \perp \vec{z} \& \& S_{Length}^j > T_{S_{Length}})$ $\& \& S_T^{ij} = down \& \& (U_H \geq T_{U_H})$ $\vec{z} = (0, 0, 1); S_L^i, S_P^i, \text{and } S_H^i$ are the geometric structure, principal direction, and height of $i$ -th segment, respectively; $S_L^j, S_P^j, \text{and } S_{Length}^j$ are the geometric structure, principal direction, and length of the $j$ -th segment; $S_T^{ij}$ is the topological relation between the $i$ -th and $j$ -th segment; and $U_H$ is the height of the utility pole	A utility pole object should have height more than $T_{U_H}$ ; one linear segment, $i$ , perpendicular to the ground; the height of the $i$ -th segment is more than $T_{S_H}$ ; it should have one linear segment, $j$ , parallel to the ground with length more than $T_{S_{Length}}$ ; the average height of the $i$ -th segment is lower than that of the $j$ -th segment	$T_{S_H} = 8.0 \text{ m}$ $T_{S_{Length}} = 10.0 \text{ m}$ $T_{U_H} = 8.0 \text{ m}$
Traffic signs	$(\exists S_L^i = linear \& \& S_P^i // \vec{z} \& \& S_H^i > T_{S_{Linear}})$ $\& \& (\exists S_L^j = planar \& \& S_N^j \perp \vec{z} \& \& S_H^j > T_{S_{Planar}})$ $\& \& S_T^{ij} = down \& \& (TS_H \geq T_{TS_H})$ $\vec{z} = (0, 0, 1); S_L^i, S_P^i, \text{and } S_H^i$ are the geometric structure, principal direction, and height of the $i$ -th segment, respectively; $S_L^j, S_N^j, \text{and } S_H^j$ are the geometric structure, normal vector, and the height and width of the $j$ -th segment, respectively; $S_T^{ij}$ is the topological relation between the $i$ -th and $j$ -th segment; and $TS_H$ is the height of the traffic sign	A traffic sign object should have one linear segment, $i$ , perpendicular to the ground; the height of the $i$ -th segment is more than $T_{S_{Linear}}$ ; it should have one planar segment, $j$ , perpendicular to ground; the height and width of the $j$ -th segment are more than $T_{S_{Planar}}^i$ and $T_{S_{Planar}}^j$ , respectively; the average height of the $i$ -th segment is lower than that of the $j$ -th segment; and the height should be more than $TS_H$	$T_{S_{Linear}} = 2.0 \text{ m}$ $T_{S_{Planar}}^i = 0.5 \text{ m}$ $T_{S_{Planar}}^j = 0.5 \text{ m}$ $TS_H = 2.5 \text{ m}$
Trees	$(\exists S_L^i = linear \& \& S_P^i // \vec{z} \& \& S_H^i > T_{S_H})$ $\& \& (\exists S_L^j = volumetric)$ $\& \& S_T^{ij} = down$ $\& \& (Tree_{H_i} \geq T_{Tree_{H_i}} \& \& Tree_{\Delta H} < 0)$ $\vec{z} = (0, 0, 1); S_L^i, S_P^i, \text{and } S_H^i$ are the geometric structure, principal direction, and height of the $i$ -th segment, respectively; $S_L^j$ and $S_C^j$ are the geometric structure and color of the $j$ -th segment, respectively; $S_T^{ij}$ is the topological relation between the $i$ -th and $j$ -th segments; $Tree_H$ is the height of the tree; and $Tree_{\Delta H}$ is the height difference between the geometrical center and the barycenter	A tree object should have one linear segment, $i$ , perpendicular to ground; the height of the $i$ -th segment is more than $T_{S_H}$ ; it should have one volumetric segment; the average height of the $i$ -th segment is less than that of $j$ -th segment; the height of the tree should be more than $Tree_{H_i}$ ; and the height difference between the geometrical center and the barycenter of the tree should be less than 0	$T_{S_H} = 1.5 \text{ m}$ $T_{Tree_{H_i}} = 2.0 \text{ m}$
Street lamps	$(\exists S_L^i = linear \& \& S_P^i // \vec{z} \& \& S_H^i > T_{S_H})$ $\& \& (SL_H > T_{SL_H})$ $\vec{z} = (0, 0, 1); S_L^i, S_P^i, \text{and } S_H^i$ are the geometric structure, principal direction, and height of the $i$ -th segment, respectively; and $SL_H$ is the height of the street lamp.	A street lamp object should have one linear segment, $i$ , perpendicular to the ground; the height of the $i$ -th segment is more than $T_{S_H}$ ; and the height of the street lamp should be more than $T_{SL_H}$	$T_{S_H} = 5.0 \text{ m}$ $T_{SL_H} = 6.0 \text{ m}$
Enclosures	$(\exists S_L^i = planar \& \& S_N^i \perp \vec{z} \& \& S_H^i > T_{S_H} \& \& S_W^i > T_{S_W})$ $\& \& (EN_H \geq T_{EN_H} \& \& EN_W \geq T_{EN_W})$ $\vec{z} = (0, 0, 1); S_L^i, S_N^i, S_H^i, \text{and } S_W^i$ are the geometric structure, principal direction, height and width of the $i$ -th segment, respectively; and $EN_H$ , and $EN_W$ are the height and width of the enclosure, respectively.	An enclosure object should have one planar segment, $i$ , perpendicular to the ground; the height and width of the $i$ -th segment are more than $T_{S_H}$ and $T_{S_W}$ , respectively; and the height and width of the enclosure should be more than $T_{EN_H}$ and $T_{EN_W}$ , respectively	$T_{S_H} = 2.0 \text{ m}$ $T_{S_W} = 3.0 \text{ m}$ $T_{EN_H} = 2.0 \text{ m}$ $T_{EN_W} = 5.0 \text{ m}$
Cars	$Car_H \in [T_{Car_H}^{\min}, T_{Car_H}^{\max}] \& \& Car_W \in [T_{Car_W}^{\min}, T_{Car_W}^{\max}]$ $\& \& Car_{Length} \in [T_{Car_{Length}}^{\min}, T_{Car_{Length}}^{\max}] \& \& Car_{NOS} \geq T_{Car_{NOS}}$  $Car_H, Car_W, Car_{Length}$ are the height, width and length of the car, respectively	A car object should have the height, width and length in a certain range $[T_{Car_H}^{\min}, T_{Car_H}^{\max}], [T_{Car_W}^{\min}, T_{Car_W}^{\max}], [T_{Car_{Length}}^{\min}, T_{Car_{Length}}^{\max}]$ ; and it should have at least $T_{Car_{NOS}}$ planar segments	$T_{Car_H}^{\min} = 1.5 \text{ m}$ $T_{Car_H}^{\max} = 5.0 \text{ m}$ $T_{Car_W}^{\min} = 1.5 \text{ m}$ $T_{Car_W}^{\max} = 3.0 \text{ m}$ $T_{Car_{Length}}^{\min} = 2.7 \text{ m}$ $T_{Car_{Length}}^{\max} = 20.0 \text{ m}$ $T_{Car_{NOS}} = 3 \text{ m}$

**Table 3**

MLS data sets from two sets of equipment.

	Number of points (millions)	Point density (points/m <sup>2</sup> )	Data length (km)	Data width (km)
VMX-450 data	202	120	3.85	2.98
SSW-MMTS data	694	77	8.15	5.02



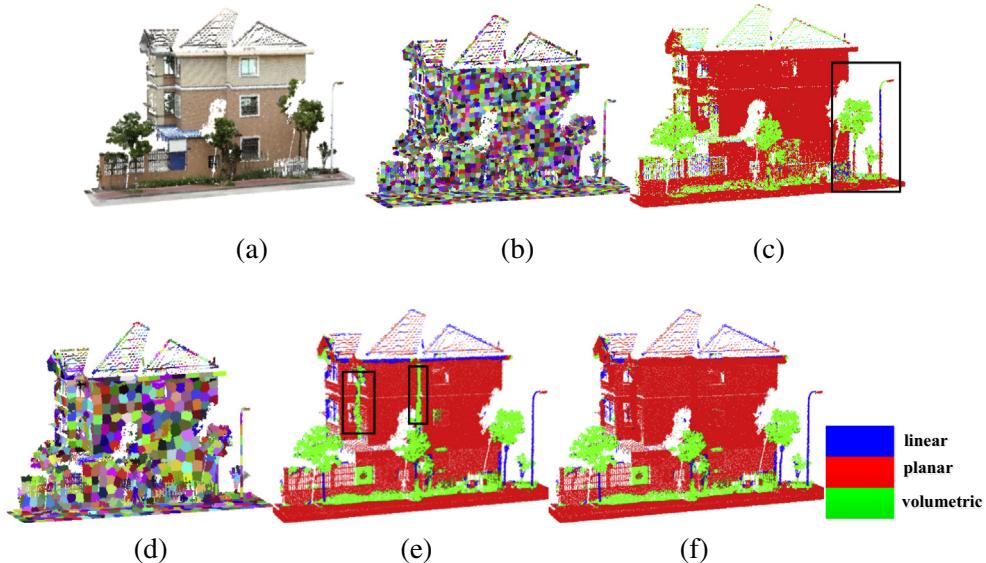
**Fig. 2.** Overview of the MLS data sets, (a) VMX-450 mobile mapping system, (b) SSW-MMTS mobile mapping system, (c) the panoramic image of typical scene in VMX-450 data set, (d) the panoramic image of typical scene in SSW-MMTS data set, (e) VMX-450 data set, and (f) SSW-MMTS data set.

#### 2.4. Merging segments with semantic knowledge of urban objects for object extraction and classification

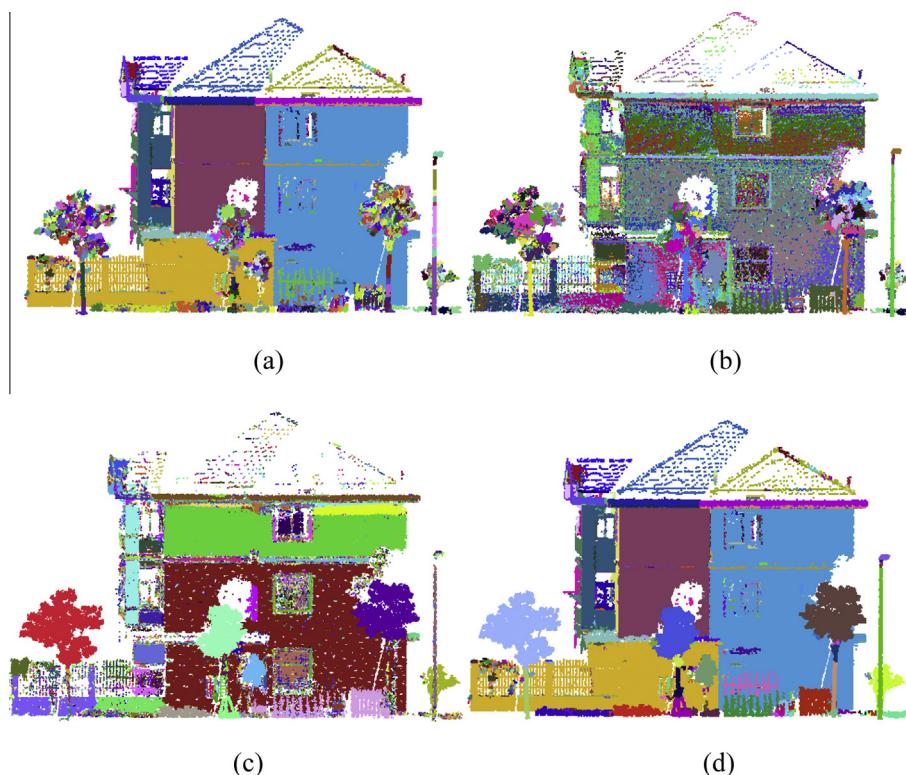
Urban objects generally have specific associated semantic knowledge. For example, poles show long and thin shapes, vegetation (e.g., trees) and cars are broad and short, and building walls show wide vertical shapes. This semantic knowledge corresponding to types of urban objects should be abstracted to merge segments for objects detection and classification. The semantic knowledge of seven types of common urban objects, namely, buildings, traffic signs, utility poles, street lamps, trees, enclosures, and cars were formed as formal representations, and are listed in Table 2.

The segment with the highest saliency is selected as the seed for the merging operation. The adjacent segments of the seed segment are merged if the minimum distance between their centroids is less than a (arbitrary) threshold value (e.g., 0.5 m). The merged segment is then selected as seed to search the adjacent segments for further merging, until no further merging occurs, forming a meaningful unit, regarded as a complete object or part-object. The size and geometric structure of each meaningful unit segment are calculated and the relations between these meaningful units are assessed.

Each meaningful unit is classified as one type of object according to the representation listed in Table 2.



**Fig. 3.** Supervoxelization of MLS point clouds: (a) original data; (b) generated voxels, small size; (c) geometric structure classification using small size voxels; (d) generated voxels, large size; (e) geometric structure classification using large size voxels; and (f) geometric structure classification integrating the results from different voxel sizes.



**Fig. 4.** Supervoxels segmentation based on single and multiple cues: (a) normal vector based segmentation; (b) principal direction based segmentation; (c) color based segmentation; and (d) three cues based segmentation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

As before, the meaningful unit with the highest saliency is chosen and merged with adjacent meaningful units according to pre-defined rules corresponding to each type of object:

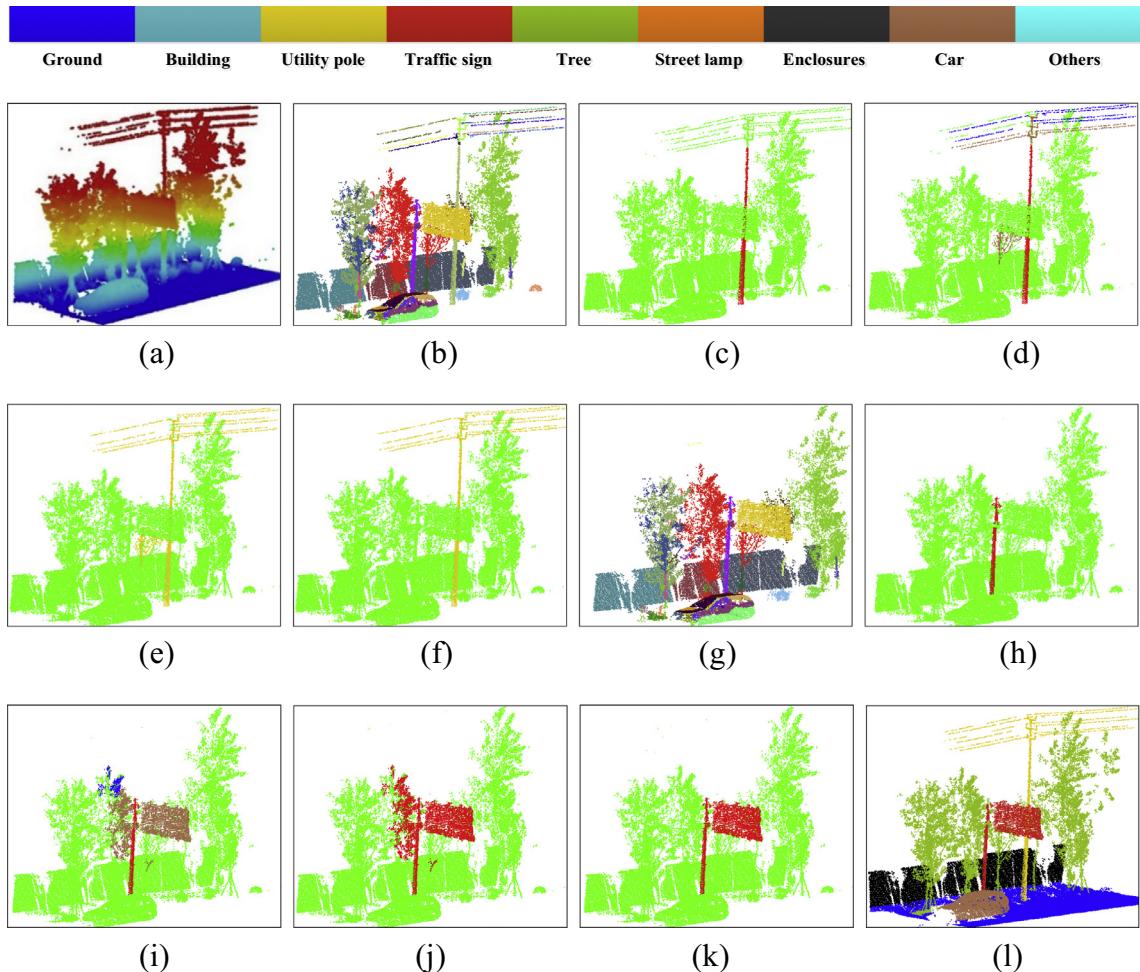
**Rule 1:** Building objects: adjacent segments of planar geometric structure are merged.

**Rule 2:** Utility poles: adjacent segments of linear geometric structure parallel or perpendicular to the ground are merged.

**Rule 3:** Traffic signs: adjacent segments of the linear or planar geometric structures are merged, but only where the planar segments are perpendicular to the ground.

**Rule 4:** Trees: adjacent segments of the linear or volumetric geometric structure are merged; but only where the 2D spatial distance between centers of the linear segment and volumetric segment is within a specified value (e.g., 1.0 m).

**Rule 5:** Streetlamps: adjacent segments of the linear geometric structure are merged.



**Fig. 5.** Extracting urban objects in a hierarchical order: (a) original data; (b) segmentation results; (c) segment with the highest saliency; (d) merging adjacent segments; (e) potential classification of the identified object; (f) classification of the identified object; (g) removal of the classified object; (h) segment of the highest (remaining) saliency; (i) merging adjacent segments; (j) potential classification of the identified object; (k) classification of the identified object; and (l) final results of urban objects extraction.

**Rule 6:** Enclosures: adjacent segments of the planar geometric structure merged, if the planar segments are perpendicular to ground.

**Rule 7:** Cars: adjacent segments of the planar geometric structure are merged.

The meaningful unit is then classified as one type of object from the above process, and the data are removed from the segmentation result. The remaining meaningful units (or segments) are then sequentially merged in decreasing saliency order, following the above description, until all segments are processed.

The representation of each object type (Table 2) describes the shapes, topology between segments, and the thresholds of shapes. For example, suppose that one cluster is classified as a building. The formal representation of building is described as

$$\text{Building} = \left( \exists S_L^i = \text{planar} \& \& S_N^i \perp \vec{z} \& \& S_H^i > 3 \& \& S_W^i > 3 \right) \\ \& \& (B_H \geq 5 \& \& B_W \geq 3 \& \& B_{NOS} \geq 2), \vec{z} = (0, 0, 1)$$

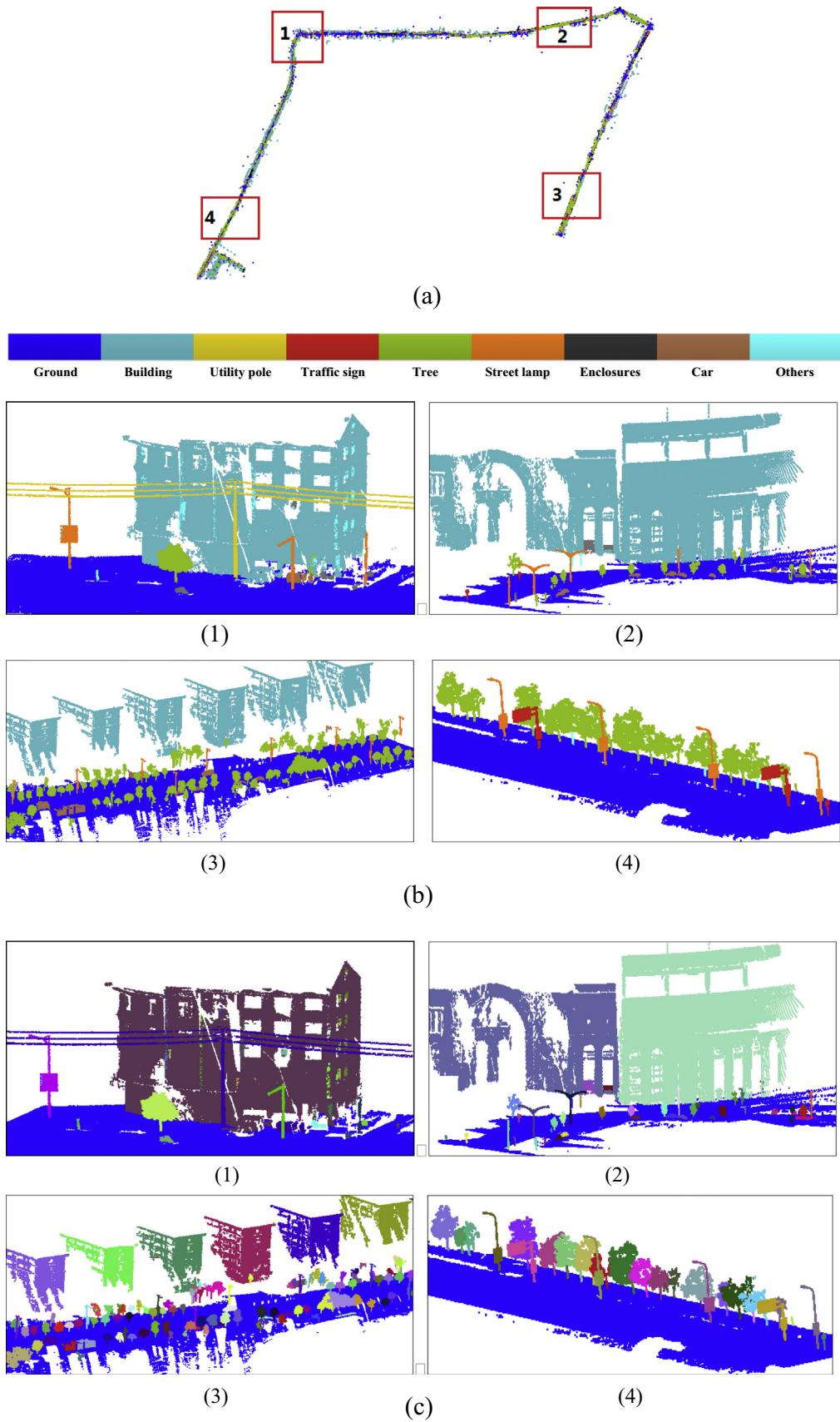
where  $S_L^i$ ,  $S_N^i$ ,  $S_H^i$ ,  $S_W^i$  indicate the geometric structure (e.g., linear, planar, volumetric), the normal vector, the height and width of the  $i$ -th segment, respectively;  $B_H$ ,  $B_W$  indicate the height and width of the building, respectively; and  $B_{NOS}$  indicates the number of segments comprising the building.

This indicates that one building object should have one planar segment,  $i$ , perpendicular to the ground. Moreover, the height and width of the  $i$ -th segment should both more than 3 m; the height and width of the building should be more than 5 m and 3 m, respectively; and one building should have at least two planar segments.

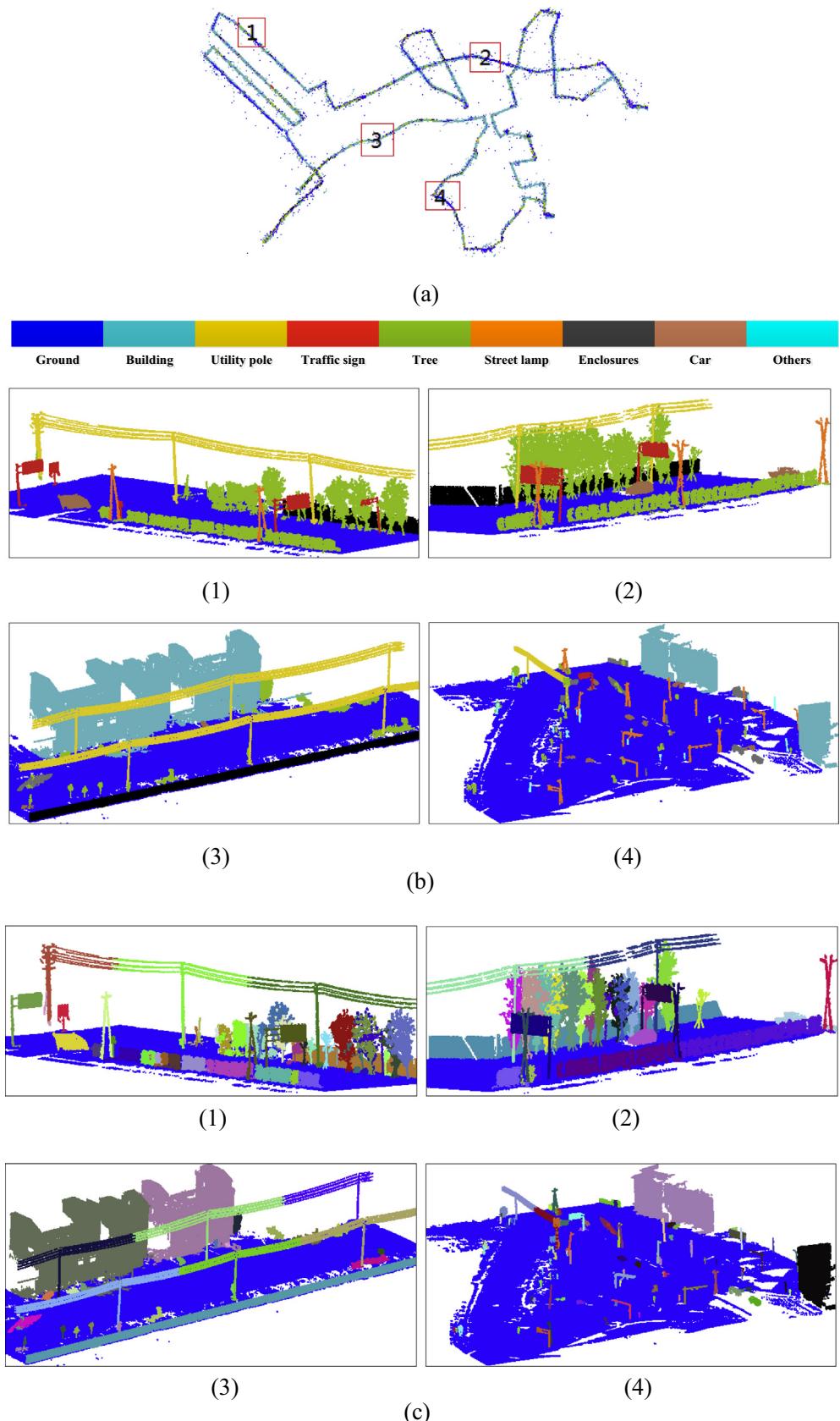
### 3. Results and analysis

#### 3.1. Data description

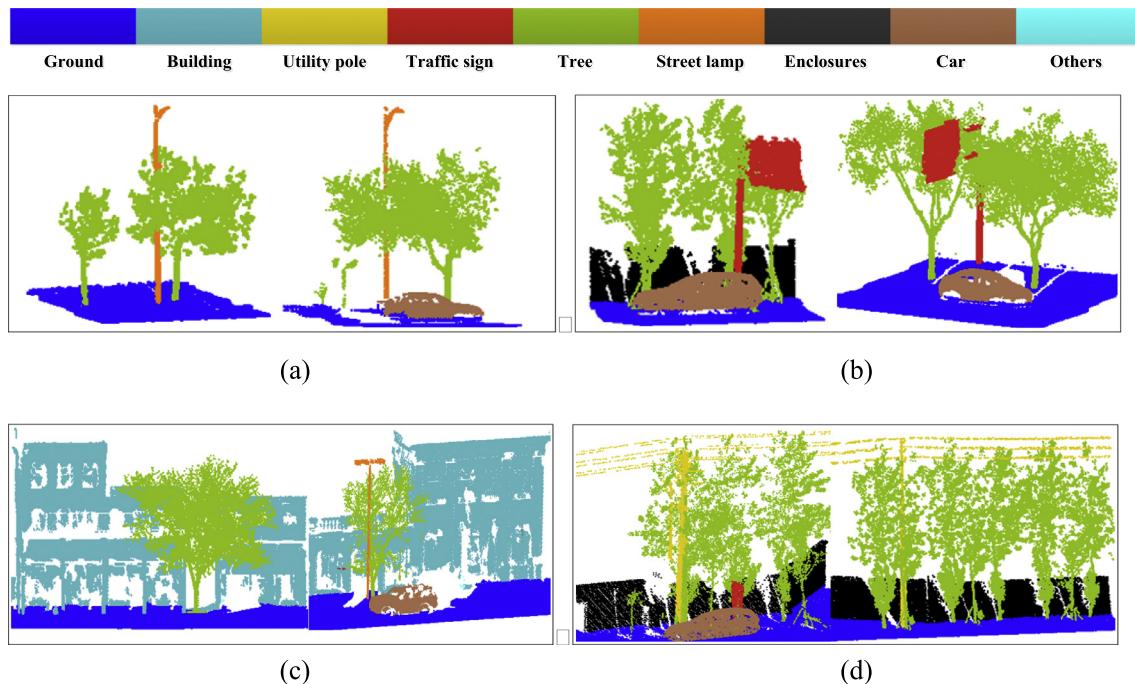
Two MLS data sets were acquired using the Riegl VMX-450 mobile mapping system and the SSW-MMTS mobile mapping system in different urban environments. The Riegl VMX-450 mobile mapping system was equipped with two Riegl VQ-450 laser scanners with a field of view (FOV) of 360°, a VMX-450-CS6 digital camera, a portable control unit box, and a GNSS/INS unit. The effective measurement rate of the VMX-450 was up to 1,100,000 points/s. The SSW-MMTS mobile mapping system was equipped with one laser scanner with the maximum range of 300 m, navigation and positioning system (IMU, GPS, DMI), and six digital cameras (22 million pixels). Table 3 lists the description of the data sets, and Fig. 2 shows the Riegl VMX-450 and SSW-MMTS mobile mapping systems, the panoramic image of typical urban scenes and an overview of the two data sets.



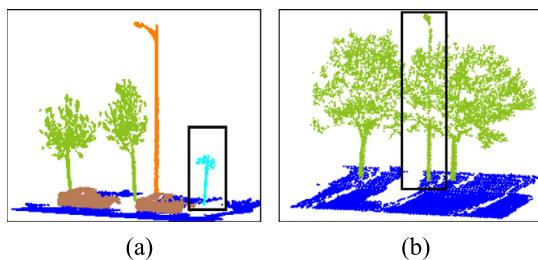
**Fig. 6.** Urban objects extraction results of VMX-450 data set: (a) urban scene; (b) classification of areas 1–4 colored by the types of objects; and (c) classification of areas 1–4 colored by the flags of objects. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 7.** Urban objects extraction results of SSW-MMTS data set: (a) urban scene; (b) classification of areas 1–4 colored by the types of objects; and (c) classification of areas 1–4 colored by the flags of objects. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 8.** Urban objects extraction from several typical scenarios: (a) streetlamps; (b) traffic signs; (c) buildings; and (d) utility poles.



**Fig. 9.** Error cases of urban objects classification: (a) trees without canopy were misclassified as others; (b) streetlamps largely occluded by trees were misclassified as trees.

### 3.2. Supervoxel generation, segmentation, and classification

According to the proposed method, voxels at two different sizes were generated for the two MLS datasets. The average point span of the collected data sets is approximately 2 cm. The small voxel size was specified as  $5 \times$  average point span, to locate more than one point in any given voxel. The large voxel size was specified as  $5 \times$  small voxel size to generate larger scale voxels.

Fig. 3 illustrates the workflow of generating supervoxels from the MLS data sets. Fig. 3a shows the MLS point clouds. Fig. 3b and c illustrate the generated voxels and the corresponding geometric structure classification at small size. Fig. 3d and e are the generated voxels and the corresponding geometric structure classification at large size. Each voxel is dotted in one color. Comparing Fig. 3c and e, it is clear that the geometric structure of many voxels was misclassified using fixed sized voxels. For example, the points of the pole-like objects were misclassified as planar with small sized voxels, but this error was corrected using large sized voxels. Our proposed method solves this misclassification for the general case by integrating the results from the different sized voxels, according to the strategies listed in Table 1 (Fig. 3f), demonstrating that integrating the results from different voxel sizes improves the quality of the geometric structure classification of point clouds, and laying a good foundation for further segmentation.

Fig. 4a–c describe the segmentation results segmented by normal vectors, principal directions, and colors, respectively. Over-segmentation and under-segmentation occurred based on a single cue (e.g., normal vectors). For example, segmentation based the normal vectors of supervoxels obtains good results for facades (Fig. 4b) but leads to over-segmentation of pole-like objects (Fig. 4a). Nevertheless, Fig. 4d illustrates good segmentation by the proposed method combining the segments of each graph. The resulting segments have better consistency with physical objects compared with those of any single-cue based segmentation.

### 3.3. Urban objects extraction results

Fig. 5 shows the workflow of extracting urban objects in a hierarchical order ranked by the saliency of the segments.

The saliency of each segment in Fig. 5b was calculated and ranked according to Eq. (5). The red segment in Fig. 5c indicates the highest saliency segment. The saliency of the highest ranking was merged (Fig. 5d) then recognized as a potential object (Fig. 5e). Fig. 5d and e show that several segments were incorrectly clustered. The red area represents the highest saliency segment, brown regions denote the segments directly adjacent with the most salient segment, blue regions indicate segments which are adjacent with the brown regions, and the green regions indicate others. However, the incorrect merging was corrected following the defined merging rules of urban objects (Fig. 5f). This shows that semantic knowledge based classification integrates well with the hierarchical strategy to correct improper merging, thus producing good quality object detection and classification.

Following the hierarchical extraction strategy, the objects in the segments were extracted one by one, as illustrated in Fig. 5b–k, until all the objects are extracted (Fig. 5l).

Figs. 6 and 7 show the extracted urban objects from the Riegl VMX-450 and SSW-MMTS MLS datasets, respectively. Figs. 6c and 7b–c illustrate the extraction results from different areas of the two MLS datasets, respectively, colored by object types and flags. Comparing between the extracted objects color by object types and flags, the proposed method correctly extracts urban

**Table 4**

Precision, recall rate, and overall accuracy of the proposed method for urban objects extraction.

Data	Object	Building	Utility pole	Traffic signs	Tree	Street lamp	Enclosure	Car	Others	Classification overall	Precision %	Recall %
VMX-450 data	Building	227	0	0	0	0	1	0	5	233	97.4	98.3
	Utility pole	0	186	0	0	2	0	0	9	197	94.4	93.9
	Traffic signs	0	2	157	1	3	0	0	3	166	94.6	95.2
	Tree	0	3	1	2583	2	0	0	248	2837	91.0	91.0
	Street lamp	0	1	3	2	838	0	0	52	896	93.5	94.1
	Enclosure	2	0	0	0	0	43	1	0	46	93.5	95.6
	Car	0	0	0	0	0	0	294	25	319	92.2	90.2
	Others	2	6	4	252	46	1	31	2906	3248	89.5	89.5
True overall		231	198	165	2838	891	45	326	3248	7942	Overall accuracy: 91.1%	
SSW-MMTS data	Building	4082	0	0	0	0	1	2	102	4187	97.5	98.0
	Utility pole	0	1634	2	0	1	0	0	91	1728	94.6	94.2
	Traffic signs	0	2	497	1	2	0	0	22	524	94.8	95.6
	Tree	0	9	4	6281	12	0	0	581	6887	91.2	90.5
	Street lamp	0	4	2	8	2290	0	0	154	2458	93.2	94.3
	Enclosure	2	0	0	0	0	27	0	0	29	93.1	96.4
	Car	0	0	0	0	0	0	2285	167	2452	93.2	91.0
	Others	80	86	15	650	124	0	224	11,266	12,445	90.5	90.9
True overall		4164	1735	520	6940	2429	28	2511	12,383	30,710	Overall accuracy: 92.3%	

**Table 5**

Performance comparison between the proposed method and that of [Yang and Dong \(2013\)](#).

	Yang and Dong (2013)		The proposed method	
	Precision %	Recall %	Precision %	Recall %
Utility poles	90.6	91.4	94.4	93.9
Traffic signs	89.2	90.2	94.6	95.2
Trees	85.4	84.2	91.0	91.0
Streetlamp	86.3	89.6	93.5	94.1
Overall accuracy (%)	87.8		91.1	
Time performance (min)	478		243	

objects from the MLS datasets with both numbers and types. See, for example, the dense trees along the roadway ([Fig. 6b](#)).

[Fig. 8](#) shows that the proposed method demonstrates good performance in extracting and classifying urban objects from complicated urban scenes, where dense objects are mixed in a local area, even in the cluttered situation of occlusion and overlapping between neighboring objects. Several different types of urban objects, such as streetlamps ([Fig. 8a](#)), traffic signs ([Fig. 8b](#)), buildings ([Fig. 8c](#)), and utility poles ([Fig. 8d](#)), are correctly extracted. This establishes that the predefined Rules 1–7 combine well with the formal representation of urban objects to allow combination of adjacent segments as meaningful units, providing good performance in the extraction of urban objects in mixed object images.

However, there remain a few incorrect classifications. For example: trees without canopy were misclassified as others ([Fig. 9a](#)); a few streetlamps, largely occluded by the trees, were misclassified as trees ([Fig. 9b](#)).

#### 3.4. Quantitative evaluation of urban objects extraction

To evaluate the performance of the proposed method for detecting urban objects, the extracted urban objects in the two MLS data sets were compared with those manually marked. [Table 4](#) lists the accuracy and recall values of urban objects extraction from the two MLS data sets. The proposed method achieves good performance in extracting urban objects of buildings, ground, streetlamps, trees, telegraph poles, traffic signs, cars, and enclosures, with an overall accuracy of 91.1% and 92.3% for the two data sets, respectively.

A quantitative comparison was undertaken between the proposed method and that of [Yang and Dong \(2013\)](#), and the relevant precision, recalls, and computing cost are shown in [Table 5](#).

The proposed method has better precision, recall and overall accuracy of urban objects extraction than does the method of

[Yang and Dong \(2013\)](#). In particular, the proposed method, utilizing voxels rather than points for segmentation, shows a large improvement in computing efficiency.

#### 4. Conclusion

In this paper, we propose an automated method to extract urban objects from MLS over-segmentations of urban scenes. The proposed method generates multi-scale supervoxels from MLS point clouds and segments these supervoxels rather than individual points, greatly reducing computing costs. The proposed method defines a set rules for merging adjacent segments according to the saliency of segments and the types of urban objects, and forms a formal representation of semantic knowledge of several types of common urban objects (e.g., buildings, trees, lamps), resulting in good classification and extraction of urban objects. Experiments demonstrate that the proposed method greatly improves both time efficiencies and the robustness of urban objects extraction and classification, with overall accuracy of better than 91%.

The proposed method extracts urban objects in a hierarchical order ranked by the saliency of the segments, resulting in the reliable extraction and classification of multiple mixed objects in a local area. However, there remains room for further improvement in extracting more complicated urban objects, such as buildings with unstructured parts (e.g., free-form surfaces) and overpasses.

#### Acknowledgements

Work described in this paper was jointly supported by National Basic Research Program of China (No. 2012CB725301) and the NSFC project (No. 41071268).

## Appendix A

The mathematical formulations of transforming RGB to CIE Lab.

$$\begin{cases} r = \text{gamma}\left(\frac{R}{255.0}\right) \\ g = \text{gamma}\left(\frac{G}{255.0}\right), \\ b = \text{gamma}\left(\frac{B}{255.0}\right) \end{cases} \quad (\text{A1})$$

where  $\text{gamma}(x) = \begin{cases} \left(\frac{x+0.055}{1.055}\right)^{2.4} & x > 0.04045 \\ \frac{x}{12.92} & \text{else} \end{cases}$  and  $\begin{cases} R \in [0, 255] \\ G \in [0, 255] \\ B \in [0, 255] \end{cases}$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = 100 * [M] * \begin{bmatrix} r \\ g \\ b \end{bmatrix}, \quad (\text{A2})$$

$$\text{where } [M] = \begin{bmatrix} 0.436052025, 0.385081593, 0.143087414 \\ 0.222491598, 0.716886060, 0.060621486 \\ 0.013929122, 0.097097002, 0.714185470 \end{bmatrix}$$

$$\begin{cases} L = 116 * f\left(\frac{Y}{Y_n}\right) - 16 \\ a = 500 * \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right], \\ b = 200 * \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \end{cases} \quad (\text{A3})$$

where  $f(t) = \begin{cases} t^{\frac{1}{3}} & t > 0.008856 \\ 7.787 * t + 16/116 & \text{else} \end{cases}$ , and the  $(X_n, Y_n, Z_n)$  are the tristimulus values of the reference white point.

## References

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 2274–2282.
- Aijazi, A.K., Checchin, P., Trassoudaine, L., 2013. Segmentation based classification of 3D urban point clouds: a super-voxel based approach with evaluation. *Remote Sens.* 5, 1624–1650.
- Barnea, S., Filin, S., 2013. Segmentation of terrestrial laser scanning data using geometry and image information. *ISPRS J. Photogramm. Remote Sens.* 76, 33–48.
- Boyko, A., Funkhouser, T., 2011. Extracting roads from dense point clouds in large scale urban environment. *ISPRS J. Photogramm. Remote Sens.* 66, S2–S12.
- Cabo, C., Ordoñez, C., García-Cortés, S., Martínez, J., 2014. An algorithm for automatic detection of pole-like street furniture objects from mobile laser scanner point clouds. *ISPRS J. Photogramm. Remote Sens.* 87, 47–56.
- Felzenszwalb, P.F., Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. *Int. J. Comput. Vision* 59 (2), 167–181.
- Guan, H., Li, J., Yu, Y., Wang, C., Chapman, M., Yang, B., 2014. Using mobile laser scanning data for automated extraction of road markings. *ISPRS J. Photogramm. Remote Sens.* 87, 93–107.
- Hernández, J., Marcotegui, B., 2009a. Filtering of artifacts and pavement segmentation from mobile lidar data. *Int. Arch. Photogramm. Remote Sens. Spatial Inform. Sci.* 38 (Part3/W8), 329–333.
- Hernández, J., Marcotegui, B., 2009b. Point cloud segmentation towards urban ground modeling. In: URBAN'09, 5th GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, Shanghai, China, 20–22 May 2009, pp. 5 (on CDROM).
- Jochem, A., Höfle, B., Rutzinger, M., 2011. Extraction of vertical walls from mobile laser scanning data for solar potential assessment. *Remote Sens.* 3, 650–667.
- Lari, Z., Habib, A., 2013. New approaches for estimating the local point density and its impact on lidar data segmentation. *Photogramm. Eng. Remote Sens.* 79, 195–207.
- Lehtomäki, M., Jaakkola, A., Hyppä, J., Kukko, A., Kaartinen, H., 2010. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sens.* 2, 641–664.
- Li, D., Elberink, S.O., 2013. Optimizing detection of road furniture (pole-like objects) in mobile laser scanner data. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inform. Sci.* 1, 163–168.
- Lim, E.H., Suter, D., 2008. Multi-scale conditional random fields for over-segmented irregular 3D point-clouds classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 23–28 June, Anchorage, AK, vol. 1–3, pp. 856–862.
- Lim, E.H., Suter, D., 2009. 3D terrestrial lidar classifications with super-voxels and multi-scale conditional random fields. *Comput. Aided Des.* 41 (10), 701–710.
- Monnier, F., Vallet, B., Soheilian, B., 2012. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. In: Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Annals), Melbourne, Australia, vol. 25, pp. 245–250.
- Pu, S., Vosselman, G., 2009. Building facade reconstruction by fusing terrestrial laser points and images. *Sensors* 9, 4525–4542.
- Pu, S., Rutzinger, M., Vosselman, G., Oude Elberink, S., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS J. Photogramm. Remote Sens.* 66, S28–S39.
- Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. *ISPRS J. Photogramm. Remote Sens.* 93, 243–255.
- Yao, W., Hinz, S., Stilln, U., 2009. Object extraction based on 3D-segmentation of lidar data by combining mean shift with normalized cuts: two examples from urban areas. In: Joint Urban Remote Sensing Event, 20–22 May, pp. 1–6.
- Yang, B., Dong, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* 81, 19–30.
- Yang, B., Fang, L., Li, Q., Li, J., 2012. Automated extraction of road markings from mobile LiDAR point clouds. *Photogramm. Eng. Remote Sens.* 78, 331–338.
- Yang, B., Fang, L., Li, J., 2013a. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* 79, 80–93.
- Yang, B., Wei, Z., Li, Q., Li, J., 2013b. Semiautomated building facade footprint extraction from mobile LiDAR point clouds. *IEEE Geosci. Remote Sens. Lett.* 10, 766–770.
- Zhu, L., Hyypä, J., 2014. The use of airborne and mobile laser scanning for modeling railway environments in 3D. *Remote Sens.* 6, 3075–3100.