3-D Feature Matching for Point Cloud Object Extraction

Yongtao Yu, *Member, IEEE*, Haiyan Guan, *Senior Member, IEEE*, Dilong Li, Shenghua Jin, Taiyue Chen, Cheng Wang, *Senior Member, IEEE*, and Jonathan Li, *Senior Member, IEEE*

***Abstract*— Effective object extraction plays an important role in many point cloud-based applications. This letter proposes a 3-D feature matching framework for point cloud object extraction. To determine the optimal affine transformation parameters for each template feature point, a convex dissimilarity function and the locally affine-invariant geometric constraints are designed to construct the overall objective function. The 3-D feature matching framework is integrated into a point cloud object extraction workflow. Extraction results on six test data sets show that average completeness, correctness, quality, and F_1-measure of 0.96, 0.97, 0.93, and 0.96, respectively, are obtained in extracting light poles, vehicles, and palm trees. Comparative studies also confirm that the proposed method performs effectively and robustly, and exhibits superior or compatible performance over the other compared methods.**

***Index Terms*— Feature matching, light pole extraction, mobile light detection and ranging (LiDAR), object extraction, point cloud, vehicle extraction.**

I. Introduction

LIGHT detection and ranging (LiDAR) technology has provided an efficient and cost-effective solution to performing 3-D measurement of land features. To date, a variety of LiDAR systems, offering different levels of accuracies and measurement rates, have been manufactured for different applications and purposes, such as mobile, aerial, and terrestrial LiDAR systems, as well as multispectral and hyperspectral LiDAR systems. Due to the superior advantages of LiDAR systems and their resultant point cloud data, they have been used in a wide range of fields such as intelligent transportation systems, forest inventory, heritage documentation, surveying, and mapping engineering. Meanwhile, techniques and algorithms on point cloud data processing have attracted great attention and have been paid great efforts. As an important research topic, object extraction from point cloud data has been intensively studied and a number of achievements have been obtained in the literature.

In [1], 3-D Hough forest (HF) was designed for extracting road scene objects. To improve feature representations, supervoxel neighborhood-based HF [2] and deep-learning-based HF [3] were also developed for object extraction. Convolutional neural network (CNN) was used in [4] to extract instant objects. In this method, depth images were generated for appearance-based classification. In [5], a 3-D object matching (OM) framework was proposed to provide a template-driven strategy. To depict object-oriented feature representations, bag of contextual-visual words (BoCVWs) model was developed in [6]. The BoCVWs model was able to characterize the contextual statistical features of point cloud objects. In [7], a preRoIpooling convolution technique was presented for vehicle extraction. In addition, two-level point processes [8], grid-cell-based method [9], geodesic morphology [10], object-based point cloud analysis [11], decision trees [12], etc., were also exploited for vehicle extraction tasks.

In [13], hierarchical Euclidean clustering was adopted to extract buildings using a divide-and-conquer strategy. Taking advantage of the geometric features of buildings, neural oscillator networks [14] and marked point process [15] were proposed for delineating building boundaries. In addition, multisensor data fusion techniques were also exploited for building extraction [16], [17]. To assist in road infrastructure inventory, a bag-of-visual-phrases model was developed in [18] to extract traffic signs. Considering the linear structures and highly reflective properties, segmentation and filtering strategies were also applied to extract traffic signs [19], [20]. Pairwise 3-D shape context (PSC) [21] and ball falling (BF) and location guided segmentation [22] were proposed to extract light poles. In [23], a complete processing chain composed of Laplacian smoothing, Latent Dirichlet allocation topic model, and Markov random field was developed to extract transmission lines. Moreover, methods, such as profile analysis [24], Bayesian approach [25], and a dual growing method [26], were suggested to extract trees. Instead of processing 3-D point clouds, some methods converted them into georeferenced feature images to extract road surface features, such as cracks [27], road markings [28], and manhole covers [29].

Manuscript received February 11, 2019; revised April 19, 2019; accepted May 14, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61603146 and Grant 41671454, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20160427, and in part by the Natural Science Research in Colleges and Universities of Jiangsu Province under Grant 16KJB520006. *(Corresponding author: Yongtao Yu.)*

Y. Yu, S. Jin, and T. Chen are with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian 223003, China (e-mail: allennessy.yu@gmail.com; allennessy@hyit.edu.cn).

H. Guan is with the School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: guanhy.nj@nuist.edu.cn).

D. Li is with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430072, China (e-mail: scholar.dll@gmail.com).

C. Wang and J. Li are with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China (e-mail: cwang@xmu.edu.cn; junli@xmu.edu.cn).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2019.2918073

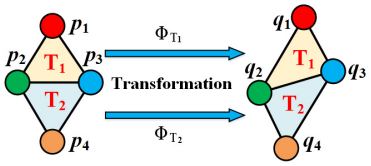


Fig. 1. Illustration of the local affine transformation model. T_1 and T_2 are two triangles generated through Delaunay Triangulation. $\mathbf{\Phi}_{T_1}$ and $\mathbf{\Phi}_{T_2}$ are the transformation parameters for template feature points on triangles T_1 and T_2.

In this letter, we develop a 3-D feature matching framework for point cloud object extraction. To determine the optimal transformation parameters for each template feature point, a continuous and convex dissimilarity function and the locally affine-invariant geometric constraints were, respectively, designed to measure the feature and geometric dissimilarities between the template and scene feature points.

II. 3-D Feature Matching

A. Problem Formulation

The issue of feature matching can be defined as matching a group of template feature points, representing an object of interest, to another group of scene feature points, representing a scene containing an instance of that object. The matched scene feature points should preserve similar local features and relative spatial relationships to the corresponding template feature points. Different from the existing feature matching strategies seeking for point-to-point matching patterns, in this letter, we propose an affine-transformation-based feature matching framework, which dedicates to determine the optimal transformation parameters for each template feature point such that the matching position (unnecessary being a specific scene feature point) of each template feature point falls into the scene with similar local appearance and geometric structure.

Denote n_t and n_s as the numbers of template and scene feature points, respectively. Let $\mathbf{P} = \{p_i = [x_{p_i}, y_{p_i}, z_{p_i}]^T | i = 1, 2, \ldots, n_t\}$ and $\mathbf{Q} = \{q_j = [x_{q_j}, y_{q_j}, z_{q_j}]^T | j = 1, 2, \ldots, n_s\}$ be the sets of template and scene feature points, respectively. Then, our feature matching objective is to determine an optimal affine transformation function $T_i(\mathbf{\Phi}_i)$: $\boldsymbol{R}^n \rightarrow \boldsymbol{R}^3$, $i = 1, 2, \ldots, n_t$ for each template feature point p_i based on the scene feature points in $\mathbf{Q}$. $T_i(\mathbf{\Phi}_i)$ transforms template feature point p_i into a position in the scene with transformation parameters $\mathbf{\Phi}_i \in \mathbf{R}^n$. In this letter, we formulate the affine transformation function as follows:

$$T_i(\mathbf{\Phi}_i) = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha_4 & \alpha_5 & \alpha_6 \\ \alpha_7 & \alpha_8 & \alpha_9 \end{bmatrix} \begin{bmatrix} x_{p_i} \\ y_{p_i} \\ z_{p_i} \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \tag{1}$$

where $T_i(\mathbf{\Phi}_i)$: $\boldsymbol{R}^{12} \rightarrow \boldsymbol{R}^3$ computes the matching position of template feature point p_i under transformation parameters $\mathbf{\Phi}_i = [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, \xi_1, \xi_2, \xi_3]^T \in \mathbf{R}^{12}$. To provide a local affine transformation model allowing certain global deformations, first, we organize the template feature points into a Delaunay Triangulation representation. Then, every three feature points defining a triangle share the same set of affine transformation parameters (see Fig. 1). In this way, we define a separate affine transformation function for each triad of template feature points.

One objective of feature matching is to match each template feature point p_i to its matching position $T_i(\mathbf{\Phi}_i)$ in the scene with the constraint of preserving similar local appearance. To this end, we define a dissimilarity measure function $mc_i(q)$: $\boldsymbol{R}^3 \rightarrow \boldsymbol{R}$, $i = 1, 2, \ldots, n_t$ for each template feature point p_i to measure the local feature dissimilarities between p_i and its matching position q. Specifically, two feature points having similar local appearances should produce a low dissimilarity measure result.

To solve the feature matching problem, our overall objective is to determine a group of optimal transformation parameters $\mathbf{\Phi}_{T_1}, \mathbf{\Phi}_{T_2}, \ldots, \mathbf{\Phi}_{T_m}$, where m is the number of triangles, for the Delaunay-triangulated representation of template feature points $p_1, p_2, \ldots, p_{n_t}$ to minimize the following objective function:

$$\begin{aligned} \min_{\mathbf{\Phi}_{T_1}, \mathbf{\Phi}_{T_2}, \ldots, \mathbf{\Phi}_{T_m}} & \sum_{t=1}^{m} \sum_{p_i \in T_t} mc_i(T_i(\mathbf{\Phi}_{T_t})) \\ & + \lambda \cdot G(\mathbf{\Phi}_{T_1}, \mathbf{\Phi}_{T_2}, \ldots, \mathbf{\Phi}_{T_m}) \\ \text{s.t. } & C_j(\mathbf{\Phi}_{T_1}, \mathbf{\Phi}_{T_2}, \ldots, \mathbf{\Phi}_{T_m}) \leq 0, \; j = 1, 2, \ldots, n_c, \\ & T_i(\mathbf{\Phi}_{T_a}) = T_i(\mathbf{\Phi}_{T_b}) \quad \text{for all } p_i \in T_a \text{ and } p_i \in T_b. \end{aligned} \tag{2}$$

where $mc_i(T_i(\mathbf{\Phi}_{T_t}))$ computes the local feature dissimilarity between p_i and $T_i(\mathbf{\Phi}_{T_t})$. $G(\mathbf{\Phi}_{T_1}, \mathbf{\Phi}_{T_2}, \ldots, \mathbf{\Phi}_{T_m})$ is a convex term reflecting the local geometric dissimilarities between the template feature points and the matching positions. $C_j(\mathbf{\Phi}_{T_1}, \mathbf{\Phi}_{T_2}, \ldots, \mathbf{\Phi}_{T_m}) \leq 0$, $j = 1, 2, \ldots, n_c$ defines a set of convex constraints. Here, n_c is the number of constraints. $T_i(\mathbf{\Phi}_{T_a}) = T_i(\mathbf{\Phi}_{T_b})$ defines a set of equality constraints that guarantee one template feature point's matching positions computed by different triangles' transformation parameters are the same position. λ weights the feature dissimilarity and the geometric dissimilarity measures. In this study, it is set to be 0.5 to favor feature dissimilarity more to well handle geometric deformations of objects.

B. Feature Dissimilarity Measure

To characterize the local appearance of a feature point, we adopt the fast point feature histograms (FPFH) feature descriptor [30]. Denote $\text{cost}_{i,\ j}$, $i = 1, 2, \ldots, n_t$, $j = 1, 2, \ldots, n_s$ as the feature dissimilarity between template feature point p_i and scene feature point q_j. Then, we define $\text{cost}_{i,j}$ as the square root of the χ^2 distance between the FPFH descriptors of p_i and q_j:

$$cost_{i,j} = \sqrt{\sum_k \frac{\left(H_{p_i}^k - H_{q_j}^k\right)^2}{H_{p_i}^k + H_{q_j}^k}} \tag{3}$$

where $H_{p_i}^k$ and $H_{q_j}^k$ are the kth components of the FPFH descriptors of feature points p_i and q_j, respectively. Then, for each template feature point p_i, we define a discrete feature dissimilarity function as follows:

$$MC_i(q_j) = cost_{i,j}, \quad j = 1, 2, \ldots, n_s, \; q_j \in \mathbf{Q}. \tag{4}$$

The above discrete feature dissimilarity function still provides a point-to-point matching pattern and is nonconvex. To solve this problem, we relax each $MC_i(q_j)$ to construct a continuous and convex dissimilarity function $mc_i(q)$ that can be effectively optimized through convex optimization techniques.

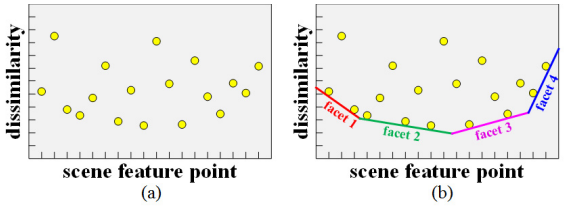


Fig. 2. Illustrations of (a) feature dissimilarities viewed as a 4-D point set, and (b) lower convex hull (facets).

For each template feature point p_i, we organize all the scene feature points together with their feature dissimilarities $\text{cost}_{i,j}$ as a set of 4-D points, whose first three dimensions are the position of a scene feature point and the fourth dimension corresponds to the feature dissimilarity [see Fig. 2(a)]. Then, we design the convex dissimilarity function $mc_i(q)$ for template feature point p_i based on the lower convex hull of the 4-D point set associated with p_i with respect to the feature dissimilarity dimension. As shown in Fig. 2(b), the facets are the lower convex hull of the 4-D point set. Denote $\{w = a_k x + b_k y + c_k z + d_k | k = 1, 2, \ldots, n_\text{f}\}$ as the hyperplane functions defining the n_f facets on the lower convex hull. $[a_k,\ b_k,\ c_k,\ d_k]^\text{T}$ are the hyperplane parameters. Then, we define the continuous convex dissimilarity function as follows:

$$mc_i([x, y, z]^T) = \max_k (a_k x + b_k y + c_k z + d_k), \quad k = 1, 2, \ldots, n_\text{f} \tag{5}$$

where $[x,\ y,\ z]^\text{T}$ can be any position in the scene domain. To effectively minimize the dissimilarity function in (5), we convert it into an equivalent linear programming problem:

$$\begin{aligned} &\min_{x,y,z} \ mc_i([x, y, z]^T) \Leftrightarrow \\ &\min_{x,y,z,u_i} \ u_i \\ &\text{s.t. } a_k x + b_k y + c_k z + d_k \le u_i, \quad k = 1, 2, \ldots, n_\text{f} \end{aligned} \tag{6}$$

where u_i is an auxiliary variable representing the upper bound of $mc_i([x,\ y,\ z]^\text{T})$.

To minimize the dissimilarity measure $mc_i(T_i(\mathbf{\Phi}_{T_t}))$ in (2), we rewrite the affine transformation function $T_i(\mathbf{\Phi}_{T_t})$ into $T_i(\mathbf{\Phi}_{T_t}) = [x_i(\mathbf{\Phi}_{T_t}), y_i(\mathbf{\Phi}_{T_t}), z_i(\mathbf{\Phi}_{T_t})]^T$, where $x_i(\mathbf{\Phi}_{T_t})$, $y_i(\mathbf{\Phi}_{T_t})$, and $z_i(\mathbf{\Phi}_{T_t})$ are the three components of the matching position of template feature point p_i. Then, by substituting x, y, and z in (6), we obtain the following convex optimization model:

$$\begin{aligned} &\min_{\mathbf{\Phi}_{T_t}} \ mc_i(T_i(\mathbf{\Phi}_{T_t})) \Leftrightarrow \\ &\min_{\mathbf{\Phi}_{T_t}, u_i} \ u_i \\ &\text{s.t. } a_k x_i(\mathbf{\Phi}_{T_t}) + b_k y_i(\mathbf{\Phi}_{T_t}) + c_k z_i(\mathbf{\Phi}_{T_t}) + d_k \le u_i, \\ &\qquad k = 1, 2, \ldots, n_\text{f} \end{aligned} \tag{7}$$

C. Geometric Dissimilarity Measure

Recall that the matching positions should preserve similar relative spatial relationships to the template feature points. To this end, we formulate the geometric dissimilarity measure using locally affine-invariant geometric constraints. Based on the Delaunay triangulation representation of the template feature points, each template feature point p_i can be represented by an affine combination of its connected neighbors on the triangulated mesh as follows:

$$p_i = \sum_{p_k \in N_{p_i}} w_{ik} p_k \tag{8}$$

where N_{p_i} is the set of connected neighbors of p_i on the triangulated mesh, and w_{ik} is the affine combination coefficient. By using the locally affine-invariant geometric constraints, the geometric dissimilarity measure in (2) is designed as follows:

$$\begin{aligned} &G(\mathbf{\Phi}_{T_1}, \mathbf{\Phi}_{T_2}, \ldots, \mathbf{\Phi}_{T_m}) \\ &\quad = \sum_{t=1}^{m} \sum_{p_i \in T_t} \left\| T_i(\mathbf{\Phi}_{T_t}) - \sum_{p_k \in N_{p_i}} w_{ik} T_k(\mathbf{\Phi}_{T_s}) \right\|_2 . \end{aligned} \tag{9}$$

Finally, by combining the feature dissimilarity measure in (7) and the geometric dissimilarity measure in (9), we obtain the overall objective function, which can be effectively solved through convex optimization techniques. After optimization, the matching position of template feature point p_i can be computed by $T_i(\mathbf{\Phi}_{T_t})$ and the result of the objective function reflects the cost for matching the template with the scene.

III. Results and Discussion

A. Data Sets

To evaluate the performance of the proposed 3-D feature matching framework on point cloud object extraction, we collected five point cloud data sets using the RIEGL VMX-450 mobile LiDAR system in Xiamen, China. The RIEGL VMX-450 mobile LiDAR system integrating two laser scanners can provide a maximum measurement rate of 1.1 million measurements per second and a line scan speed of up to 400 scans per second. The first data set covering a road length of about 4260 m was used to extract light poles. The second data set covering a road distance of about 2522 m was also collected for light pole extraction. The third data set covering a road segment of about 776 m was selected for evaluating vehicle extraction performance. The fourth data set covering a road section of about 2108 m was also used to assess vehicle extraction performance. The fifth data set covering a road area of about 1813 m was used to evaluate palm tree extraction performance. In addition, the public KITTI vision benchmark [31] was also used to test vehicle extraction performance.

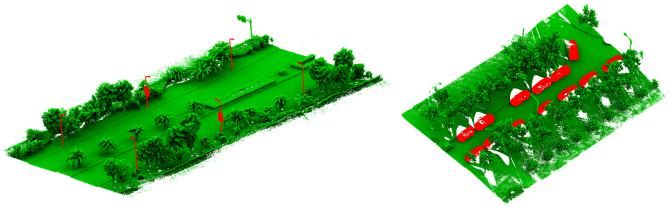
B. Object Extraction

We integrated the proposed 3-D feature matching framework into a point cloud object extraction workflow. As proposed in our previous work [5], for a point cloud scene being processed, first, the large-volume ground points were removed from the scene using the voxel-based upward growing method. This method can rapidly filter out ground points and maintain a good completeness of the off-ground objects. Then, the discrete and unordered off-ground points were grouped into separate clusters using the Euclidean distance clustering method. The Euclidean distance clustering method can effectively segment the isolated objects; however, the overlapped objects cannot be well segmented. Next, to further segment such clusters containing multiple objects, we adopted the

TABLE I

OBJECT EXTRACTION RESULTS AND PROCESSING TIME OBTAINED ON DIFFERENT POINT CLOUD DATA SETS

Object	Dataset	Quantitative Evaluation				Time (minutes)
		cpt	*crt*	*qat*	*fmr*	
Light pole	I	0.97	0.97	0.95	0.97	36.74
	II	0.95	0.95	0.91	0.95	18.93
Vehicle	III	0.96	0.97	0.93	0.96	19.88
	IV	0.97	0.98	0.95	0.97	54.06
	KITTI	0.91	0.97	0.90	0.94	127.52
Palm tree	V	0.98	0.98	0.96	0.98	41.25
Average		0.96	0.97	0.93	0.96	-



(a) (b)

Fig. 3. Subset of (a) light pole extraction and (b) vehicle extraction results.

voxel-based normalized cut segmentation method to obtain individual objects. Finally, we applied the proposed 3-D feature matching framework to a point cloud template and each of the segmented individual objects to extract the specific type of objects.

To perform feature matching, first, for the point cloud template and each of the segmented individual objects, we oversegmented them into a supervoxel representation using the voxel cloud connectivity segmentation (VCCS) algorithm [32]. Then, for each supervoxel, the point closest to the supervoxel center was selected as the feature point. In this way, we obtained a group of template feature points representing the template and a group of scene feature points representing an object. Next, we carried out the proposed 3-D feature matching framework between the template feature points and each group of the scene feature points. After optimization, the result of the overall objective function in (2), namely, matching cost, reflected the similarity between the template and the object. That is, the lower the matching cost, the more similar the two objects. Finally, the matching costs from all individual objects were thresholded to extract the specific type of objects.

To evaluate the performance of the proposed 3-D feature matching-based object extraction method, we applied it to the aforementioned six point cloud data sets to extract light poles, vehicles, and palm trees, respectively. To quantitatively assess the object extraction results, we adopted the following four measures: *completeness* (*cpt*), *correctness* (*crt*), *quality* (*qat*), and F_1*-measure* (*fmr*) [5]. Table I shows the details of the object extraction results obtained on the six data sets. As reflected in Table I, for each of the data sets, most of the objects of interest were correctly extracted. However, due to the similarities of some other objects to the template, they were falsely extracted as true targets caused by low matching costs. For example, some polelike objects had very low matching costs to the light pole template; thus, they were falsely labeled as light poles. In addition, caused by occlusions of other nearby targets, some objects of interest were scanned with severe incompleteness. These objects obtained very high matching costs; therefore, they were failed to be extracted.

TABLE II

OBJECT EXTRACTION RESULTS AND PROCESSING TIME OBTAINED BY DIFFERENT METHODS

Object	Dataset	Method	Quantitative Evaluation				Time (minutes)
			cpt	*crt*	*qat*	*fmr*	
Light pole	I	HF	0.91	0.94	0.86	0.92	37.19
		OM	0.95	**0.97**	0.92	0.96	35.21
		PSC	0.93	0.94	0.88	0.93	**31.57**
		BF	0.96	**0.97**	0.93	0.96	37.15
		VoxelNet	**0.97**	**0.97**	**0.95**	**0.97**	39.22
		Vote3Deep	0.94	0.96	0.91	0.95	36.78
	II	HF	0.89	0.91	0.82	0.90	19.13
		OM	0.94	**0.95**	0.89	0.94	18.42
		PSC	0.91	0.92	0.84	0.91	**16.68**
		BF	**0.95**	**0.95**	**0.90**	**0.95**	20.32
		VoxelNet	**0.95**	0.94	**0.90**	0.94	21.07
		Vote3Deep	0.93	0.93	0.87	0.93	19.79
Vehicle	III	HF	0.92	0.94	0.87	0.93	20.21
		dHF	0.94	0.96	0.90	0.95	19.18
		CNN	0.95	**0.97**	0.92	**0.96**	19.72
		BoCVWs	**0.96**	**0.97**	**0.93**	**0.96**	**17.32**
		VoxelNet	**0.96**	0.96	**0.93**	**0.96**	22.75
		Vote3Deep	0.95	0.94	0.89	0.94	20.59
	IV	HF	0.93	0.94	0.88	0.93	55.19
		dHF	0.95	0.96	0.92	0.95	53.28
		CNN	0.96	0.97	0.93	0.96	54.77
		BoCVWs	0.96	**0.98**	0.94	**0.97**	**49.22**
		VoxelNet	**0.97**	**0.98**	**0.95**	**0.97**	58.43
		Vote3Deep	0.96	0.96	0.93	0.96	56.07
	KITTI	HF	0.72	0.91	0.70	0.80	129.89
		dHF	0.75	0.92	0.72	0.83	124.62
		CNN	0.77	0.92	0.73	0.84	125.44
		BoCVWs	0.85	0.94	0.82	0.89	**113.53**
		VoxelNet	**0.88**	**0.96**	**0.86**	**0.92**	150.41
		Vote3Deep	0.76	0.94	0.74	0.84	131.92

Specifically, the proposed method achieved average completeness, correctness, quality, and F_1-measure of 0.96, 0.97, 0.93, and 0.96, respectively, in extracting light poles, vehicles, and palm trees from the six data sets. For visual inspections, Fig. 3 shows a subset of object extraction results from the test data sets. The proposed method was implemented using C++ and run on an HP Z820 workstation (CPU: 2 GHz). The processing time of the proposed method on the six data sets was also reported in Table I for computational performance analysis. As a whole, the proposed 3-D feature matching-based method showed promising performance in point cloud object extraction tasks.

C. Comparative Study

To further evaluate the performance and accuracy of the proposed 3-D feature matching-based object extraction method, we conducted a set of comparative experiments with the following six methods: HF method [2], OM method [5], PSC method [21], BF method [22], VoxelNet [33], and Vote3Deep [34], for light pole extraction on data sets I and II, and the following six methods: HF method [2], deep HF method (dHF) [3], CNN method [4], BoCVWs method [6], VoxelNet [33], and Vote3Deep [34], for vehicle extraction on data sets III, IV, and KITTI. Table II presents the object extraction results. For light pole extraction, the HF method obtained lower performances than the other five methods, whereas the BF and VoxelNet methods performed similar to our proposed method. For vehicle extraction, the BoCVWs and VoxelNet methods exhibited similar performances to our

proposed method and achieved better extraction results than the other four methods. In addition, the processing time of each method was also reported in Table II for computational performance comparison. Comparatively, our proposed method performed robustly and showed superior or compatible performance over the other compared methods in extracting the specific type of objects from point cloud data.

IV. Conclusion

This letter has presented a 3-D feature matching framework for point cloud object extraction. Rather than seeking for a point-to-point matching pattern, an optimal affine transformation was defined for each template feature point to compute its corresponding matching position in the scene. A continuous and convex dissimilarity function and the locally affine-invariant geometric constraints were designed to measure the feature and geometric dissimilarities, respectively, between the template and the scene. The 3-D feature matching framework has been integrated into a point cloud object extraction workflow to extract light poles, vehicles, and palm trees on six selected mobile LiDAR point cloud data sets. Extraction results showed that the proposed method achieved average completeness, correctness, quality, and F_1-measure of 0.96, 0.97, 0.93, and 0.96, respectively. A set of comparative studies also confirmed that the proposed method performed robustly and obtained superior or compatible performance over the other compared methods in extracting point cloud objects.

References

[1] H. Wang *et al.*, "Object detection in terrestrial laser scanning point clouds based on Hough forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1807–1811, Oct. 2014.

[2] H. Wang, C. Wang, H. Luo, P. Li, Y. Chen, and J. Li, "3-D point cloud object detection based on supervoxel neighborhood with Hough forest framework," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 4, pp. 1570–1581, Apr. 2015.

[3] Y. Yu, J. Li, H. Guan, and C. Wang, "Automated detection of three-dimensional cars in mobile laser scanning point clouds using DBM-Hough-forests," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 4130–4142, Jul. 2016.

[4] A. Börcs, B. Nagy, and C. Benedek, "Instant object detection in LiDAR point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 992–996, Jul. 2017.

[5] Y. Yu, J. Li, H. Guan, and C. Wang, "Automated extraction of urban road facilities using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2167–2181, Aug. 2015.

[6] Y. Yu, J. Li, H. Guan, C. Wang, and C. Wen, "Bag of contextual-visual words for road scene object detection from mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3391–3406, Dec. 2016.

[7] Y. Zeng *et al.*, "RT3D: Real-time 3-D vehicle detection in LiDAR point cloud for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3434–3440, Oct. 2018.

[8] A. Börcs and C. Benedek, "Extraction of vehicle groups in airborne LiDAR point clouds with two-level point processes," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1475–1489, Mar. 2015.

[9] W. Yao and U. Stilla, "Comparison of two methods for vehicle extraction from airborne LiDAR data toward motion analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 4, pp. 607–611, Jul. 2011.

[10] W. Yao, S. Hinz, and U. Stilla, "Automatic vehicle extraction from airborne LiDAR data of urban areas aided by geodesic morphology," *Pattern Recognit. Lett.*, vol. 31, no. 10, pp. 1100–1108, 2010.

[11] J. Zhang, M. Duan, Q. Yan, and X. Lin, "Automatic vehicle extraction from airborne LiDAR data using an object-based point cloud analysis method," *Remote Sens.*, vol. 6, no. 9, pp. 8405–8423, Sep. 2014.

[12] J. Eum, M. Bae, J. Jeon, H. Lee, S. Oh, and M. Lee, "Vehicle detection from airborne LiDAR point clouds based on a decision tree algorithm with horizontal and vertical features," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 409–418, Jan. 2017.

[13] S. Sun and C. Salvaggio, "Aerial 3D building detection and modeling from airborne LiDAR point clouds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1440–1449, Jun. 2013.

[14] C. Liu, B. Shi, X. Yang, N. Li, and H. Wu, "Automatic buildings extraction from LiDAR data in urban area by neural oscillator network of visual cortex," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 6, no. 4, pp. 2008–2019, Aug. 2013.

[15] Y. Yu, J. Li, H. Guan, and C. Wang, "A marked point process for automated building detection from LiDAR point-clouds," *Remote Sens. Lett.*, vol. 4, no. 11, pp. 1127–1136, Oct. 2013.

[16] G. Zhou and X. Zhou, "Seamless fusion of LiDAR and aerial imagery for building extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7393–7407, Nov. 2014.

[17] M. Awrangjeb, M. Ravanbakhsh, and C. S. Fraser, "Automatic detection of residential buildings using LiDAR data and multispectral imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 5, pp. 457–467, Sep. 2010.

[18] Y. Yu, J. Li, C. Wen, H. Guan, H. Luo, and C. Wang, "Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 113, pp. 106–123, Mar. 2016.

[19] C. Wen *et al.*, "Spatial-related traffic sign inspection for inventory purposes using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 27–37, Jan. 2016.

[20] H. Guan, W. Yan, Y. Yu, L. Zhong, and D. Li, "Robust traffic-sign detection and classification using mobile LiDAR data with digital images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 5, pp. 1715–1724, May 2018.

[21] Y. Yu, J. Li, H. Guan, C. Wang, and J. Yu, "Semiautomated extraction of street light poles from mobile LiDAR point-clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1374–1386, Mar. 2015.

[22] F. Wu *et al.*, "Rapid localization and extraction of street light poles in mobile LiDAR point clouds: A supervoxel-based approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 292–305, Feb. 2017.

[23] J. Yang and Z. Kang, "Voxel-based extraction of transmission lines from airborne LiDAR point cloud data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 10, pp. 3892–3904, Oct. 2018.

[24] C. Paris, D. Valduga, and L. Bruzzone, "A hierarchical approach to three-dimensional segmentation of LiDAR data at single-tree level in a multilayered forest," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 4190–4203, Jul. 2016.

[25] T. Lahivaara *et al.*, "Bayesian approach to tree detection based on airborne laser scanning data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2690–2699, May 2014.

[26] L. Li, D. Li, H. Zhu, and Y. Li, "A dual growing method for the automatic extraction of individual trees from mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 120, pp. 37–52, Oct. 2016.

[27] H. Guan *et al.*, "Iterative tensor voting for pavement crack extraction using mobile laser scanning data," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1527–1537, Mar. 2015.

[28] H. Guan, J. Li, Y. Yu, C. Wang, M. Chapman, and B. Yang, "Using mobile laser scanning data for automated extraction of road markings," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 93–107, Jan. 2014.

[29] Y. Yu, H. Guan, and Z. Ji, "Automated detection of urban road manhole covers using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3258–3269, Dec. 2015.

[30] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, Kobe, Japan, May 2009, pp. 3212–3217.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3354–3361.

[32] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "Voxel cloud connectivity segmentation—Supervoxels for point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2027–2034.

[33] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4490–4499.

[34] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, Singapore, May/Jun. 2017, pp. 1355–1361.