ISPRS Journal of Photogrammetry and Remote Sensing 113 (2016) 106–123

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data

Yongtao Yu [a,b], Jonathan Li [a,c,*], Chenglu Wen [a], Haiyan Guan [d], Huan Luo [a], Cheng Wang [a]

[a] Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen University, Xiamen, FJ 361005, China
[b] Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian, JS 223003, China
[c] Department of Geography & Environmental Management, University of Waterloo, Waterloo, ON N2L3G1, Canada
[d] College of Geography & Remote Sensing, Nanjing University of Information Science and Technology, Nanjing, JS 210044, China

ARTICLE INFO

Article history:
Received 6 September 2015
Received in revised form 10 January 2016
Accepted 11 January 2016

Keywords:
Bag-of-visual-phrases
Deep Boltzmann machine (DBM)
Mobile laser scanning (MLS)
Point cloud
Traffic sign detection
Traffic sign recognition (TSR)

ABSTRACT

This paper presents a novel algorithm for detection and recognition of traffic signs in mobile laser scanning (MLS) data for intelligent transportation-related applications. The traffic sign detection task is accomplished based on 3-D point clouds by using bag-of-visual-phrases representations; whereas the recognition task is achieved based on 2-D images by using a Gaussian-Bernoulli deep Boltzmann machine-based hierarchical classifier. To exploit high-order feature encodings of feature regions, a deep Boltzmann machine-based feature encoder is constructed. For detecting traffic signs in 3-D point clouds, the proposed algorithm achieves an average recall, precision, quality, and *F*-score of 0.956, 0.946, 0.907, and 0.951, respectively, on the four selected MLS datasets. For on-image traffic sign recognition, a recognition accuracy of 97.54% is achieved by using the proposed hierarchical classifier. Comparative studies with the existing traffic sign detection and recognition methods demonstrate that our algorithm obtains promising, reliable, and high performance in both detecting traffic signs in 3-D point clouds and recognizing traffic signs on 2-D images.

1. Introduction

Traffic signs provide road users with correct, detailed road information, thereby ensuring road users to rapidly approach their destinations. Traffic signs also function to regulate and control traffic activities, thereby ensuring traffic safety and traffic smoothness. To facilitate management and improve efficiency, an effective and automated traffic sign recognition system is urgently demanded by transportation agencies to monitor the status and measure the usability of traffic signs. In addition, the accurate functionality and localization information of traffic signs also provides important inputs to many intelligent transportation-related applications, such as driver assistance and safety warning systems (Zheng et al., 2004; Cheng et al., 2007) and autonomous driving (Choi et al., 2012; Broggi et al., 2013; Seo et al., 2015). Specifically, the absence or lack of visibility of necessary traffic signs might cause inconvenience to road users, sometimes even cause terrible traffic accidents and casualties. Therefore, exploring and developing effective, automated traffic sign detection and recognition techniques are essential to the transportation agencies to rapidly update traffic sign inventory and improve traffic quality and safety.

Traditionally, the measurement and maintenance of traffic signs were basically accomplished through field work, where field workers from transportation agencies conducted on-site inspections and maintenance on a regular basis. In fact, such field work was time consuming, labor intensive, costly, and inefficient to operate large-scale, complicated road networks. Recently, with the advance of optical imaging techniques, mobile mapping systems (MMS) using digital or video camera(s) (Murray et al., 2011; Brogan et al., 2013) have emerged as an effective tool for a wide range of transportation applications. The images collected by a camera-based MMS have provided a promising data source for rapid detection and recognition of traffic signs along roadways. However, the MMS images suffer greatly from object distortions, motion blurs, noises, and illumination variations. In addition, caused by viewpoint variations, traffic signs are sometimes occluded or partially occluded by the nearby objects (e.g., trees) in the images. Therefore, it is still a great challenge to achieve high-quality, high-accuracy, and automated detection and recognition of traffic signs from MMS images.

* Corresponding author. Tel.: +1 519 8884567; fax: +1 519 7460658.
E-mail addresses: junli@xmu.edu.cn, junli@uwaterloo.ca (J. Li).

http://dx.doi.org/10.1016/j.isprsjprs.2016.01.005
0924-2716/

Since last decade, benefitting from the integration of laser scanning and position and orientation technologies, mobile laser scanning (MLS) systems have been designed and extensively used in fields of transportation, road feature inventory, computer games, cultural heritage documentation, and basic surveying and mapping (Williams et al., 2013). MLS systems can rapidly acquire highly dense and accurate 3-D point clouds along with color imagery. The 3-D point clouds provide accurate geometric and localization information of the objects; whereas the color imagery provides detailed texture and content information of the objects. Therefore, by fusing imagery and 3-D point clouds, MLS systems provide a promising solution to traffic sign detection (based on 3-D point clouds) and recognition (based on imagery).

In this paper, we present a novel algorithm combining bag-of-visual-phrases (BoVPs) and hierarchical deep models for detecting and recognizing traffic signs from MLS data. The proposed algorithm includes three stages: visual phrase dictionary generation, traffic sign detection, and traffic sign recognition. At the visual phrase dictionary generation stage, the training MLS data are supervoxelized to construct feature regions for generating a visual word vocabulary, and to construct spatial word patterns for generating a visual phrase dictionary. At the traffic sign detection stage, individual semantic objects are first segmented, supervoxelized, featured, and quantized to form BoVPs representations. Then, traffic signposts are detected based on similarity measures between the BoVPs of the query object and the semantic objects. Finally, traffic signs are located and segmented via percentile-based analysis. At the traffic sign recognition stage, a Gaussian-Bernoulli deep Boltzmann machine (DBM) based hierarchical classifier is applied to the registered traffic sign regions to recognize traffic signs.

The main contributions of this paper are as follows: (1) a DBM-based feature encoder is proposed to generate high-order feature encodings of feature regions; (2) a supervoxel-based BoVPs model is proposed to depict point cloud objects; (3) an extended voxel-based normalized cut segmentation method is developed to segment overlapped semantic objects. In this paper, "high-order feature" denotes the high-level abstraction of a set of features or the "feature of features"; whereas "low-order feature" denotes a specific single feature.

2. Existing methods

In the following sections, we present a detailed review of existing methods for traffic sign detection based on MLS point clouds and traffic sign recognition based on images.

2.1. Traffic sign detection

Currently, most existing methods for traffic sign detection in MLS point clouds are based on their prior knowledge, including position, shape, and laser reflectivity. Given the fact that traffic signs are placed close to the boundaries of the road, Chen et al. (2007) proposed a processing chain of cross section analysis, individual object segmentation, and linear structure inference to detect traffic signs. To exploit pole-like structures of traffic signs, Yokoyama et al. (2011) used a combination of Laplacian smoothing and principal component analysis (PCA), where Laplacian smoothing functioned to smooth each point cloud segment to suppress measurement noise and point distribution bias; whereas PCA was performed on the smoothed segments to infer pole-like objects. Pu et al. (2011) proposed to detect traffic signs based on percentile analysis and planar shape analysis. A 3-D object matching framework was developed by Yu et al. (2015b) for detecting traffic signs of varying shapes, completeness, or hidden in trees. In addition, Hough forest methods (Wang et al., 2014, 2015), shape-based method (Golovinskiy et al., 2009b), mathematical morphology and supervised learning method (Serna and Marcotegui, 2014), and LiDAR and vision-based real-time traffic sign detection method (Zhou and Deng, 2014) were also developed for traffic sign detection.

To present clear traffic signals, traffic signs are made by highly reflective materials. As a result, traffic signs usually exhibit high retro-reflectivity (in a form of intensity) in the MLS point clouds. Such intensity information becomes an important clue for distinguishing traffic signs from other pole-like objects (Ai and Tsai, 2015). Considering pole-like properties of traffic signs, Wen et al. (2015) proposed an intensity-based pole-like object detection method. This method first removed ground points from the scene and segmented off-ground points into isolated objects; then, traffic signs were extracted based on eigenvalue analysis and object-based intensity filtering. In addition, Chen et al. (2009) detected traffic signs by using a random sampling consensus based method. Similarly, Vu et al. (2013) developed a template-driven method to detect traffic signs with the prior knowledge of symmetric shapes and highly reflective planes perpendicular to the direction of travel.

2.2. Traffic sign recognition

Generally, to alert road users and regulate traffic activities, traffic signs are usually designed with specific colors, shapes, and distinguishing contents. Such prior information provides important clues to recognize the functionalities of different traffic signs. Greenhalgh and Mirmehdi (2012) proposed to detect traffic signs by using maximally stable extremal regions (MSERs), which were robust to variations in contrast and lighting conditions. Based on different-sized histogram-of-oriented-gradient descriptors, classifiers including K-D trees, random forests, and support vector machines (SVMs) were evaluated to conduct traffic sign recognition (Zaklouta and Stanciulescu, 2012). A multi-view scheme, combining 2D and 3D analysis, was proposed for traffic sign detection, recognition, and 3D localization (Timofte et al., 2014). Cireşan et al. (2012) developed a multi-column deep neural network for traffic sign recognition. To handle various appearances and model between-class dissimilarities, sparse representation based graph embedding (SRGE) was developed for traffic sign recognition (Lu et al., 2012). Yuan et al. (2014) proposed a color global and local oriented edge magnitude pattern (Color Global LOEMP), which can effectively combine color, global spatial structure, global direction structure, and local shape information, as well as balancing distinctiveness and robustness. In addition, a detailed evaluation of different features and different classifiers was performed in Mathias et al. (2013) for traffic sign recognition purpose.

To effectively deal with the high variability of sign appearance in uncontrolled environments, an error-correcting output code framework was proposed to recognize traffic signs (Baró et al., 2009). Similarity measures were used to alleviate the shortcomings and improve the performance of template-matching-based traffic sign recognition methods (Ruta et al., 2010; Paclík et al., 2006). In addition, segmentation-based methods, such as color-space thresholding (Cheng et al., 2001; Gómez-Moreno et al., 2010) and chromatic/achromatic decomposition (de la Escalera et al., 1997; Fang et al., 2003; Fleyeh, 2006), were also exploited for traffic sign recognition. Recently, some other methods, including convolutional neural network (CNN) (Jin et al., 2014), feature-based methods (Møgelmose et al., 2015; Greenhalgh and Mirmehdi, 2015), geometric matching methods (Xu, 2009), vision-based methods (Møgelmose et al., 2012; González et al., 2011), multi-view classification (Hazelhoff et al., 2014), eigen-based method (Fleyeh and Davami, 2011), supervised low-rank matrix recovery model (Pei et al., 2013), decision fusion and reasoning module (Meuter et al.,

2011), and correlating Fourier descriptors (Larsson et al., 2011), were also exploited for traffic sign recognition.

3. Visual phrase dictionary generation

In this section, we present the technical and implementation details for the supervoxel-based visual phrase dictionary generation from MLS point clouds. Such a visual phrase dictionary can be further used to construct BoVPs for depicting 3-D point cloud objects.

3.1. Feature region generation

To generate the visual phrase dictionary, we randomly select a group of training data, each of which has a road segment of approximately 50 m, from the collected MLS point clouds. Due to the scanning properties of MLS systems in direct ground views and high laser measurement rates, ground points usually account for a great portion of the resultant point clouds in a survey scene. Such large-volume ground points almost exist in all scenes and contribute very little to the generation of the visual phrase dictionary, since the traffic signs to be detected are located off the ground. Therefore, to effectively narrow researching regions and enhance the distinctiveness of the visual phrase dictionary, a preprocessing is first performed on the training data to remove ground points. In this paper, a voxel-based upward growing approach is used to rapidly and effectively label a point cloud into ground and off-ground points (Yu et al., 2015b). This approach has the capabilities of effectively handling large scenes with strong ground fluctuations and preserving the completeness of off-ground objects from their bottoms.

Next, the training data are over-segmented into supervoxels using the voxel cloud connectivity segmentation (VCCS) algorithm (Papon et al., 2013). In the VCCS algorithm, there are two important parameters: voxel resolution and seed resolution. The voxel resolution determines the operable unit of the voxel-cloud space; whereas the seed resolution is used to select seed points for constructing initial supervoxels. In this paper, we set the voxel resolution and seed resolution at 0.05 m and 0.1 m, respectively. Then, to model the neighboring relationships among the supervoxels, an adjacency graph $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$ is constructed for all supervoxels in each training sample (see Fig. 1(a)). In the adjacency graph $\boldsymbol{G}$, the vertices $\boldsymbol{V} = \{v_i\}$ are represented by the supervoxel centers; the edges $\boldsymbol{E} = \{e_{ij}\}$ are directly connected between each pair of neighboring supervoxels. Based on the adjacency graph, for each supervoxel centered at v, its associated feature region is constructed by including supervoxel v and its first-order neighbors on the adjacency graph (see Fig. 1(a)). Here, the first-order neighbors of supervoxel v are the supervoxels directly connected to supervoxel v on the adjacency graph. As demonstrated in Wang et al. (2015), such a feature region generation strategy by embedding first-order supervoxel neighbors achieves higher saliencies and distinctiveness than directly treating single supervoxels as feature regions.

3.2. Feature region description

Most of existing 3-D descriptors are developed to exploit local low-order geometric or statistical features of individual feature points (Körtgen et al., 2003; Rusu et al., 2008, 2009). Here, "low-order feature" denotes a specific single feature. However, only few studies have focused on analyzing entire distribution features of local point cloud regions. Recently, deep learning techniques have been attracting increasing attention for their superior capabilities to exploit hierarchical, deep feature representations (Carneiro and Nascimento, 2013; Chen et al., 2013; Salakhutdinov et al., 2013). Among the deep learning models, deep Boltzmann machines (DBMs) are proven to be a powerful, robust, and highly distinctive feature generation model (Salakhutdinov et al., 2013). Thus, in this paper, we propose a DBM-based feature encoder to generate high-order feature encodings of feature regions. Each feature region v is described by a feature vector composed of two components $P_v = (\boldsymbol{f}_v, I_n)$, where the first component $\boldsymbol{f}_v$ represents a high-order geometric feature representation generated by the DBM-based feature encoder; the second component $I_n \in [0, 1]$ is a texture feature represented by the interpolated normalized intensity of feature region v. By concatenating these two features, P_v is capable of depicting both geometric and texture features of feature region v. The interpolated normalized intensity I_n is computed based on the normalized intensities of the points in feature region v as follows:

$$I_n = \frac{\sum_{k=1}^{n_v} w_k I_k}{\sum_{k=1}^{n_v} w_k} \tag{1}$$

where n_v is the number of points in feature region v; $I_k \in [0, 1]$ is the normalized intensity of the kth point in feature region v; w_k is the intensity weight of I_k and it is computed as follows:

$$w_k = \frac{I_k - I_{\min}}{I_{\max} - I_{\min}} \tag{2}$$

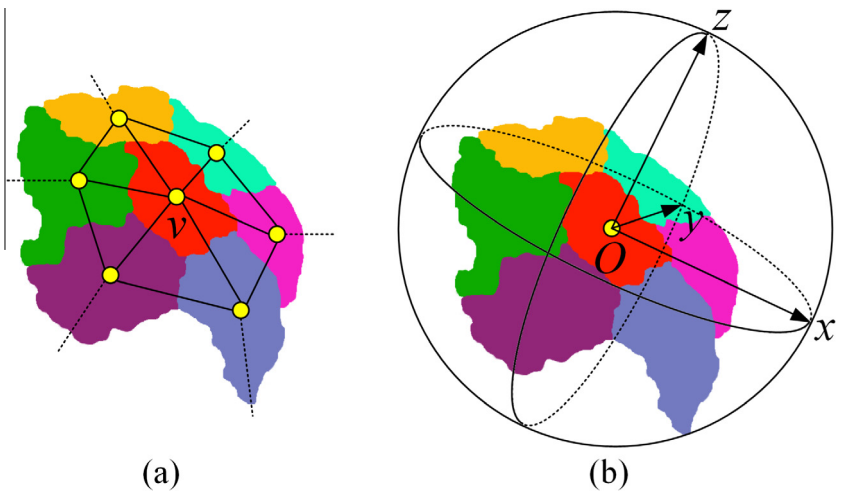

Fig. 1. Illustrations of (a) adjacency graph construction and feature region generation, and (b) bounding sphere and local reference frame construction of a feature region.

where I_{min} and I_{max} are the minimum and maximum normalized intensities in feature region v, respectively. In this way, the points with higher normalized intensities contribute more to the calculation of I_n.

Next, we present a detailed construction of the DBM-based feature encoder. First, for a feature region v, a bounding sphere centered at v is calculated. Then, a local reference frame (LRF) (Tombari et al., 2010) centered at v is constructed to deal with rotation variations of feature regions (see Fig. 1(b)). To this end, first, we construct a scatter matrix for the points in feature region v as follows:

$$\mathbf{S}_{3\times3} = \frac{1}{n_v}\sum_{k=1}^{n_v}(p_k - c_v)(p_k - c_v)^{\mathrm{T}} \tag{3}$$

where c_v denotes the center of feature region v; p_k is a point in feature region v. Then, through eigenvalue decomposition on $\mathbf{S}_{3\times3}$, we obtain three eigenvalues λ_1, λ_2, and λ_3 ($\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$) and the associated eigenvectors e_1, e_2, and e_3. Finally, the LRF is defined based on the two unique unambiguous orthogonal eigenvectors (i.e., e_1 and e_3) corresponding to the largest and the smallest eigenvalues as follows:

$$\mathrm{LRF} = \{x, y, z\} = \{e_1, e_3 \times e_1, e_3\} \tag{4}$$

To effectively model the spatial geometrical distributions of a feature region, we propose a polar sphere partition strategy to partition a feature region into a set of bins based on the constructed LRF and the bounding sphere. As shown in Fig. 2, the proposed polar sphere partition strategy is composed of three partition models: sector, polarity, and shell partition models.

The partition procedure is carried out as follows: for a feature region v, first, its associated bounding sphere is partitioned into N_{sec} sectors with equal angle intervals along the latitudinal direction, as shown in Fig. 2(a) and (d). Second, each sector is further partitioned into N_{pol} polarities with equal angle intervals along the longitudinal direction, as shown in Fig. 2(b) and (e). Third, the bounding sphere is partitioned into N_{she} shells, a set of concentric spheres, with equal radius intervals, as shown in Fig. 2 (c) and (f). With the combination of the above three partition models, the bounding sphere of a feature region is partitioned into a set of $N_{sec} \times N_{pol} \times N_{she}$ bins, thereby realizing the partition of the feature region. The proposed polar sphere partition strategy can well model the spatial geometrical distributions of a feature region, and it is also scale-and-rotation-invariant. After feature region partition, we linearly arrange the bins into a binary vector with a length of $N_{sec} \times N_{pol} \times N_{she}$. In the binary vector, an entry with a value of 1 corresponds to a bin containing points; whereas an entry with a value of 0 corresponds to an empty bin. Then, the binary-vector-encoded feature regions form the training data to construct a DBM model.

As shown in Fig. 3(a), we train a three-layer DBM model using the binary-vectorized feature regions. In Fig. 3(a), the lines connecting adjacent layers means a bidirectional connection, where signals go upward and feedback comes downward. This model consists of a visible layer and three hidden layers. Let $\boldsymbol{p}$ denote a vector of binary visible units that represent a binary-vectorized feature region. Denote $\boldsymbol{h}^1$, $\boldsymbol{h}^2$, and $\boldsymbol{h}^3$ as the binary hidden units in each hidden layer. Then, the energy of the joint configuration $\{\boldsymbol{p}, \boldsymbol{h}^1, \boldsymbol{h}^2, \boldsymbol{h}^3\}$ of the DBM model is defined as follows (Salakhutdinov et al., 2013):

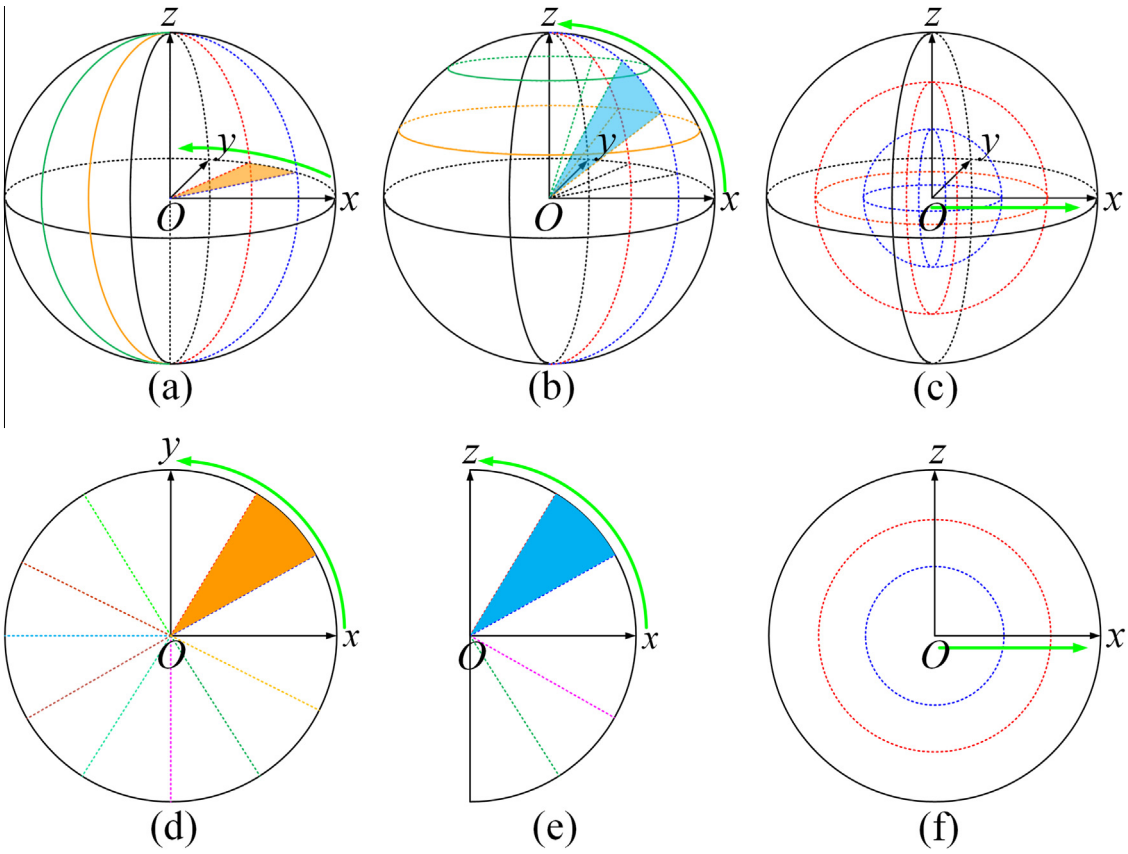

Fig. 2. Illustrations of the polar sphere partition strategy: (a), (d) sector partition model, (b), (e) polarity partition model, and (c), (f) shell partition model. The first row shows the 3-D views and the second row shows the 2-D views of the partition models.

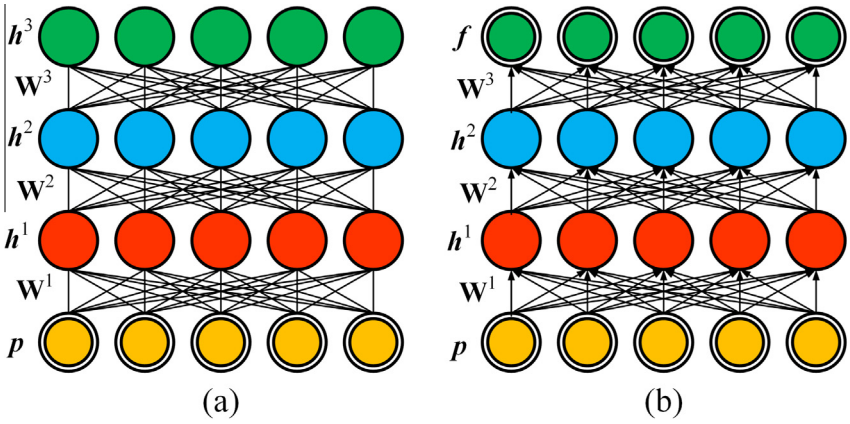

Fig. 3. Illustrations of (a) a three-layer DBM model, and (b) the DBM-based feature encoder.

$$E(\boldsymbol{p},\boldsymbol{h}^1,\boldsymbol{h}^2,\boldsymbol{h}^3;\theta) = -\boldsymbol{p}^{\mathrm{T}}\mathbf{W}^1\boldsymbol{h}^1 - (\boldsymbol{h}^1)^{\mathrm{T}}\mathbf{W}^2\boldsymbol{h}^2 - (\boldsymbol{h}^2)^{\mathrm{T}}\mathbf{W}^3\boldsymbol{h}^3 - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{b} - (\boldsymbol{h}^1)^{\mathrm{T}}\boldsymbol{a}^1 - (\boldsymbol{h}^2)^{\mathrm{T}}\boldsymbol{a}^2 - (\boldsymbol{h}^3)^{\mathrm{T}}\boldsymbol{a}^3 \quad (5)$$

where $\theta = \{\mathbf{W}^1, \mathbf{W}^2, \mathbf{W}^3, \boldsymbol{b}, \boldsymbol{a}^1, \boldsymbol{a}^2, \boldsymbol{a}^3\}$ are the model parameters. $\mathbf{W}^1$ represents the visible-to-hidden symmetric interaction term; $\mathbf{W}^2$ and $\mathbf{W}^3$ represent the hidden-to-hidden symmetric interaction terms; $\boldsymbol{b}$ represents the biases in the visible layer; $\boldsymbol{a}^1$, $\boldsymbol{a}^2$, and $\boldsymbol{a}^3$ represent the biases in the hidden layers. The marginal distribution over the visible vector $\boldsymbol{p}$ takes the following form:

$$P(\boldsymbol{p};\theta) = \frac{\sum_{\boldsymbol{h}^1,\boldsymbol{h}^2,\boldsymbol{h}^3}\exp\left(-E(\boldsymbol{p},\boldsymbol{h}^1,\boldsymbol{h}^2,\boldsymbol{h}^3;\theta)\right)}{\sum_{\boldsymbol{p}',\boldsymbol{h}^1,\boldsymbol{h}^2,\boldsymbol{h}^3}\exp\left(-E(\boldsymbol{p}',\boldsymbol{h}^1,\boldsymbol{h}^2,\boldsymbol{h}^3;\theta)\right)} \quad (6)$$

The conditional distributions over the visible and three sets of hidden units are expressed as:

$$p(h_j^1 = 1|\boldsymbol{p},\boldsymbol{h}^2) = g\left(\sum_i w_{ij}^1 p_i + \sum_m w_{jm}^2 h_m^2 + a_j^1\right) \quad (7)$$

$$p(h_m^2 = 1|\boldsymbol{h}^1,\boldsymbol{h}^3) = g\left(\sum_j w_{jm}^2 h_j^1 + \sum_k w_{mk}^3 h_k^3 + a_m^2\right) \quad (8)$$

$$p(h_k^3 = 1|\boldsymbol{h}^2) = g\left(\sum_m w_{mk}^3 h_m^2 + a_k^3\right) \quad (9)$$

$$p(p_i = 1|\boldsymbol{h}^1) = g\left(\sum_j w_{ij}^1 h_j^1 + b_i\right) \quad (10)$$

where $g(x) = 1/(1+\mathrm{e}^{-x})$ is a logistic function.

Exact maximum likelihood learning in this DBM model is intractable. To effectively learn this model, a greedy layer-wise pre-training is first performed to initialize the model parameters θ (Salakhutdinov and Hinton, 2012). Then, an iterative joint training algorithm integrated with variational and stochastic approximation approaches is applied to fine-tune the model parameters (Salakhutdinov et al., 2013).

Once the DBM model is trained, the stochastic activities of binary features in each hidden layer are replaced by deterministic, real-valued probabilities to construct a multi-layer DBM-based feature encoder (see Fig. 3(b)). Given an input binary vector $\boldsymbol{p}_v$ associated with a feature region v, the output of the feature encoder generates a high-order feature representation for feature region v:

$$\boldsymbol{f}_v^{\mathrm{T}} = g\left(g\left(g\left(\boldsymbol{p}_v^{\mathrm{T}}\mathbf{W}^1 + (\boldsymbol{a}^1)^{\mathrm{T}}\right)\mathbf{W}^2 + (\boldsymbol{a}^2)^{\mathrm{T}}\right)\mathbf{W}^3 + (\boldsymbol{a}^3)^{\mathrm{T}}\right) \quad (11)$$

3.3. Visual phrase dictionary generation

Before generating the visual phrase dictionary, first, we vector-quantize the high-order feature characterized feature regions that are generated from the training data to generate a visual word vocabulary. In our implementation, the vector quantization is carried out using k-means clustering based on the χ^2 distance as follows:

$$WDis(P_{v_i}, P_{v_j}) = \sum_k \frac{\left[P_{v_i}(k) - P_{v_j}(k)\right]^2}{P_{v_i}(k) + P_{v_j}(k)} \quad (12)$$

where P_{v_i} and P_{v_j} are the feature representations of feature regions v_i and v_j, respectively.

After vector quantization, each cluster center is taken as a distinct visual word. Finally, such visual words form a visual word vocabulary (see Fig. 4(a)). Each visual word in the vocabulary encodes a unique feature for the feature regions. Based on the visual word vocabulary, each feature region is assigned to a unique visual word by ascertaining the cluster center with the shortest distance to the feature region (i.e., the most similar visual word).

Generally, spatial contextual information exhibits richer, more salient, and distinctive representations than only using individual local feature regions. Thus, in this paper, to take advantage of spatial contextual information around a feature region, we construct a spatial word pattern for each feature region. For a feature region v, its spatial word pattern is constructed by including the visual words of feature region v and its k-nearest neighboring feature regions. In fact, if too many feature regions are combined, the repeatability of the combination might decrease. Thus, in this paper, we fix the maximum number of visual words (N_p) in a spatial word pattern as four. The spatial word pattern is represented by a bag-of-visual-words (BoVWs) using a standard "term frequency" weighting (Sivic and Zisserman, 2009; Baeza-Yates and Ribeiro-Neto, 1999).

Suppose we have a visual word vocabulary of V words. Then, each spatial word pattern is represented by a V-dimensional vector of word frequencies:

$$q = (\omega_1, \omega_2, \ldots, \omega_i, \ldots, \omega_V)^{\mathrm{T}} \quad (13)$$

where ω_i denotes the word frequency of visual word i in the vocabulary in spatial word pattern q, and it takes the following form:

$$\omega_i = \frac{n_i}{\sum_{j=1}^V n_j} \quad (14)$$

where n_i is the number of occurrences of visual word i in spatial word pattern q.

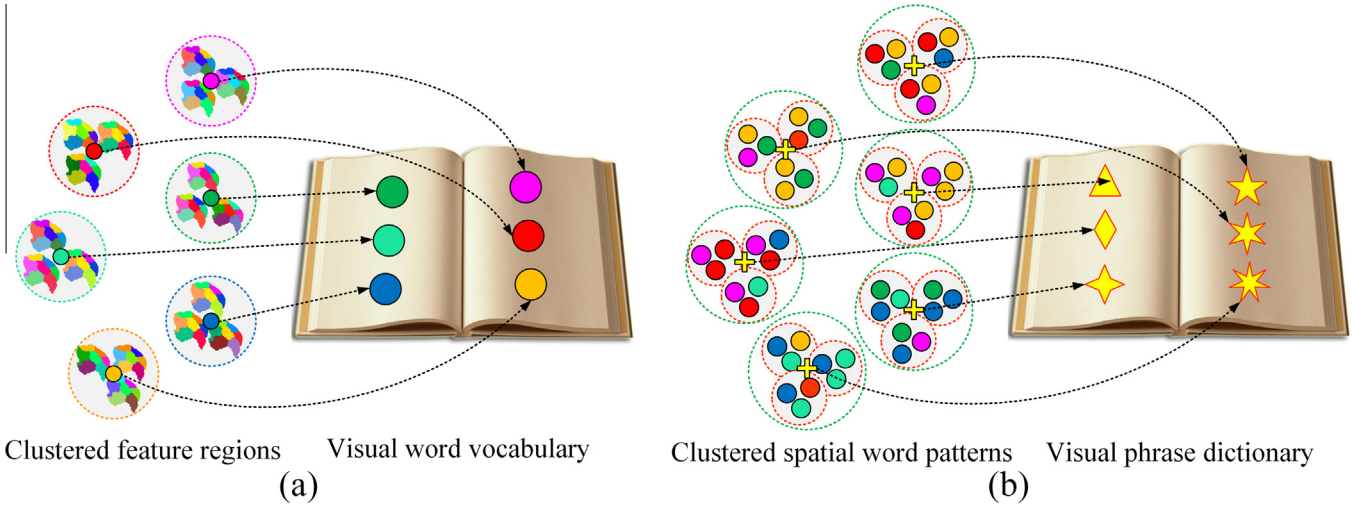

Fig. 4. Illustrations of (a) visual word vocabulary generation and (b) visual phrase dictionary generation.

In this way, each feature region is associated with a spatial word pattern containing a BoVWs for modeling its spatial contextual information. Afterward, we vector-quantize the spatial word patterns of the feature regions to generate a visual phrase dictionary. In our implementation, considering the high sparsity of the BoVWs of a spatial word pattern, the vector quantization is accomplished using k-means clustering based on the cosine distance as follows:

$$PDis(q_i, q_j) = \frac{q_i^{\mathrm{T}} q_j}{\|q_i\|_2 \cdot \|q_j\|_2} \tag{15}$$

where q_i and q_j are the BoVWs of spatial word patterns i and j, respectively. Because cosine distance has demonstrated to be the most effective measure for analyzing and clustering high-dimensional, highly sparse, and non-negative vectors (Bolovinou et al., 2013). As shown in Fig. 4(b), after vector quantization, we construct a visual phrase dictionary, where each cluster center forms a unique, meaningful visual phrase.

4. Traffic sign detection

In this section, we present a traffic sign detection framework by using the generated visual phrase dictionary. For a search scene, semantic objects are first segmented through a combination of Euclidean distance clustering and extended voxel-based normalized cut segmentation. Then, the query object and each of the segmented semantic objects are supervoxelized, featured, and quantized to form BoVPs representations. Next, traffic signposts are detected based on the similarity measures between the BoVPs of the query object and the segmented semantic objects. Finally, traffic signs are located and segmented through percentile-based analysis.

4.1. Semantic object segmentation

For a search scene, to reduce the computational complexity, a preprocessing is first performed to remove ground points from the scene using the voxel-based upward growing approach (Yu et al., 2015b). Fig. 5(b) shows the obtained off-ground points after ground point removal.

Currently, many methods have been proposed for segmenting point clouds into semantic objects, such as min-cut based segmentation method (Golovinskiy and Funkhouser, 2009a), two-step segmentation method (Zhou et al., 2014), shape-based segmentation method (Yang and Dong, 2013), etc. In this paper, to group the discrete, unorganized off-ground points into semantic objects, first, we apply a Euclidean distance clustering method (Yu et al., 2015a, 2015b) to the off-ground points to partition them into separated clusters. Specifically, with a given clustering distance d_c, Euclidean distance clustering groups discrete points based on their Euclidean distances to their neighbors. Then, in order to further separate overlapped objects which cannot be effectively partitioned via Euclidean distance clustering, we propose an extended voxel-based normalized cut segmentation method, which is an improved version of the voxel-based normalized cut segmentation method (Yu et al., 2015a, 2015b). To improve segmentation performance on clusters containing seriously overlapped objects, except for geometric features of voxels, intensity features of voxels are also considered to compute the weights on the edges of the complete weighted graph:

$$w_{ij} = \begin{cases} \exp\left(-\frac{\left\|p_i^{\mathrm{H}} - p_j^{\mathrm{H}}\right\|_2^2}{\sigma_{\mathrm{H}}^2}\right) \cdot \exp\left(-\frac{\left|p_i^{\mathrm{V}} - p_j^{\mathrm{V}}\right|^2}{\sigma_{\mathrm{V}}^2}\right) \cdot \exp\left(-\frac{\left|I_i^{\mathrm{n}} - I_j^{\mathrm{n}}\right|^2}{\sigma_{\mathrm{I}}^2}\right), \\ \quad \text{if } \left\|p_i^{\mathrm{H}} - p_j^{\mathrm{H}}\right\|_2 \leqslant d_{\mathrm{H}} \\ 0, \qquad \text{otherwise} \end{cases} \tag{16}$$

where w_{ij} is the weight on the edge connecting voxels i and j; $p_i = (x_i, y_i, z_i)$ and $p_j = (x_j, y_j, z_j)$ are the centroids of voxels i and j, respectively. $p_i^{\mathrm{H}} = (x_i, y_i)$ and $p_j^{\mathrm{H}} = (x_j, y_j)$ are the coordinates of the centroids on the XY plane; $p_i^{\mathrm{V}} = z_i$ and $p_j^{\mathrm{V}} = z_j$ are the z coordinates of the centroids; I_i^{n} and I_j^{n} are the interpolated normalized intensities of the points in voxels i and j, respectively. The interpolated normalized intensity of a voxel can be computed using Eq. (1). σ_{H}^2, σ_{V}^2, and σ_{I}^2 are the variances of the horizontal, vertical, and intensity distributions, respectively. d_{H} is a distance threshold restraining the maximum valid horizontal distance between two voxels. Fig. 5(c) shows the semantic object segmentation results by using the proposed Euclidean distance clustering and extended voxel-based normalized cut segmentation.

4.2. Bag-of-visual-phrases quantization

Before carrying out traffic sign detection from the segmented off-ground semantic objects, we perform a vector quantization on a 3-D point cloud object to create a BoVPs representation based

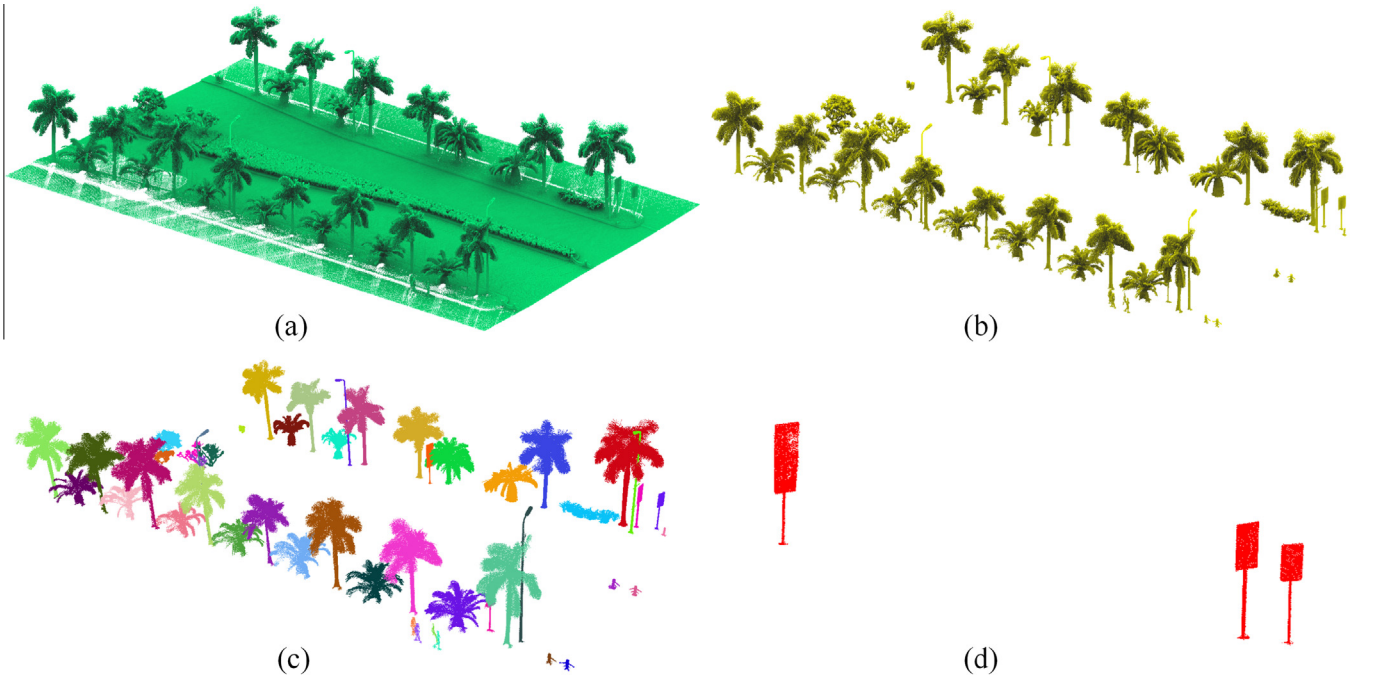

Fig. 5. Illustrations of (a) raw point cloud of a search scene, (b) off-ground points obtained after ground point removal, (c) segmented semantic objects after Euclidean distance clustering and extended voxel-based normalized cut segmentation, and (d) detected traffic signposts.

on the generated visual phrase dictionary in Section 3. Similar to the operations at the training stage, a 3-D point cloud object is first supervoxelized to generate feature regions. Then, each feature region is characterized using the DBM-based feature encoder and assigned to a unique visual word based on the generated visual word vocabulary. Next, a spatial word pattern represented by a BoVWs is constructed for each feature region. Finally, based on the visual phrase dictionary, a representative visual phrase is assigned to each spatial word pattern by ascertaining the nearest cluster center (i.e., the most similar visual phrase) to the spatial word pattern under the cosine distance metric. After vector quantization, a 3-D point cloud object is composed of a set of visual phrases, each of which encodes a distinctive feature on the point cloud object. Then, we organize such a set of visual phrases into a BoVPs representation for depicting this point cloud object. In this paper, we adopt the standard "term frequency-inverse document frequency" weighting (Sivic and Zisserman, 2009; Baeza-Yates and Ribeiro-Neto, 1999) to construct the BoVPs.

Here, we denote a 3-D point cloud object as a document. Given a visual phrase dictionary of K phrases, each document d is represented by a K-dimensional vector of weighted phrase frequencies as follows:

$$\gamma_d = (s_1, s_2, \ldots, s_i, \ldots, s_K)^{\mathrm{T}} \tag{17}$$

where s_i denotes the term frequency-inverse document frequency of phrase i in the dictionary in document d, and it takes the following form:

$$s_i = \frac{m_i^d}{\sum_{j=1}^{K} m_j^d} \log \frac{M}{M_i} \tag{18}$$

where m_i^d is the number of occurrences of phrase i in document d; M is the total number of documents in the database; M_i is the number of documents containing phrase i. This weighting is a product of two terms: the *phrase frequency* and the *inverse document frequency*. Intuitively, the phrase frequency well weights the phrases occurring more often in a particular document; whereas the inverse document frequency downweights the phrases appearing often in the database, thereby improving the distinctiveness of different documents.

In this way, a 3-D point cloud object is represented by a K-dimensional BoVPs for depicting its unique, distinctive features. This representation is used for traffic sign detection in the following section.

4.3. Traffic sign detection

To detect a specific category of traffic signs (e.g., rectangular, circular, and triangular traffic signs), first, a clean and completely scanned query object (a traffic signpost) is selected from the collected point cloud data. Then, the query object and each of the semantic objects in the search scene are supervoxelized, characterized, and quantized to form BoVPs representations. Next, based on the BoVPs, the normalized histogram intersection distance metric is used to measure the similarity between the query object and a semantic object (Jiang et al., 2015). For a query object Q and a semantic object P, the similarity between them is defined as follows:

$$Sim(Q,P) = \frac{\sum_{i=1}^{K} \min(\gamma_Q^i, \gamma_P^i)}{\sum_{i=1}^{K} \max(\gamma_Q^i, \gamma_P^i)} \tag{19}$$

where γ_Q and γ_P are the BoVPs of objects Q and P, respectively. Consequently, we compute a series of similarity measures between the query object and all the semantic objects in the search scene. Finally, the similarity measures from all semantic objects are thresholded to obtain traffic signposts. Fig. 5(d) shows the detected traffic signposts from the segmented semantic objects in Fig. 5(c).

To accurately locate and segment traffic signs from the detected traffic signposts, in this paper, we propose a percentile-based traffic sign localization method. As shown in Fig. 6(a), first, a traffic signpost is horizontally partitioned into a series of percentiles (i.e., cross-sections) with an equal height interval h_p. Then, as shown in Fig. 6(b), the points in each percentile are projected onto

the *XY* plane and a horizontal circle is fitted based on these projected points. The radius of each fitted circle is used to model the horizontal extension of each percentile. After horizontal circle fitting for each percentile, we obtain a set of radii, which can effectively model the horizontal extensions of a traffic signpost (see Fig. 6(c)). Next, to depict the horizontal extension changes, we calculate a radius difference between each pair of adjacent percentiles in a bottom-up way (see Fig. 6(d)). For the percentiles on the pole, their radius differences change very slightly. However, at the joint of the pole and the board, the radius difference changes dramatically. Such radius difference changes provide useful information for the localization of the traffic sign. Thus, based on the radius difference information across adjacent percentiles, the traffic sign is correctly located and segmented from a traffic signpost by ascertaining the first percentile with a radius difference exceeding a predefined threshold (e.g., 0.1 m) in a bottom-up way (see Fig. 6(f)).

5. Traffic sign recognition

5.1. On-image traffic sign detection

Due to the lack of informative textures of MLS point clouds, the task of traffic sign recognition cannot be accomplished only based on the point cloud data. Fortunately, along the acquisition of 3-D point cloud data, MLS systems simultaneously capture image data using the on-board digital cameras. Therefore, in this paper, the images captured by the on-board cameras of the MLS system are used for traffic sign recognition. Based on the detected traffic sign point clouds in Section 4, on-image traffic sign detection is first performed by projecting the 3-D points of each detected traffic sign onto a 2-D image. The point-cloud-to-image registration process is composed of the following two steps: (1) map the 3-D points in the WGS84 coordinate system onto the camera coordinate system, and (2) project the points in the camera coordinate system onto the image plane defined by the camera system.

Denote CMCS as the camera coordinate system, BODY as the vehicle coordinate system, ECEF as the global coordinate system (WGS84 used in this study), and NED as the north-east-down coordinate system. First, mapping a 3-D point in the ECEF onto the CMCS takes the following three transformations: ECEF-to-NED, NED-to-BODY, and BODY-to-CMCS. In the VMX-450 MLS system, a highly integrated and accurately calibrated system, used in this study, three transformation matrices ($\mathbf{C}_{\text{ECEF2NED}}$, $\mathbf{C}_{\text{NED2BODY}}$, and $\mathbf{C}_{\text{BODY2CMCS}}$) are provided for the above transformations. Thus, for a 3-D point P_{ECEF} in the ECEF, its corresponding mapped point P_{CMCS} in the CMCS is computed as follows:

$$P_{\text{CMCS}} = \mathbf{C}_{\text{BODY2CMCS}} \cdot \mathbf{C}_{\text{NED2BODY}} \cdot \mathbf{C}_{\text{ECEF2NED}} \cdot P_{\text{ECEF}}. \tag{20}$$

Then, point P_{CMCS} is projected onto a 2-D image plane by obtaining the corresponding image pixel coordinates according to the 3-D points in the CMCS. Fig. 7(b) shows the projection results of a traffic sign onto a 2-D image. Finally, on the 2-D image, a bounding box is determined for the projected traffic sign points (see Fig. 7(b)). The image pixels within the bounding box form a traffic sign region, which is used for traffic sign recognition in the following section (see Fig. 7(c)).

5.2. Traffic sign recognition

To effectively classify traffic signs into specific categories, in this paper, we propose a supervised Gaussian-Bernoulli DBM model to construct a hierarchical classifier. As shown in Fig. 8(a), we jointly train a three-layer Gaussian-Bernoulli DBM model from the normalized, labeled training data. This model consists of a visual layer, a label layer, and three hidden layers. Denote $\boldsymbol{t}$ as a vector of real-valued visible units that represents a normalized, linearly vectorized traffic sign training sample. Denote $\boldsymbol{L}$ as a binary label vector with a "1-of-K" encoding pattern (Salakhutdinov et al., 2013). That is, for a vector of K elements, only one element is encoded with a value of one and the other elements are encoded with zeros. Denote $\boldsymbol{H}^1$, $\boldsymbol{H}^2$, and $\boldsymbol{H}^3$ as the binary hidden units in each hidden layer. Then, the energy of the joint configuration $\{\boldsymbol{t}, \boldsymbol{L}, \boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{H}^3\}$ is defined as:

$$\begin{aligned} E(\boldsymbol{t}, \boldsymbol{L}, \boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{H}^3; \psi) = & \frac{1}{2}\sum_i \frac{t_i^2}{\sigma_i^2} - \sum_{i,j} \frac{t_i}{\sigma_i} w_{ij}^1 H_j^1 - \sum_{jm} H_j^1 w_{jm}^2 H_m^2 \\ & - \sum_{mk} H_m^2 w_{mk}^3 H_k^3 - \sum_{nk} l_n w_{nk}^{\text{L}} H_k^3 \\ & - \sum_i \frac{t_i}{\sigma_i} b_i - \sum_j H_j^1 a_j^1 - \sum_m H_m^2 a_m^2 \\ & - \sum_k H_k^3 a_k^3 \end{aligned} \tag{21}$$

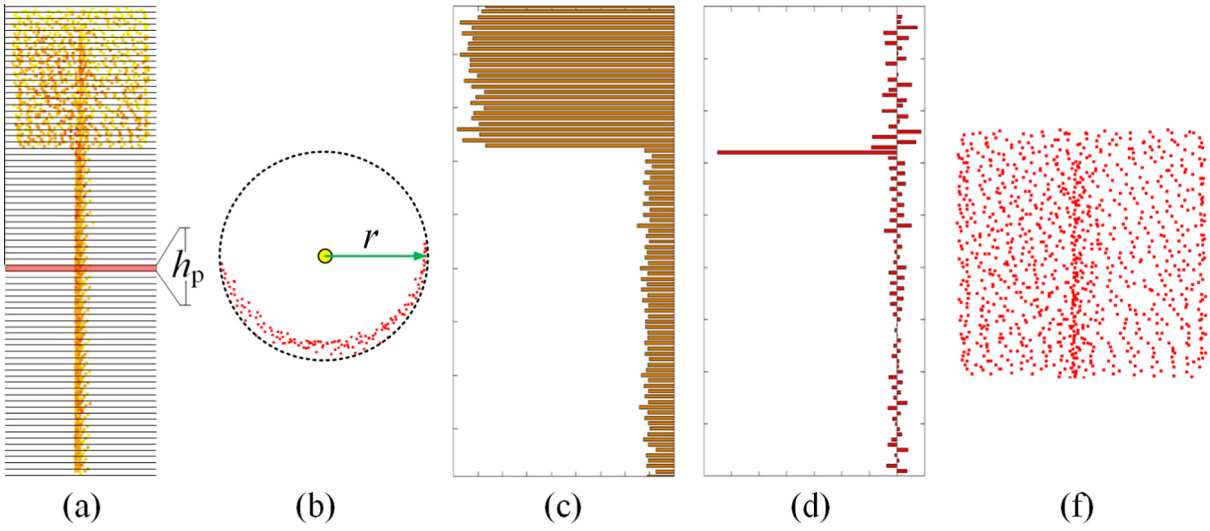

Fig. 6. Illustration of the percentile-based traffic sign localization method. (a) Percentile partition, (b) horizontal circle fitting of a percentile, (c) percentile radii, (d) radius differences between adjacent percentiles, and (f) segmented traffic sign.

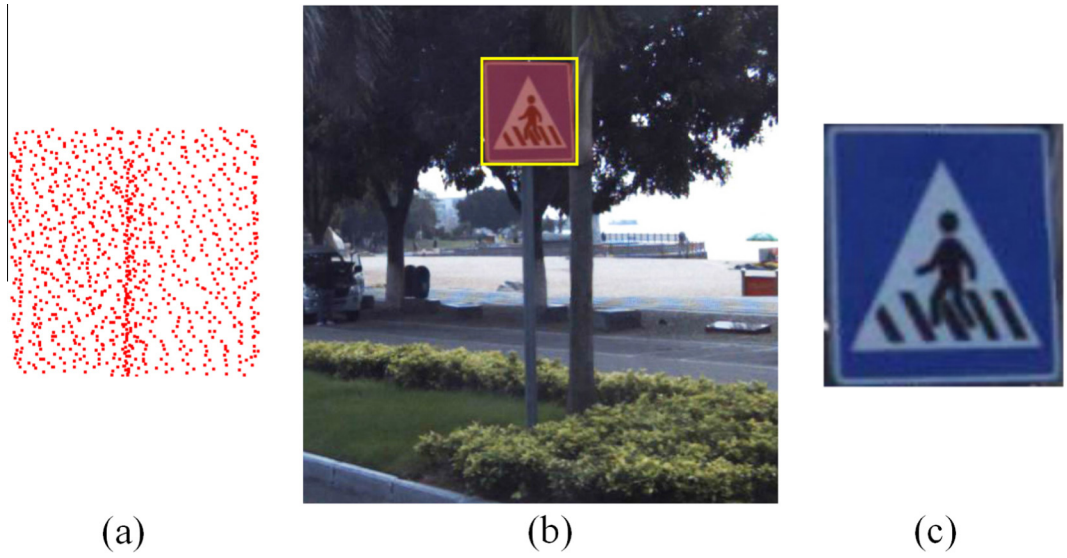

Fig. 7. Illustration of on-image traffic sign detection. (a) A detected traffic sign from the MLS point cloud, (b) projected traffic sign points onto the image and the detected traffic sign region, and (c) the segmented traffic sign.

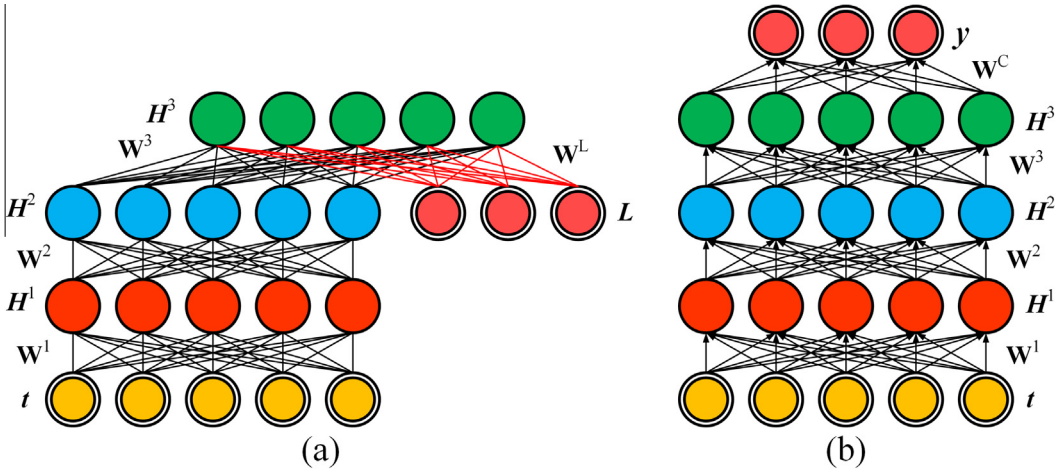

Fig. 8. Illustrations of (a) a supervised Gaussian-Bernoulli DBM model, and (b) a hierarchical classifier for traffic sign recognition.

where $\psi = \{\mathbf{W}^1, \mathbf{W}^2, \mathbf{W}^3, \mathbf{W}^L, \boldsymbol{\sigma}^2, \boldsymbol{b}, \boldsymbol{a}^1, \boldsymbol{a}^2, \boldsymbol{a}^3\}$ are the model parameters. $\mathbf{W}^1$ represents the visible-to-hidden symmetric interaction term; $\mathbf{W}^L$ represents the label-to-hidden symmetric interaction term; $\mathbf{W}^2$ and $\mathbf{W}^3$ represent the hidden-to-hidden symmetric interaction terms; $\boldsymbol{\sigma}^2$ represents the variances of the visible units; $\boldsymbol{b}$ represents the biases in the visible layer; $\boldsymbol{a}^1$, $\boldsymbol{a}^2$, and $\boldsymbol{a}^3$ represent the biases in the hidden layers. The marginal distribution over the visible vector $\boldsymbol{t}$ with a label vector $\boldsymbol{L}$ takes the following form:

$$P(\boldsymbol{t}, \boldsymbol{L}; \psi) = \frac{\sum_{\boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{H}^3} \exp\left[-E(\boldsymbol{t}, \boldsymbol{L}, \boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{H}^3; \psi)\right]}{\int_{\boldsymbol{t}'} \sum_{\boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{H}^3, \boldsymbol{L}'} \exp\left[-E(\boldsymbol{t}', \boldsymbol{L}', \boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{H}^3; \psi)\right] d\boldsymbol{t}'} \tag{22}$$

The conditional distributions over the visible, label, and three sets of hidden units are expressed as follows:

$$p(H_j^1 = 1 | \boldsymbol{t}, \boldsymbol{H}^2) = g\left(\sum_i w_{ij}^1 \frac{t_i}{\sigma_i} + \sum_m w_{jm}^2 H_m^2 + a_j^1\right) \tag{23}$$

$$p(H_m^2 = 1 | \boldsymbol{H}^1, \boldsymbol{H}^3) = g\left(\sum_j w_{jm}^2 H_j^1 + \sum_k w_{mk}^3 H_k^3 + a_m^2\right) \tag{24}$$

$$p(H_k^3 = 1 | \boldsymbol{H}^2, \boldsymbol{L}) = g\left(\sum_m w_{mk}^3 H_m^2 + \sum_n w_{nk}^L l_n + a_k^3\right) \tag{25}$$

$$p(t_i = x | \boldsymbol{H}^1) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x - \sigma_i \sum_j w_{ij}^1 H_j^1 - b_i)}{2\sigma_i^2}\right) \tag{26}$$

$$p(l_n | \boldsymbol{H}^3) = \frac{\exp\left(\sum_k w_{nk}^L H_k^3\right)}{\sum_s \exp\left(\sum_k w_{sk}^L H_k^3\right)} \tag{27}$$

To effectively train the model parameters of the supervised Gaussian-Bernoulli DBM model, first, a greedy layer-wise pre-training (Salakhutdinov and Hinton, 2012) is applied to initialize the model parameters ψ. Then, an iterative joint training algorithm integrated with variational and stochastic approximation approaches is applied to fine-tune the model parameters (Salakhutdinov et al., 2013).

After the supervised Gaussian-Bernoulli DBM model is trained, we place a logistic regression (LR) layer containing a set of softmax units on the top of the highest hidden layer to construct a

hierarchical classifier (see Fig. 8(b)) (Salakhutdinov et al., 2013). The number of softmax units indicates the number of classes. In the hierarchical classifier, the stochastic activities of binary features in each hidden layer are replaced by deterministic, real-valued probability estimations. Then, the most representative, high-order feature $\boldsymbol{H}^3$ is used as the input to the LR layer. Finally, standard back-propagation of error derivatives is performed to fine-tune the hierarchical classifier (Salakhutdinov and Hinton, 2012). Given a visible vector $\boldsymbol{t}$, the output $\boldsymbol{y}$ of the hierarchical classifier is computed as follows:

$$(\boldsymbol{H}^3)^{\mathrm{T}} = g\left(g\left(g\left(\frac{\boldsymbol{t}^{\mathrm{T}}}{\boldsymbol{\sigma}^{\mathrm{T}}}\mathbf{W}^1 + (\boldsymbol{a}^1)^{\mathrm{T}}\right)\mathbf{W}^2 + (\boldsymbol{a}^2)^{\mathrm{T}}\right)\mathbf{W}^3 + (\boldsymbol{a}^3)^{\mathrm{T}}\right) \tag{28}$$

$$y_n = \frac{\exp\left(\sum_k w_{kn}^{\mathrm{C}} H_k^3\right)}{\sum_s \exp\left(\sum_k w_{ks}^{\mathrm{C}} H_k^3\right)} \tag{29}$$

Then, the class label L^* of vector $\boldsymbol{t}$ associated with a traffic sign is determined as follows:

$$L^* = \arg\max_n y_n. \tag{30}$$

In order to classify the detected traffic signs using the proposed hierarchical classifier, first, a traffic sign image is resized into a square shape with a size of $N \times N$ pixels. Then, the pixel values of the resized image are normalized into the range of [0, 1]. Next, the normalized image is linearly arranged into a real-valued vector. Such a vector forms the input data to the hierarchical classifier. Finally, the class information of the traffic sign is inferred using Eqs. (28)–(30).

6. Results and discussion

6.1. RIEGL VMX-450 system and MLS data sets

In this study, a RIEGL VMX-450 system was used to collect both point clouds and images in Xiamen City, China. This system is composed of (1) two RIEGL VQ-450 laser scanners, (2) four 2452×2056 pixels CS6 color cameras, and (3) an integrated IMU/GNSS/DMI position and orientation system. The two laser scanners achieve a maximum effective measurement rate of 1.1 million measurements per second, a line scan speed of up to 400 lines per second, and a maximum valid range of 800 m.

We collected four data sets on Ring Road South (RRS), Xiahe Road (XHR), Zhongshan Road (ZSR), and Hubin Road West (HRW), respectively. These roads are typical urban road scenes with a considerable number of traffic signs for regulating traffic activities. A detailed description of these four data sets are listed in Table 1. At the visual phrase dictionary generation stage, we manually, at random, selected a total number of 80 point cloud segments, each of which has a road section of approximately 50 m, from the collected point cloud data. These point cloud segments do not overlap with the four selected data sets. After ground point removal, all the objects within each point cloud segment were used to train the DBM-based feature encoder and build the visual phrase dictionary. To train the supervised Gaussian-Bernoulli DBM model for traffic sign recognition, we collected a set of standard traffic sign pictograms with correctly assigned class labels from the Ministry of Transport of the People's Republic of China (see Fig. 9). At the Gaussian-Bernoulli DBM model training stage, each traffic sign pictogram was resized into a square shape with a size of 80×80 pixels. To augment the traffic sign pictograms to cover different conditions in real scenes, each traffic sign pictogram was processed with various distortions, including illumination changes, rotations, and Gaussian-noise contaminations (Chigorin and Konushin, 2013). After augmentation, a labeled traffic sign data set containing 161,792 training samples was used to train the Gaussian-Bernoulli DBM model.

Table 1
Descriptions of the four selected data sets.

Data set	Road length (km)	Point number	Point density (points/m^2)	Image number
RRS	11	1,728,001,432	4082	10,896
XHR	5	2,073,208,821	4419	9324
ZSR	14	2,497,168,131	3977	12,448
HRW	9	1,366,054,611	4033	10,912

6.2. Parameter sensitivity analysis

In the proposed algorithm, the configurations of the following three parameters have a significant impact on the traffic sign detection performance based on 3D point clouds: feature region construction pattern, visual phrase dictionary size (K), and spatial word pattern size (N_{p}). In order to obtain an optimal configuration for each of these parameters, we conducted a group of experiments to test the performance of each parameter configuration on the traffic sign detection results. For feature region construction, we tested the following two construction patterns: using single supervoxels and using the combination of supervoxels and their first-order neighbors. For visual phrase dictionary generation, we tested the following six configurations: K = 90,000, 100,000, 110,000, 120,000, 130,000, and 140,000. For spatial word pattern construction, we tested the following six configurations: N_{p} = 1, 2, 3, 4, 5, and 6. The traffic sign detection results of these parameter configurations were presented and analyzed using precision-recall curves (see Fig. 10).

As shown in Fig. 10(a), by integrating first-order supervoxel neighbors to construct feature regions, the detection performance improves greatly than that of treating only single supervoxels as feature regions. This is because feature regions with first-order neighborhood information can produce more meaningful, salient, and distinctive feature encodings than that of using only local supervoxels. Thus, the proposed feature region construction pattern by integrating first-order supervoxel neighbors performs better in traffic sign detection. As shown in Fig. 10(b), the detection performance improves as the dictionary size increases. This is because the more the visual phrases in the dictionary, the higher degrees of distinctions between different categories of objects. However, when the dictionary size exceeds 120,000, the detection performance is almost stable. In addition, the increase of the dictionary size brings computational burdens at the dictionary generation stage. Thus, balancing detection performance and computational complexity, we set the dictionary size at K = 120,000. As shown in Fig. 10(c), when $N_{\mathrm{p}} \leqslant 4$, the detection performance improves with the increase of the number of visual words in a spatial word pattern. This is because, by considering spatial contextual information of feature regions, the quantized visual phrases are more likely to obtain salient, distinctive feature encodings, thereby capable of differentiating objects from different categories. However, when $N_{\mathrm{p}} \geqslant 5$, the detection performance drops dramatically. In fact, if too many local feature regions are combined, the repeatability of the combination decreases accordingly, leading to a detection performance decrease. Therefore, we set the spatial word pattern size at N_{p} = 4.

6.3. Traffic sign detection on point clouds and images

To evaluate the performance of our proposed traffic sign detection algorithm, we applied it to the aforementioned four data sets (i.e., RRS, XHR, ZSR, and HRW data sets). After parameter sensitivity analysis, the optimal parameter configurations used in the

Fig. 9. Illustration of a subset of the traffic sign pictograms used for training.

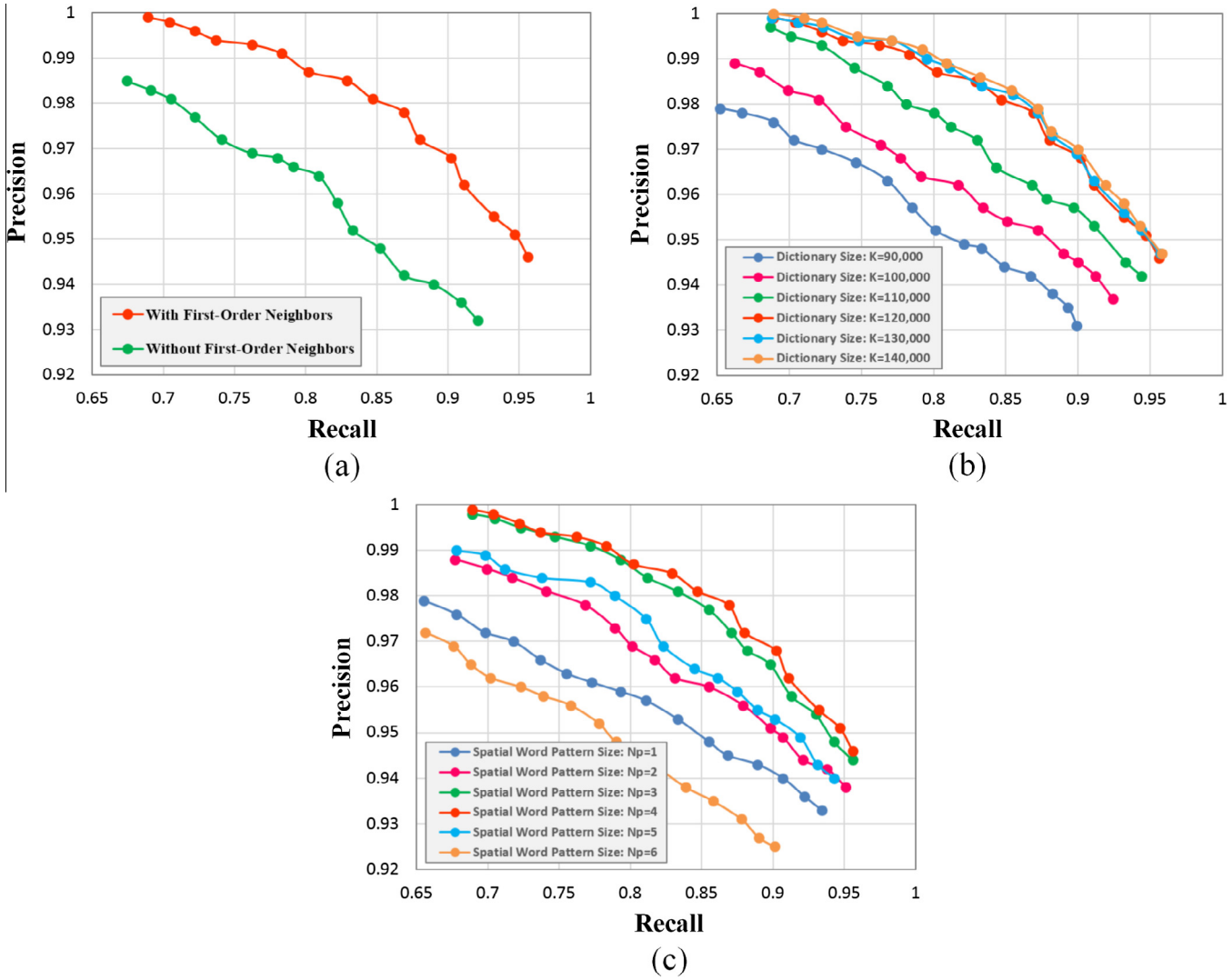

Fig. 10. Illustrations of the traffic sign detection performances obtained under different parameter configurations: (a) feature region construction with and without first-order supervoxel neighbors, (b) visual phrase dictionary size, and (c) spatial word pattern size.

proposed algorithm are detailed in Table 2. To generate high-order feature encodings of feature regions, we constructed a 6480-1000-1000-100 DBM-based feature encoder. To detect traffic signs of different shapes (e.g., rectangular, circular, and triangular traffic signs), a group of query objects were selected from the collected point clouds.

The traffic sign detection results, along with the manually labeled ground truth, on the four selected data sets are shown in Table 3, where TP, FN, and FP denote the numbers of true positives (correctly detected traffic signs), false negatives, and false positives, respectively. Fig. 11 shows an example of a part of traffic sign detection results in 3-D point clouds. Fig. 12 shows a subset of detected traffic signs on 2-D images. As reflected by the traffic sign detection results, the majority of traffic signs of different shapes and conditions were correctly detected. However, due to high geometric similarities of some road-scene objects to traffic signs, these objects were falsely detected as traffic signs by using the proposed algorithm (see Fig. 13(a)). Moreover, some traffic signs are attached to light poles, traffic lights, and utility poles. Such traffic signs failed to be detected by using the proposed algorithm (see Fig. 13(b)). In addition, some incompletely scanned traffic signs caused by occlusions were also undetected because of insufficient salient features. On the whole, the proposed algorithm obtained very promising performance in detecting traffic signs from both 3-D point clouds and 2-D images.

To further demonstrate the effectiveness of the proposed traffic sign detection algorithm, we evaluated the traffic sign detection performance in challenging image scenarios such as strong illuminations, poor illuminations, large viewpoints, partial occlusions, and cluttered backgrounds. Fig. 14 shows some examples of traffic signs detected in challenging image scenarios. For image-based traffic sign detection algorithms, it is greatly challenging to effectively detect such traffic signs in such challenging scenarios. However, in our proposed algorithm, the traffic sign detection task was totally accomplished based on 3-D point clouds rather than 2-D images. Since 3-D point clouds provide real-world 3-D geometric topologies of traffic signs and are immune to environmental illumination conditions, traffic signs can be effectively detected from 3-D point clouds without the impact of viewpoint variations and illumination variations that often occur in 2-D images. Then, after projecting the detected traffic signs in 3-D point clouds onto 2-D images, such traffic signs in challenging image scenarios can be accurately detected.

(1) Performance evaluation

To quantitatively assess the accuracy of the traffic sign detection results on the four selected data sets, we adopted the following four indices: *recall*, *precision*, *quality*, and *F-score* (Yu et al., 2015a, 2015b). *Recall* evaluates the proportion of true positives in the ground truth; *precision* measures the proportion of true positives in the detected components; *quality* and *F-score* are two overall measures. The four indices are defined as follows: $recall = TP/(TP + FN)$, $precision = TP/(TP + FP)$, $quality = TP/(TP + FN + FP)$, and $F\text{-}score = 2 \times recall \times precision/(recall + precision)$, where *TP*, *FN*, and *FP* denote the numbers of true positives, false negatives, and false positives, respectively. The quantitative evaluation results on the four selected data sets are detailed in Table 3. The proposed traffic sign detection algorithm achieved an average recall, precision, quality, and *F*-score of 0.956, 0.946, 0.907, and 0.951, respectively, on the four selected data sets. Therefore, the proposed algorithm performs efficiently in detecting traffic signs from both MLS point cloud and image data.

Table 2
Parameters and their optimal configurations.

N_{sec}	N_{pol}	N_{she}	V	N_p	K
36	18	10	120,000	4	120,000
d_c	σ_H	σ_V	σ_I	d_H	h_p
0.15 m	2 m	10 m	1.0	5 m	0.03 m

The proposed algorithm was implemented using C++ and tested on an HP Z820 8-core-16-thread workstation. The computing costs at the visual phrase dictionary generation and traffic sign detection stages were recorded for time complexity evaluation. The total computing times for training the DBM-based feature encoder and generating the visual phrase dictionary containing 120,000 visual phrases were about 5.4 h and 41 min, respectively. At the traffic sign detection stage, each data set was first partitioned into a group of data segments with a road length of about 50 m. Then, all the segments were fed into a multithread computing environment containing 16 parallel threads. Such a parallel computing strategy dramatically improves the computational efficiency and reduces the time complexity of the proposed algorithm. Table 4 details the computing time of each processing step at the traffic sign detection stage. As reflected by Table 4, BoVPs quantization of off-ground semantic objects took the majority of the total processing time. The total computing times for traffic sign detection were about 43, 45, 59, and 52 min for the RRS, XHR, ZSR, and HRW data sets, respectively. Therefore, the proposed algorithm is suitable for rapidly handling large-volume MLS point clouds toward traffic sign detection.

(2) Comparative studies of using different 3-D descriptors

In this paper, we proposed a DBM-based feature encoder to generate high-order feature representations of feature regions. To demonstrate the superior performance of the DBM-based feature encoder in exploiting salient, distinctive features of feature regions, we compared it with the following two 3-D descriptors: 3-D shape context (3DSC) (Körtgen et al., 2003) and fast point feature histograms (FPFH) (Rusu et al., 2009). In our experiments, Point Cloud Library (PCL) (Rusu and Cousins, 2011), an open source library for 3-D point cloud processing, was used to implement the 3DSC and the FPFH descriptors. After parameter sensitivity analysis, for computing the 3DSC descriptor, the optimal radius, number of shells, and number of sectors were set at 0.14 m, 7, and 72, respectively; the optimal radius for computing the FPFH descriptor was set at 0.12 m. Accordingly, two visual phrase dictionaries were built based on the features obtained by the 3DSC and the FPFH descriptors, respectively. The traffic sign detection results and quantitative evaluations based on the 3DSC and the FPFH descriptors are listed in Table 5. As shown in Table 5, the FPFH descriptor obtained better traffic sign detection results than the 3DSC descriptor. Comparatively, the DBM-based feature encoder shows better performance than those achieved using the 3DSC and the FPFH descriptors. This is because the 3DSC and the FPFH descriptors can only obtain low-order statistical features. However, the DBM-based feature encoder can obtain high-order abstract feature representations, which are actually a combination and high-level abstraction of a set of low-order features. Therefore, the DBM-based feature encoder is more powerful to generate salient, distinctive features than the 3DSC and the FPFH descriptors.

(3) Comparative studies with point cloud based traffic sign detection methods

To further demonstrate the advantageous performance of our proposed traffic sign detection algorithm, we conducted a group of experiments to compare it with the following four existing methods: Hough forest-based method (HF) (Wang et al., 2014), supervoxel neighborhood-based Hough forest method (SHF) (Wang et al., 2015), 3-D object matching-based method (OM) (Yu et al., 2015b), and intensity-based pole-like object detection method (IPLO) (Wen et al., 2015). The aforementioned four methods were implemented using the authors' public codes with default parameter configurations. A performance evaluation on

Table 3
Traffic sign detection results of different data sets obtained by different methods: the proposed algorithm, HF method (Wang et al., 2014), SHF method (Wang et al., 2015), OM method (Yu et al., 2015b), and IPLO method (Wen et al., 2015). The bold values represent the best experiment results.

Data set	Method	Ground truth	TP	FP	FN	Recall	Precision	Quality	*F*-score
RRS	Proposed	241	**231**	15	**10**	**0.959**	0.939	**0.902**	**0.949**
	HF		222	18	19	0.921	0.925	0.857	0.923
	SHF		224	18	17	0.929	0.926	0.865	0.927
	OM		229	17	12	0.950	0.931	0.888	0.940
	IPLO		227	**11**	14	0.942	**0.954**	0.901	0.948
XHR	Proposed	372	**358**	21	**14**	**0.962**	0.945	**0.911**	**0.953**
	HF		340	22	32	0.914	0.939	0.863	0.926
	SHF		343	23	29	0.922	0.937	0.868	0.929
	OM		349	20	23	0.938	0.946	0.890	0.942
	IPLO		352	**17**	20	0.946	**0.954**	0.905	0.950
ZSR	Proposed	396	**377**	**23**	**19**	**0.952**	**0.943**	**0.900**	**0.947**
	HF		366	29	30	0.924	0.927	0.861	0.925
	SHF		367	28	29	0.927	0.929	0.866	0.928
	OM		371	26	25	0.937	0.935	0.879	0.936
	IPLO		369	25	27	0.932	0.937	0.876	0.934
HRW	Proposed	307	**292**	13	**15**	**0.951**	0.957	**0.913**	**0.954**
	HF		275	15	32	0.896	0.948	0.854	0.921
	SHF		280	14	27	0.912	0.952	0.872	0.932
	OM		284	15	23	0.925	0.950	0.882	0.937
	IPLO		285	**12**	22	0.928	**0.960**	0.893	0.944

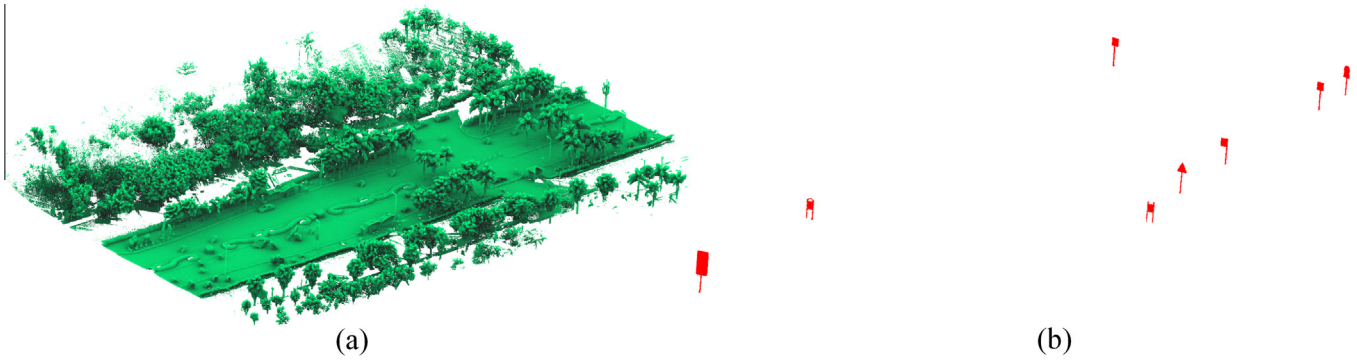

Fig. 11. Illustration of a part of traffic sign detection results in 3-D point clouds. (a) A raw point cloud, and (b) detected traffic signs.

Fig. 12. Illustration of a subset of detected traffic signs on 2-D images.

Fig. 13. Illustrations of (a) falsely detected non-traffic sign objects, and (b) undetected traffic signs.

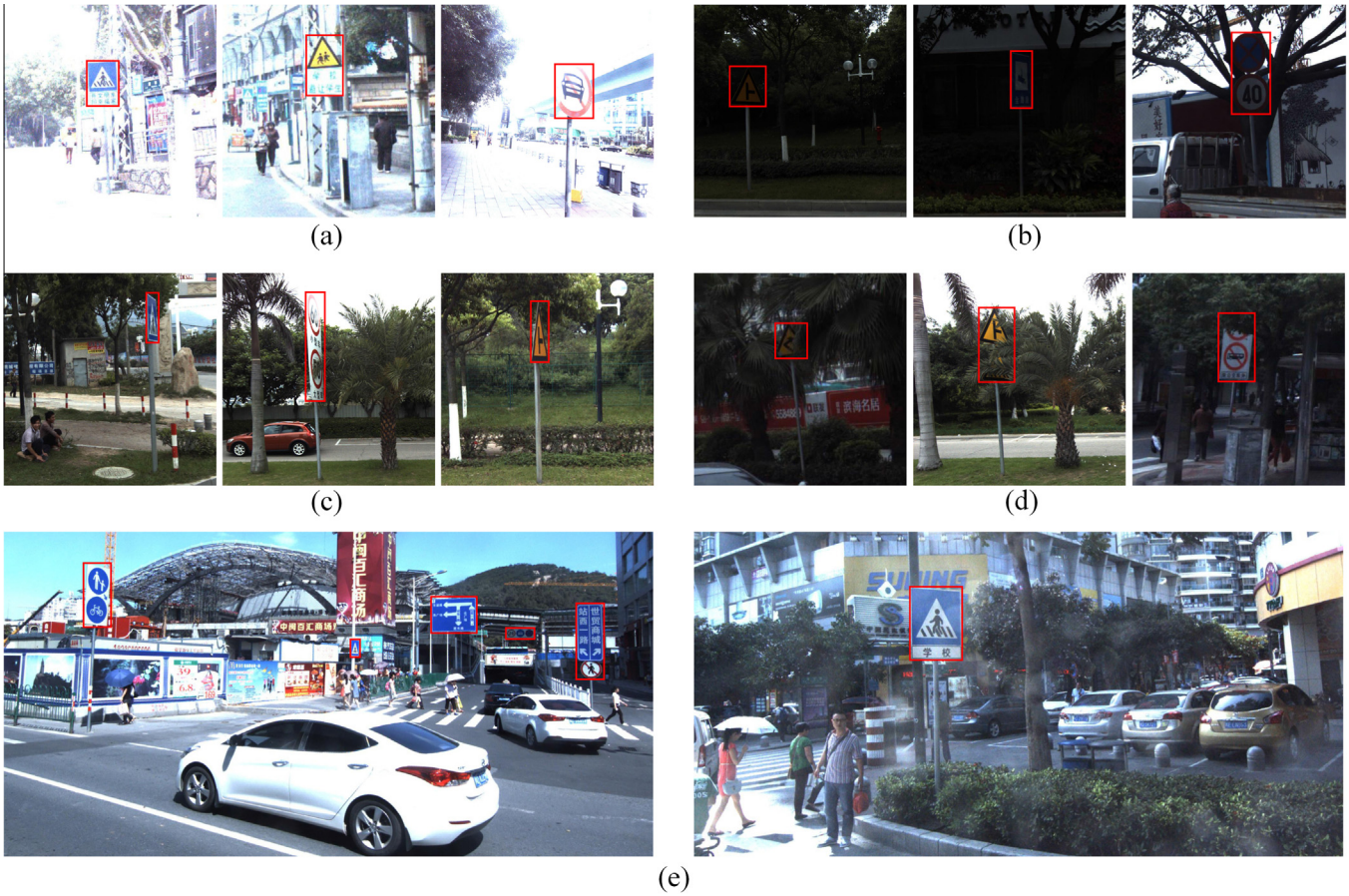

Fig. 14. Examples of traffic signs detected in challenging scenarios. (a) Strong illuminations, (b) poor illuminations, (c) large viewpoints, (d) partial occlusions, and (e) cluttered backgrounds.

Table 4
Computing time of each processing step at the traffic sign detection stage (seconds).

Data set	Ground removal	Object segmentation	BoVPs quantization	Traffic sign detection	Total
RRS	34	163	2334	25	2556
XHR	23	214	2427	45	2709
ZSR	41	228	3235	37	3541
HRW	32	236	2798	52	3118

traffic sign detection from 3-D point clouds was conducted on the four selected data sets by using the aforementioned four methods. The traffic sign detection results, as well as quantitative evaluations, obtained by using these four methods are detailed in Table 3.

Comparatively, the OM and IPLO methods obtained better performance than the HF and SHF methods. Based on only low-level geometric features of local 3-D patches, the HF and SHF methods lack of sufficient feature representations to distinguish traffic signs from other pole-like objects. Thus, a relatively greater number of

Table 5
Traffic sign detection results obtained by using different 3-D descriptors: the proposed DBM-based feature encoder, 3DSC (Körtgen et al., 2003) and FPFH (Rusu et al., 2009). The bold values represent the best experiment results.

Data set	Descriptor	Ground truth	TP	FP	FN	Recall	Precision	Quality	*F*-score
RRS	Proposed	241	**231**	**15**	**10**	**0.959**	**0.939**	**0.902**	**0.949**
	3DSC		215	18	26	0.892	0.923	0.830	0.907
	FPFH		223	17	18	0.925	0.929	0.864	0.927
XHR	Proposed	372	**358**	**21**	**14**	**0.962**	**0.945**	**0.911**	**0.953**
	3DSC		334	22	38	0.898	0.938	0.848	0.918
	FPFH		342	23	30	0.919	0.937	0.866	0.928
ZSR	Proposed	396	**377**	**23**	**19**	**0.952**	**0.943**	**0.900**	**0.947**
	3DSC		358	25	38	0.904	0.935	0.850	0.919
	FPFH		363	**23**	33	0.917	0.940	0.866	0.928
HRW	Proposed	307	**292**	**13**	**15**	**0.951**	**0.957**	**0.913**	**0.954**
	3DSC		272	17	35	0.886	0.941	0.840	0.913
	FPFH		285	15	22	0.928	0.950	0.885	0.939

false positives were generated by using the HF and SHF methods. The OM and IPLO methods partially rely on the off-ground semantic object segmentation results; thus, the performance of segmentation methods affects the traffic sign detection performance. However, in the selected data sets, some traffic signs are hidden in trees and seriously overlapped with the trees; therefore, such traffic signs failed to be detected by using the OM and IPLO methods. In addition, the IPLO method utilizes intensity properties to detect traffic signs from the detected pole-like objects. However, in the selected data sets, some traffic signs were scanned from the opposite direction of the lane, and the back sides of the traffic signs (without highly reflective materials) were scanned (see Fig. 15). Thus, such traffic signs also failed to be detected by the IPLO method. As reflected by the comparative results in Table 3, our method outperforms these methods in detecting traffic signs from 3-D point clouds.

(4) Comparative studies with image-based traffic sign detection methods

To further demonstrate the advantages of the 3-D point cloud based traffic sign detection algorithm, we compared it with the following three image-based traffic sign detection methods: template matching (TM) method (Soheilian et al., 2013), graph-based ranking and segmentation (GRS) method (Yuan et al., 2015), and color probability model (CPM) method (Yang et al., 2015). For the TM method, first, regions of interests (ROIs) are extracted based on color segmentation; then, the shape of each ROI is fitted using simple geometric forms (e.g., ellipse, quadrilateral, and triangle); finally, based on a set of reference data, a template matching process is performed to detect traffic signs. For the GRS method, first, a superpixel-based graph is designed to represent an input image; then, a ranking algorithm is applied to exploit the intrinsic manifold structure of the graph nodes; finally, a multithreshold segmentation approach is proposed to segment traffic sign regions. For the CPM method, first, an input image is transformed to traffic sign probability maps by using a color probability model; then, traffic sign proposals are extracted by finding maximally stable extremal regions from the probability maps; finally, an SVM is used to detect traffic signs from the traffic sign proposals. The aforementioned three methods were re-implemented using C++ according to their methodology descriptions. Then, with the optimal parameter configurations suggested by the corresponding methods, comparative studies were conducted on the images of the four selected data sets. The traffic sign detection results and quantitative evaluations by using the above three methods, as well as our proposed algorithm, is detailed in Table 6. In Table 6, ground truth denotes the total number of traffic signs in all test images. If a traffic sign appears in multiple images, they are regarded as individual traffic signs. Comparatively, the proposed traffic sign detection algorithm obtained better performance than the other three image-based traffic sign detection methods. The performance weakness of these image-based methods are caused by the following factors: (1) some images suffered from strong or poor illuminations (see Fig. 14(a) and (b)); some traffic signs were captured with large viewpoints (see Fig. 14(c)); some traffic signs were occluded by the nearby objects (see Fig. 14(d)); some traffic signs were captured from the back sides (see Fig. 15). Thus, such traffic signs failed to be detected. However, by using 3-D point clouds, the aforementioned problems does not exist in point clouds and have no impact on traffic sign detection. Therefore, the proposed traffic sign detection algorithm shows advantages over the three image-based traffic sign detection methods.

6.4. Traffic sign recognition on images

In our selected data sets, a total number of 1258 traffic signs were correctly detected from 1316 traffic signs (ground truth). To

Fig. 15. Illustrations of traffic signs scanned from back sides.

Table 6
Traffic sign detection results obtained by using different image-based methods: the proposed algorithm, TM method (Soheilian et al., 2013), GRS method (Yuan et al., 2015), and CPM method (Yang et al., 2015).

Data set	Method	Ground truth	TP	FP	FN	Recall	Precision	Quality	*F*-score
RRS	Proposed	1413	**1381**	**52**	**32**	**0.977**	**0.964**	**0.943**	**0.970**
	TM		1247	124	166	0.883	0.910	0.811	0.896
	GRS		1329	62	84	0.941	0.955	0.901	0.948
	CPM		1288	67	125	0.912	0.951	0.870	0.931
XHR	Proposed	2057	**1988**	**73**	**69**	**0.966**	**0.965**	**0.933**	**0.965**
	TM		1944	108	113	0.945	0.947	0.898	0.946
	GRS		1967	93	90	0.956	0.955	0.915	0.955
	CPM		1953	89	104	0.949	0.956	0.910	0.952
ZSR	Proposed	2289	**2202**	89	**87**	**0.962**	0.961	**0.926**	**0.961**
	TM		2136	145	153	0.933	0.936	0.878	0.934
	GRS		2169	**66**	120	0.948	**0.970**	0.921	0.959
	CPM		2158	87	131	0.943	0.961	0.908	0.952
HRW	Proposed	1943	**1868**	63	**75**	**0.961**	0.967	**0.931**	**0.964**
	TM		1799	95	144	0.926	0.950	0.883	0.938
	GRS		1833	**58**	110	0.943	**0.969**	0.916	0.956
	CPM		1802	74	141	0.927	0.961	0.893	0.944

classify these traffic signs into specific categories, we first projected them onto the images to obtain traffic sign regions. If a traffic sign appears in multiple images, the obtained traffic sign region with the maximum size is used for recognition. The size of the obtained traffic sign regions varies from 23×24 to 760×475 pixels. Then, the obtained traffic sign regions were resized into a square shape with a size of 80×80 pixels. Finally, an on-image traffic sign recognition was carried out using the constructed hierarchical classifier. In our implementation, we constructed a 6400-1000-1000-500-35 hierarchical classifier to classify the detected traffic signs into 35 categories according to their functionalities. The computing time for training the hierarchical classifier was about 4.9 h. To quantitatively assess the traffic sign recognition results, recognition accuracy defined as the proportion of correctly classified traffic signs in the test images was used in this study. After evaluation, a recognition accuracy of 97.54% was achieved by using our proposed hierarchical classifier. In other words, a total number of 1227 traffic signs out of 1258 traffic signs were successfully assigned to correct categories. The misclassification was basically caused by the following three factors: (1) extremely strong or poor illuminations, (2) very large viewpoints, and (3) serious occlusions. On the whole, benefiting from exploiting high-order feature representations of traffic signs, the proposed traffic sign recognition algorithm achieves very promising results and high accuracy in classifying traffic sign images.

Comparative studies. To further demonstrate the superior accuracy of our proposed traffic sign recognition algorithm, we conducted a group of tests to compare it with the following four existing methods: MSERs method (Greenhalgh and Mirmehdi, 2012), SRGE method (Lu et al., 2012), Color Global LOEMP method (Yuan et al., 2014), and CNN method (Jin et al., 2014). In this comparative study, we used two image data sets: German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2011) and Belgium Traffic Sign Classification Benchmark (BelgiumTSC) (Timofte and Van Gool, 2011). The GTSRB data set contains 43 classes of traffic signs. The training and test data sets contain 39,209 and 12,630 images, respectively. The BelgiumTSC data set contains 62 classes of traffic signs. The training and test data sets contain 4591 and 2534 images, respectively. The aforementioned four methods were re-implemented using C++ according their methodology descriptions. Then, with the optimal parameter configurations suggested by the corresponding methods, we respectively applied these four methods to the two selected data sets to conduct traffic sign recognition. The traffic sign recognition results on these two data sets by using the aforementioned four methods, as well as our proposed traffic sign recognition method, are detailed in Table 7. Comparatively, our proposed traffic sign recognition algorithm obtained similar recognition accuracies to the CNN method and relatively higher accuracies than the other three methods. By exploiting high-order feature representations of traffic signs, the CNN method and the proposed algorithm have the capability of handling various traffic sign distortions, such as illumination variations, viewpoint variations, and noise contaminations, thereby achieving better traffic sign recognition performance. In conclusion, our proposed traffic sign recognition algorithm is suitable for on-image traffic sign recognition tasks and can achieve very promising and reliable recognition results.

Table 7
Traffic sign recognition accuracies (%) on two data sets obtained by using different methods: MSERs method (Greenhalgh and Mirmehdi, 2012), SRGE method (Lu et al., 2012), LOEMP method (Yuan et al., 2014), CNN method (Jin et al., 2014), and the proposed algorithm. The bold values represent the best experiment results.

Data set	MSERs	SRGE	LOEMP	CNN	Proposed
GTSRB	87.79	98.19	97.26	**99.65**	99.34
BelgiumTSC	85.33	96.29	95.37	98.87	**98.92**

7. Conclusion

In this paper, we have proposed a novel algorithm combining 3-D point clouds and 2-D images for detecting and recognizing traffic signs based on BoVPs and hierarchical deep models. The traffic sign detection task was accomplished based on 3-D point clouds; whereas the recognition task was achieved based on 2-D images. The proposed algorithm has been evaluated on four data sets collected by a RIEGL VMX-450 system. For detecting traffic signs in 3-D point clouds, the proposed algorithm achieved an average recall, precision, quality, and *F*-score of 0.956, 0.946, 0.907, and 0.951, respectively, on the four selected data sets. For on-image traffic sign recognition, a recognition accuracy of 97.54% was achieved by using the proposed hierarchical classifier. Through computational efficiency evaluation, by adopting a multithread computing strategy, the proposed algorithm can rapidly handle large-volume MLS point clouds toward traffic sign detection. In addition, comparative studies also demonstrated that the proposed algorithm obtained promising, reliable, and high performance in both detecting traffic signs in 3-D point clouds and recognizing traffic signs on 2-D images. In conclusion, by using MLS data, we have provided an effective solution to rapid, accurate detection

and recognition of traffic signs toward transportation-related applications.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 41471379, and in part by PAPD and CICAEET. The authors would like to thank the anonymous reviewers for their valuable comments.

References

Ai, C., Tsai, Y.J., 2015. Critical assessment of an enhanced traffic sign detection method using mobile LiDAR and INS technologies. J. Transp. Eng. 141 (5), 1–12.
Baeza-Yates, R., Ribeiro-Neto, B., 1999. Modern Information Retrieval. ACM Press, New York, NY, USA.
Baró, X., Escalera, S., Vitrià, J., Pujol, O., Radeva, P., 2009. Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification. IEEE Trans. Intell. Transp. Syst. 10 (1), 113–126.
Bolovinou, A., Pratikakis, I., Perantonis, S., 2013. Bag of spatio-visual words for context inference in scene classification. Pattern Recogn. 46 (3), 1039–1053.
Brogan, M., McLoughlin, S., Deegan, C., 2013. Assessment of stereo camera calibration techniques for a portable mobile mapping system. IET Comput. Vis. 7 (3), 209–217.
Broggi, A., Buzzoni, M., Debattisti, S., Grisleri, P., Laghi, M.C., Medici, P., Versari, P., 2013. Extensive tests of autonomous driving technologies. IEEE Trans. Intell. Transp. Syst. 14 (3), 1403–1415.
Carneiro, G., Nascimento, J.C., 2013. Combining multiple dynamic models and deep learning architectures for tracking left ventricle endocardium in ultrasound data. IEEE Trans. Pattern Anal. Mach. Intell. 35 (11), 2592–2607.
Chen, B., Polatkan, G., Sapiro, G., Blei, D., Dunson, D., Carin, L., 2013. Deep learning with hierarchical convolutional factor analysis. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1887–1901.
Chen, X., Kohlmeyer, B., Stroila, M., Alwar, N., Wang, R., Bach, J., 2009. Next generation map making: geo-referenced ground-level LiDAR point clouds for automatic retro-reflective road feature extraction. In: 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November, pp. 488–491.
Chen, Y., Zhao, H., Shibasaki, R., 2007. A mobile system combining laser scanners and cameras for urban spatial objects extraction. In: IEEE International Conference on Machine Learning and Cybernetics, Pokfulam, Hong Kong, 19–22 August, vol. 3, pp. 1729–1733.
Cheng, H., Zheng, N., Zhang, X., Qin, J., Van de Wetering, H., 2007. Interactive road situation analysis for driver assistance and safety warning systems: framework and algorithms. IEEE Trans. Intell. Transp. Syst. 8 (1), 157–167.
Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J., 2001. Color image segmentation: advances and prospects. Pattern Recogn. 34 (12), 2259–2281.
Chigorin, A., Konushin, A., 2013. A system for large-scale automatic traffic sign recognition and mapping. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Antalya, Turkey, 12–13 November, vol. II-3/W3, pp. 13–17.
Choi, J., Lee, J., Kim, D., Soprani, G., Cerri, P., Broggi, A., Yi, K., 2012. Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment. IEEE Trans. Intell. Transp. Syst. 13 (2), 974–982.
Cireşan, D., Meier, U., Masci, J., Schmidhuber, J., 2012. Multi-column deep neural network for traffic sign classification. Neural Netw. 32, 333–338.
de la Escalera, A., Moreno, L.E., Salichs, M.A., Armingol, J.M., 1997. Road traffic sign detection and classification. IEEE Trans. Ind. Electron. 44 (6), 848–859.
Fang, C., Chen, S., Fu, C.S., 2003. Road sign detection and tracking. IEEE Trans. Veh. Technol. 52 (5), 1329–1341.
Fleyeh, H., 2006. Shadow and highlight invariant colour segmentation algorithm for traffic signs. In: IEEE Conference on Cybernetics and Intelligent Systems, Bangkok, Thailand, 7–9 June, pp. 1–7.
Fleyeh, H., Davami, E., 2011. Eigen-based traffic sign recognition. IET Intell. Transp. Syst. 5 (3), 190–196.
Golovinskiy, A., Funkhouser, T., 2009a. Min-cut based segmentation of point clouds. In: IEEE 12th International Conference on Computer Vision Workshops, Kyoto, Japan, 27 September–4 October, pp. 39–46.
Golovinskiy, A., Kim, V.G., Funkhouser, T., 2009b. Shape-based recognition of 3D point clouds in urban environments. In: IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October, pp. 2154–2161.
Gómez-Moreno, H., Maldonado-Bascón, S., Gil-Jiménez, P., Lafuente-Arroyo, S., 2010. Goal evaluation of segmentation algorithms for traffic sign recognition. IEEE Trans. Intell. Transp. Syst. 11 (4), 917–930.
González, Á., García-Garrido, M.Á., Llorca, D.F., Gavilán, M., Fernández, J.P., Alcantarilla, P.F., Parra, I., Herranz, F., Bergasa, L.M., Sotelo, M.Á., Revenga de Toro, P., 2011. Automatic traffic signs and panels inspection system using computer vision. IEEE Trans. Intell. Transp. Syst. 12 (2), 485–499.
Greenhalgh, J., Mirmehdi, M., 2012. Real-time detection and recognition of road traffic signs. IEEE Trans. Intell. Transp. Syst. 13 (4), 1498–1506.
Greenhalgh, J., Mirmehdi, M., 2015. Recognizing text-based traffic signs. IEEE Trans. Intell. Transp. Syst. 16 (3), 1360–1369.
Hazelhoff, L., Creusen, I.M., de With, P.H.N., 2014. Exploiting street-level panoramic images for large-scale automated surveying of traffic signs. Mach. Vis. Appl. 25 (7), 1893–1911.
Jin, J., Fu, K., Zhang, C., 2014. Traffic sign recognition with hinge loss trained convolutional neural networks. IEEE Trans. Intell. Transp. Syst. 15 (5), 1991–2000.
Jiang, Y., Meng, J., Yuan, J., Luo, J., 2015. Randomized spatial context for object search. IEEE Trans. Image Process. 24 (6), 1748–1762.
Körtgen, M., Park, G.J., Novotni, M., Klein, R., 2003. 3D shape matching with 3D shape contexts. In: 7th Central European Seminar on Computer Graphics, Budmerice, Slovakia, 22–24 April, vol. 3, pp. 5–7.
Larsson, F., Felsberg, M., Forssén, P.E., 2011. Correlating Fourier descriptors of local patches for road sign recognition. IET Comput. Vis. 5 (4), 244–254.
Lu, K., Ding, Z., Ge, S., 2012. Sparse-representation-based graph embedding for traffic sign recognition. IEEE Trans. Intell. Transp. Syst. 13 (4), 1515–1524.
Mathias, M., Timofte, R., Benenson, R., Van Gool, L., 2013. Traffic sign recognition – how far are we from the solution? In: International Joint Conference on Neural Networks, Dallas, USA, 4–9 August, pp. 1–8.
Meuter, M., Nunn, C., Görmer, S.M., Müller-Schneiders, S., Kummert, A., 2011. A decision fusion and reasoning module for a traffic sign recognition system. IEEE Trans. Intell. Transp. Syst. 12 (4), 1126–1134.
Møgelmose, A., Trivedi, M.M., Moeslund, T.B., 2012. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey. IEEE Trans. Intell. Transp. Syst. 13 (4), 1484–1497.
Møgelmose, A., Liu, D., Trivedi, M.M., 2015. Detection of U.S. traffic signs. IEEE Trans. Intell. Transp. Syst. 16 (6), 3116–3125.
Murray, S., Haughey, S., Brogan, M., Fitzgerald, C., McLoughlin, S., Deegan, C., 2011. Mobile mapping system for the automated detection and analysis of road delineation. IET Intell. Transp. Syst. 5 (4), 221–230.
Paclík, P., Novovicová, J., Duin, R.P.W., 2006. Building road-sign classifiers using a trainable similarity measure. IEEE Trans. Intell. Transp. Syst. 7 (3), 309–321.
Papon, J., Abramov, A., Schoeler, M., Wörgötter, F., 2013. Voxel cloud connectivity segmentation – supervoxels for point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June, pp. 2017–2034.
Pei, D., Sun, F., Liu, H., 2013. Supervised low-rank matrix recovery for traffic sign recognition in image sequences. IEEE Signal Process. Lett. 20 (3), 241–244.
Pu, S., Rutzinger, M., Vosselman, G., Elberink, S.O., 2011. Recognizing basic structures from mobile laser scanning data. ISPRS J. Photogram. Remote Sens. 66 (6), S28–S29.
Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M., 2008. Aligning point cloud views using persistent feature histograms. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September, pp. 3384–3391.
Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May, pp. 3212–3217.
Rusu, R.B., Cousins, S., 2011. 3D is here: point cloud library (PCL). In: IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May, pp. 1–4.
Ruta, A., Li, Y., Liu, X., 2010. Robust class similarity measure for traffic sign recognition. IEEE Trans. Intell. Transp. Syst. 11 (4), 846–855.
Salakhutdinov, R., Tenenbaum, J.B., Torralba, A., 2013. Learning with hierarchical-deep models. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1958–1971.
Salakhutdinov, R., Hinton, G., 2012. An efficient learning procedure for deep Boltzmann machines. Neural Comput. 24 (8), 1967–2006.
Seo, Y.W., Lee, J., Zhang, W., Wettergreen, D., 2015. Recognition of highway workzones for reliable autonomous driving. IEEE Trans. Intell. Transp. Syst. 16 (2), 708–718.
Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. ISPRS J. Photogram. Remote Sens. 93, 243–255.
Sivic, J., Zisserman, A., 2009. Efficient visual search of videos cast as text retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 31 (4), 591–606.
Soheilian, B., Paparoditis, N., Vallet, B., 2013. Detection and 3D reconstruction of traffic signs from multiple view color images. ISPRS J. Photogram. Remote Sens. 77, 1–20.
Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C., 2011. The German Traffic Sign Recognition Benchmark: A Multi-class Classification Competition. In: International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August, pp. 1453–1460.
Timofte, R., Zimmermann, K., Van Gool, L., 2014. Multi-view traffic sign detection, recognition, and 3D localisation. Mach. Vision Appl. 25 (3), 633–647.
Timofte, R., Van Gool, L., 2011. Sparse representation based projections. In: 22nd British Machine Vision Conference, Dundee, UK, 29 August–2 September, pp. 61.1–61.12.
Tombari, F., Salti, S., Stefano, L.D., 2010. Unique signatures of histograms for local surface description. In: 11th European Conference on Computer Vision, Crete, Greece, 5–11 September, pp. 356–369.
Vu, A., Yang, Q., Farrell, J.A., Barth, M., 2013. Traffic sign detection, state estimation, and identification using onboard sensors. In: 16th International IEEE Conference on Intelligent Transportation Systems, The Hague, The Netherlands, 6–9 October, pp. 875–880.
Wang, H., Wang, C., Luo, H., Li, P., Cheng, M., Wen, C., Li, J., 2014. Object detection in terrestrial laser scanning point clouds based on Hough forest. IEEE Geosci. Remote Sens. Lett. 11 (10), 807–1811.
Wang, H., Wang, C., Luo, H., Li, P., Chen, Y., Li, J., 2015. 3-D point cloud object detection based on supervoxel neighborhood with Hough forest framework. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 8 (4), 1570–1581.

Wen, C., Li, J., Luo, H., Yu, Y., Cai, Z., Wang, H., Wang, C., 2015. Spatial-related traffic sign inspection for inventory purposes using mobile laser scanning data. IEEE Trans. Intell. Transp. Syst. http://dx.doi.org/10.1109/TITS.2015.2418214.

Williams, K., Olsen, M.J., Roe, G.V., Glennie, C., 2013. Synthesis of transportation applications of mobile LiDAR. Remote Sens. 5 (9), 4652–4692.

Xu, S., 2009. Robust traffic sign shape recognition using geometric matching. IET Intell. Transp. Syst. 3 (1), 10–18.

Yang, B., Dong, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. ISPRS J. Photogram. Remote Sens. 81, 19–30.

Yang, Y., Luo, H., Xu, H., Wu, F., 2015. Towards real-time traffic sign detection and classification. IEEE Trans. Intell. Transp. Syst. http://dx.doi.org/10.1109/TITS.2015.2482461.

Yokoyama, H., Date, H., Kanai, S., Takeda, H., 2011. Pole-like objects recognition from mobile laser scanning data using smoothing and principal component analysis. ISPRS Arch. 38–5 (W12), 1–6.

Yu, Y., Li, J., Guan, H., Wang, C., Yu, J., 2015a. Semiautomated extraction of street light poles from mobile LiDAR point-clouds. IEEE Trans. Geosci. Remote Sens. 53 (3), 1374–1386.

Yu, Y., Li, J., Guan, H., Wang, C., 2015b. Automated extraction of urban road facilities using mobile laser scanning data. IEEE Trans. Intell. Transp. Syst. 16 (4), 2167–2181.

Yuan, X., Guo, J., Hao, X., Chen, H., 2015. Traffic sign detection via graph-based ranking and segmentation algorithms. IEEE Trans. Syst. Man Cybern. 45 (12), 1509–1521.

Yuan, X., Hao, X., Chen, H., Wei, X., 2014. Robust traffic sign recognition based on color global and local oriented edge magnitude patterns. IEEE Trans. Intell. Transp. Syst. 15 (4), 1466–1477.

Zaklouta, F., Stanciulescu, B., 2012. Real-time traffic-sign recognition using tree classifiers. IEEE Trans. Intell. Transp. Syst. 13 (4), 1507–1514.

Zheng, N., Tang, S., Cheng, H., Li, Q., Lai, G., Wang, F.W., 2004. Toward intelligent driver-assistance and safety warning systems. IEEE Intell. Syst. 19 (2), 8–11.

Zhou, L., Deng, Z., 2014. LiDAR and vision-based real-time traffic sign detection and recognition algorithm for intelligent vehicle. In: IEEE International Conference on Intelligent Transportation Systems, Qingdao, China, 8–11 October, pp. 578–583.

Zhou, Y., Wang, D., Xie, X., Ren, Y., Li, G., Deng, Y., Wang, Z., 2014. A fast and accurate segmentation method for ordered LiDAR point cloud of large-scale scenes. IEEE Geosci. Remote Sens. Lett. 11 (11), 1981–1985.