

Automatic registration of large-scale urban scene point clouds based on semantic feature points

Bisheng Yang ^{a,b,*}, Zhen Dong ^{a,b,*}, Fuxun Liang ^a, Yuan Liu ^a

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

^b Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China



ARTICLE INFO

Article history:

Received 9 July 2015

Received in revised form 15 December 2015

Accepted 16 December 2015

Keywords:

Laser scanning point clouds

Point cloud segmentation

Semantic feature point extraction

Urban scene mapping

Multi-view registration

ABSTRACT

Point clouds collected by terrestrial laser scanning (TLS) from large-scale urban scenes contain a wide variety of objects (buildings, cars, pole-like objects, and others) with symmetric and incomplete structures, and relatively low-textured surfaces, all of which pose great challenges for automatic registration between scans. To address the challenges, this paper proposes a registration method to provide marker-free and multi-view registration based on the semantic feature points extracted. First, the method detects the semantic feature points within a detection scheme, which includes point cloud segmentation, vertical feature lines extraction and semantic information calculation and finally takes the intersections of these lines with the ground as the semantic feature points. Second, the proposed method matches the semantic feature points using geometrical constraints (3-point scheme) as well as semantic information (category and direction), resulting in exhaustive pairwise registration between scans. Finally, the proposed method implements multi-view registration by constructing a minimum spanning tree of the fully connected graph derived from exhaustive pairwise registration. Experiments have demonstrated that the proposed method performs well in various urban environments and indoor scenes with the accuracy at the centimeter level and improves the efficiency, robustness, and accuracy of registration in comparison with the feature plane-based methods.

© 2016 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

1. Introduction

Terrestrial laser scanning (TLS) captures three-dimensional (3D) point clouds with high flexibility and precision and is widely used for various applications such as 3D model reconstruction (Vosselman et al., 2004; Pu and Vosselman, 2009), cultural heritage management (Guarnieri et al., 2006; Yang and Zang, 2014; Montuori et al., 2014), forest surveys (Kankare et al., 2013; García et al., 2015), landslide monitoring (Prokop and Panholzer, 2009), and urban planning (Pieraccini et al., 2006). TLS has line-of-sight instruments and limited measurement range, multiple scans from different viewpoints are needed for full coverage of a large urban environment. Because each scan refers to its own local coordinate reference, a registration step transforming all scans into a uniform coordinate reference system has to be carried out.

Point clouds collected by terrestrial laser scanning (TLS) from large-scale urban scenes contain a wide variety of objects (buildings, cars, pole-like objects, and others) with symmetric and incomplete structures, and relatively low-textured surfaces, all of which pose great challenges for automatic registration between scans. To address these challenges, many scientific studies have been carried out to register point clouds in urban scenes, mostly with a coarse to fine strategy (Salvi et al., 2007; Guo et al., 2014; Restrepo et al., 2014). For fine registration, the iterative closest point (ICP) algorithm and its variants, described in Besl and McKay (1992) and Rusinkiewicz and Levoy (2001), are the commonly used standard approaches (Weinmann et al., 2011). However, the ICP algorithm may converge to a local minimum or even be non-convergent without a priori alignment of the point clouds. Therefore, coarse registration methods are needed to search for good initial alignments between scans. Automatic coarse registration methods can be classified into three categories: feature point-based methods, feature line-based methods, and feature plane-based methods, according to the kind of features used for registration.

* Corresponding authors at: State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China.

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

In general, point features are most commonly used for point cloud registration. Böhm and Becker (2007) explored the application of the SIFT method on the automatic marker-free registration of TLS data, by using SIFT key points extracted from the reflectance data of the scans. Barnea and Filin (2008) proposed a key point-based autonomous registration method, which extracts the key points from a panoramic range image and establishes the correspondences between them using 3D Euclidean distance. Rusu et al. (2008) estimated a set of robust 16D features to describe the geometry of each point feature locally to find an initial alignment. Experiments have shown that the algorithm is invariant to pose and sampling density and can cope well with noisy laser scans. Weinmann et al. (2011) extracted characteristic two-dimensional (2D) points based on SIFT features and used radiometric and geometric information on 3D conjugate points to estimate the transformation parameters between two adjacent scans. The proposed approach was successfully applied to a benchmarked data set, resulting in fast and accurate estimation of the transformation parameters. Theiler et al. (2014) proposed a marker-free registration scheme that matches the extracted DoG (difference-of-Gaussians) or Harris key points using 4PCS (4-point congruent sets). Weber et al. (2015) proposed a pipeline to register unordered point clouds acquired by Kinect sensors automatically using fast point-feature histograms (FPFH) as described in Rusu et al. (2008) for initial alignment between two adjacent scans.

Many studies using line or plane features to register point clouds have also been widely reported. Stamos and Leordeanu (2003) proposed an autonomous registration method based on line features, which extracts the intersection lines of neighboring planes and calculates the transformation parameters between adjacent scans using at least two corresponding line pairs. Habib et al. (2005) also reported a methodology for registering photogrammetric and LiDAR data using 3D straight-line features instead of points. Yang and Zang (2014) used spatial curves as matching primitives to calculate the initial transformation parameters between the scanned point clouds of freeform surfaces (e.g., statues, cultural heritage artifacts).

Dold and Brenner (2006) presented a registration method based on plane patches that calculates rotation and translation parameters separately between two point clouds using at least three corresponding pairs of planar patches. Von Hansen (2006) presented a method for automatic and marker-free coarse registration of TLS data, which calculates the coarse parameters between two point clouds based on single plane correspondences. Theiler and Schindler (2012) developed a registration approach based on virtual tie points generated by intersecting triples of planes. Virtual tie points are matched using their descriptors, such as one divided by the condition number, the intersection angles between planes, the extent of planar segments, and the smoothness of planes.

Although the reported methods are generally able to align TLS point clouds based on the corresponding points, lines, or planes, they still have more or less difficulties in dealing with data sets from large-scale urban scenes. Most of the feature point-based methods are more sensitive to point densities and noises in comparison with the feature line-based and feature plane-based methods. The feature line-based methods take advantage only of lines derived from buildings and rarely use line features derived from pole-like objects, these methods may have difficulty dealing with data sets in the suburbs with few buildings. The feature plane-based methods require more stringent overlapping conditions and works well in cases there are at least three corresponding pairs of planar patches between neighboring scans. Moreover, the semantic attributes (e.g., shape, height, category, direction and so on) of the extracted points, lines and planes are seldom used in the existing methods. To overcome these limitations, this paper proposes a marker-free and multi-view registration method for

large-scale urban scene point clouds based on the semantic feature points. The main contribution of the proposed method is to detect a small set of semantic feature points and match them using both geometrical constraints (3-point scheme) and their semantic information (category and direction).

Following this introduction, the key steps of the proposed method will be elaborated. Then the proposed method will be validated in experimental studies, after which conclusions will be drawn.

2. Automatic registration of large-scale urban scene point clouds

The proposed method provides a marker-free and multi-view registration method for large-scale urban scene point clouds based on semantic feature points extracted. The proposed method consists of two key steps: semantic feature point extraction and point cloud registration. First, the method detects the semantic feature points within a detection scheme, which consists of three sequential steps: point cloud segmentation, vertical feature lines extraction (lines derived from pole-like objects and vertical planes) and semantic information calculation (category and direction) and finally takes the intersections of the vertical feature lines with the ground as the semantic feature points. Second, the proposed method matches the semantic feature points using geometrical constraints (3-point scheme) as well as semantic information (category and direction), and eliminates the mismatches by geometric consistency tests to calculate pairwise transformations; next, it implements multi-view registration by constructing a minimum spanning tree of the fully connected graph derived from exhaustive pairwise transformations. The framework of the proposed method is illustrated in Fig. 1.

2.1. Semantic feature points extraction

The pipeline of the proposed semantic feature point extraction method consists of three sequential steps: point cloud segmentation (Section 2.1.1), vertical feature line extraction (Section 2.1.2) and semantic information calculation (Section 2.1.3).

2.1.1. Point cloud segmentation

Segmenting point clouds into meaningful segments is a precondition of vertical feature line extraction. The proposed method segments point cloud using the following steps. First, ground points

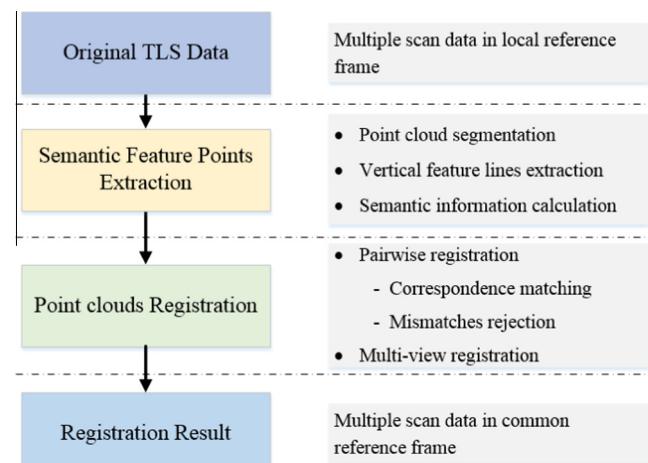


Fig. 1. Point cloud registration framework.

are removed before point cloud segmentation using the method of Hernández and Marcotegui (2009). Then, the non-ground points are divided into cross sections by a series of horizontal planes with the height interval ($CS_{interval}$), as shown in Fig. 3b, where ground points, cross sections of the non-ground points are dotted in blue and red respectively. The thickness ($CS_{thickness}$) of the cross sections is predefined according to the point densities of the point clouds. Successively, each cross-section is segmented independently applying the cluster algorithm based on connected component analysis. The variable point densities caused by occlusion and scanning distances pose great challenges for selecting a fixed suitable distance threshold of connected component analysis algorithm. Segmenting scattered points with a fixed neighborhood size (distance threshold) is likely to generate partial results (Yang and Dong, 2013). It has been found that the distance threshold between adjacent points in areas with low point density should be larger than those in regions of high density to achieve good segmentation (Fig. 2). Hence, the adaptive distance threshold should be well determined for good segmentation. Neighboring points are compared to each other and labelled with a same segment label if the distance between them is less than the adaptive distance threshold (dT_s) that is calculated as:

$$dT_s = \text{Mean}(P_s) + \text{Variation}(P_s) \quad (1)$$

where $\text{Mean}(P_s)$ and $\text{Variation}(P_s)$ are the mean and standard deviation of the length of all edges directly linking to P_s in the 2D Delaunay triangulation, as shown in Fig. 2, where $P_s^i, i \in \{1, 2, 3, 4, 5, 6\}$ are the points directly linking to P_s . The segmentation result is shown in Fig. 3c.

2.1.2. Vertical feature line extraction

To extract vertical feature lines from the segments acquired in Section 2.1.1, only segments belong to pole-like objects and vertical planes are considered. In the case of the segments belonging to pole-like objects, the segments should satisfy the cylinder equation:

$$\| (P - Q) \times Ca \| - r = 0, \quad (2)$$

where $P = (x_p, y_p, z_p)$ is a point on the cylinder surface, $Q = (x_q, y_q, z_q)$ is a point on the axis, $Ca = (Ca_x, Ca_y, Ca_z)$ is the direction of the cylinder axis with a unit length, r is the radius. Moreover, r ranges from R_{min} to R_{max} , and $Ca = (Ca_x, Ca_y, Ca_z)$ is approximately parallel to the Z-direction. If a segment satisfies Eq. (2) and the above conditions, the center point of the segment is calculated and marked as pole-like points ($Pole_{fp}$), as shown in Fig. 3d.

As for the segments belong to vertical planes, the segments should satisfy the 3D line equation:

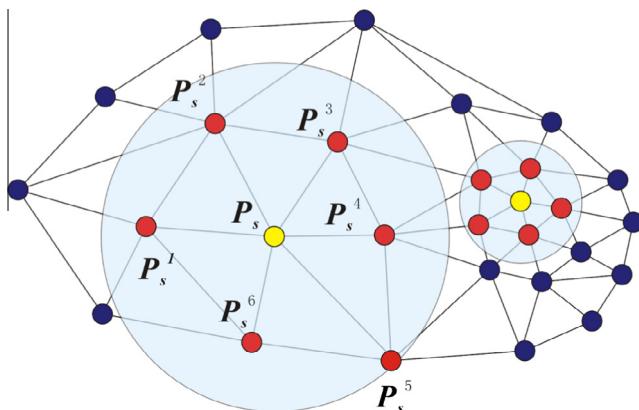


Fig. 2. Calculating the adaptive distance threshold for point cloud segmentation.

$$\frac{x - x_0}{a} = \frac{y - y_0}{b} = \frac{z - z_0}{c}, \quad (3)$$

where (x_0, y_0, z_0) and (x, y, z) are points on a straight line and (a, b, c) is the direction of the line. Moreover, the length of the fitted line is no less than L_{min} , and the direction of the fitted line (a, b, c) is approximately perpendicular to the Z-direction. Hence, the segments belong to vertical planes can be extracted by Eq. (3) and the above conditions. In addition, the starting and ending vertices of the fitted lines can also be calculated. Suppose that one starting or ending vertex is the intersection of two horizontal lines, the starting or ending vertex is marked as intersection ($Building_{IP}$). Otherwise, it is marked as vertices ($Building_{VP}$), as shown in Fig. 3d.

Hence, the three kinds of points (pole-like objects, intersections, and vertices) are clustered as vertical feature lines if the neighboring points distribute along the vertical direction. The categories of the vertical feature lines are determined by the majority of points belonging to the line, as shown in Fig. 3e.

2.1.3. Semantic information calculation

Once the three kinds of vertical feature lines have been extracted, each vertical feature line is described by one 9-tuple:

$$L_{\text{Feature}}(Pt_{\text{lowest}}, Pt_{\text{highest}}, Pt_{\text{num}}, L_{\text{height}}, L_{\text{id}}, L_{\text{category}}, L_{\text{radius}}, Pl_{\text{direction1}}, Pl_{\text{direction2}}),$$

where Pt_{lowest} , Pt_{highest} , Pt_{num} , L_{height} , and L_{id} are the lowest point, the highest point, the number of points, the height, and the unique identifier of the vertical feature line. L_{category} is the category of the vertical feature line, in which lines derived from pole-like points, intersections, or vertices are marked as L_{PFL} , L_{BBL} , and L_{BIL} respectively. L_{radius} is the radius of the line's supporting pole, which is only valid on the vertical feature lines derived from pole-like objects and calculated by averaging the radius of the fitted cylinders belonging to the pole, as shown in Fig. 3d. $Pl_{\text{direction1}}$ and $Pl_{\text{direction2}}$ are the directions of the line's supporting planes, as shown in Fig. 3e. Suppose that one vertical feature line is clustered from the intersections, both $Pl_{\text{direction1}}$ and $Pl_{\text{direction2}}$ have valid values. Otherwise, only $Pl_{\text{direction1}}$ has a valid value.

Finally, the intersection points (semantic feature points) of the three extracted kinds of vertical feature lines and the ground are calculated and denoted by V_{PFL} , V_{BBL} , and V_{BIL} respectively, as shown in Fig. 3e. Meanwhile, the semantic information of the vertical feature lines are assigned to their corresponding semantic feature points. The semantic feature points will be used for searching correspondences between adjacent scans in the next registration step.

2.2. Point cloud registration

2.2.1. Correspondence matching

The correspondence matching step establishes correspondences between a source point cloud P_s and a target point cloud P_t . Thereby, the semantic feature points (V_{PFL} , V_{BBL} , and V_{BIL}) extracted from each scan are first triangulated (three-dimensional triangulation network) independently for searching the corresponding semantic feature points from the adjacent scans. Hence, the number of triangles constructed by the exhaustive combination of the semantic points is C_{NI}^3 , where NI is the number of semantic feature points and C_{NI}^3 is a formula for the number of possible combinations of 3 points from a set of NI points. To reduce the computing burden for searching corresponding triangles, triangles meeting any one of the following conditions are eliminated before correspondence matching step:

- The triangle is isosceles or equilateral.
- The three vertices of the triangle are almost collinear.

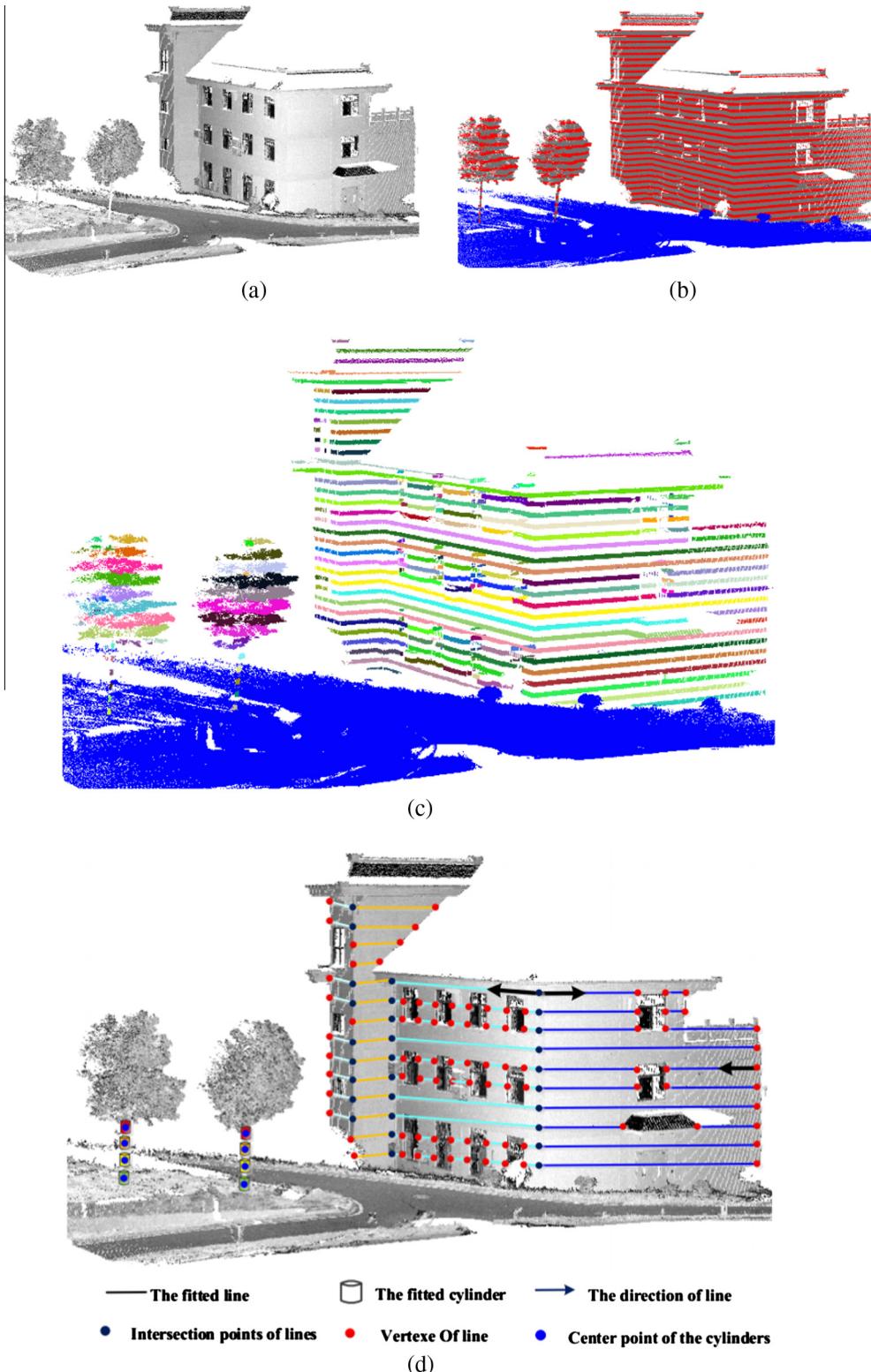


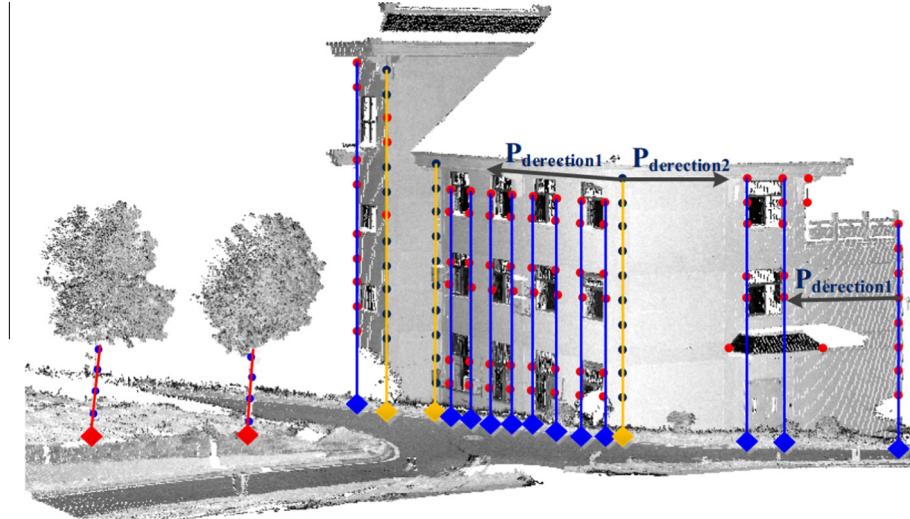
Fig. 3. Diagram of semantic feature point extraction: (a) TLS point clouds colored by intensity of reflection, (b) generated cross sections of non-ground points, (c) segmentation colored by segment flags, (d) the extracted pole-like points, vertices and intersections and (e) the extracted vertical feature lines and semantic feature points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- The length of the shortest edge of the triangle is less than a specified value.

Then remaining triangles are indexed in a hash table whose horizontal axis and vertical axes represent the triangle perimeters and triangle areas, respectively, as shown in Fig. 4. Hence, each tri-

angle can be inserted into the corresponding hash bin according to the index number calculated by Eq. (4).

$$\begin{cases} \text{line_index} = \left[\frac{\text{Area}}{\text{Bin}} \right] \\ \text{row_index} = \left[\frac{\text{Perimeter}}{\text{Bin}} \right] \end{cases} \quad (4)$$



||| The vertical feature lines derived from pole-like points, vertices and intersections.
◆◆◆ The corresponding semantic feature points of vertical feature lines.

(e)

Fig. 3 (continued)

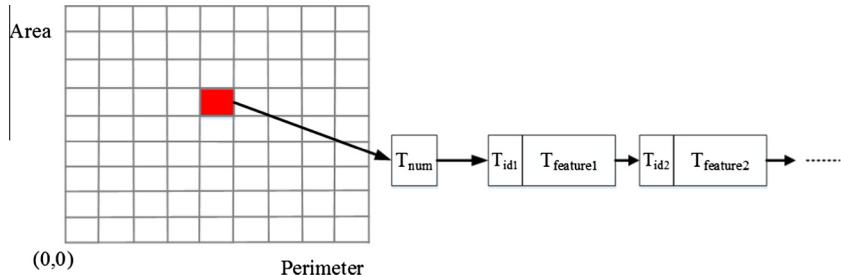


Fig. 4. Structure of the hash table.

where *line_index* and *row_index* are the indexes of line and row, the *Perimeter* and *Area* represent the perimeter and area of the triangle, *Bin* is the size of bin in the hash table, and $\lceil \cdot \rceil$ is the symbol of ceil. The structure of the hash table is illustrated in Fig. 4, where every bin in the hash table has an associated linked list which saves the number of triangles in the bin (T_{num}), their unique identities (T_{id}), and their features ($T_{feature}$). The triangles constructed by the semantic feature points of each scan are kept in one hash table, providing fast searching for corresponding semantic feature points between adjacent scans.

Let $\{T_1^S, T_2^S \dots T_{NS}^S\}$ and $\{T_1^t, T_2^t \dots T_{NT}^t\}$ be the set of triangles in the source point cloud P_s , and the target point cloud P_t , NS and NT be the number of triangles in P_s and P_t , respectively. In order to establish correspondences of triangles between P_s and P_t , for each triangle T_i^S in $\{T_1^S, T_2^S \dots T_{NS}^S\}$ the most similar triangle T_j^t in $\{T_1^t, T_2^t \dots T_{NT}^t\}$ is searched with the help of the hash table. And the similarity between two triangles is calculated according to Eq. (5).

$$\text{similarity} = \frac{1}{|\Delta_1| + |\Delta_2| + |\Delta_3|} \quad (5)$$

where $\Delta_1, \Delta_2, \Delta_3$ are the length differences between the three corresponding edges, respectively. Afterwards, T_i^S and T_j^t are tested on the following three conditions.

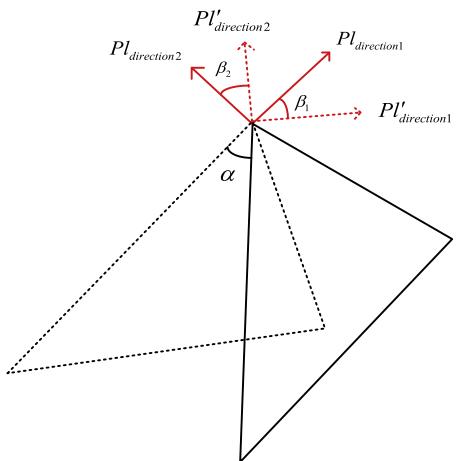


Fig. 5. Consistency diagram of the rotation angle.

Constraint 1. Let T_i^S be one of the triangles in the source point P_s and T_j^t be one of the triangles in the target point P_t . To search the most similar triangle of T_i^S from the target points, T_j^t will be qualified. To search the similar triangle of T_j^t from the source points, T_i^S will also be qualified. T_j^t and T_i^S is a matching pair.

Constraint 2.

$$\frac{\langle T_i^S - T_m^t \rangle}{\langle T_i^S - T_j^t \rangle} \geq U_{matching} \& \& \frac{\langle T_j^t - T_n^S \rangle}{\langle T_j^t - T_i^t \rangle} \geq U_{matching}. \quad (6)$$

where $\langle \rangle$ is the sum of the length differences between the corresponding edges of a triangle pair. $U_{matching}$ is a threshold greater than 1. T_j^t and T_m^t are the most similar and the second best

similar triangles in the target point P_t of T_i^S , T_i^S and T_n^S are the most similar and the second best similar triangles in the source point P_s of T_j^t . T_j^t and T_m^t may be quite similar, T_i^S and T_n^S may be quite similar as well. This special case makes it hard to determine (T_i^S, T_j^t) or (T_i^S, T_m^t) is a matching pair. This constraint ensures that the matching is clear and not based on random effects (Lowe, 2004).

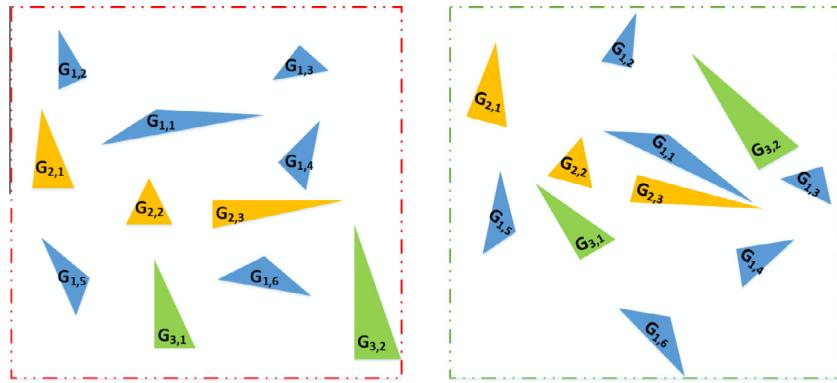


Fig. 6. Geometric consistency test between triangles. Different colors represent different groups after geometric consistency test and G_{ij} represents the j th corresponding triangle in group i . In this figure, the 6 corresponding triangles in group 1 are used to calculate the transformation matrix between adjacent scans. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

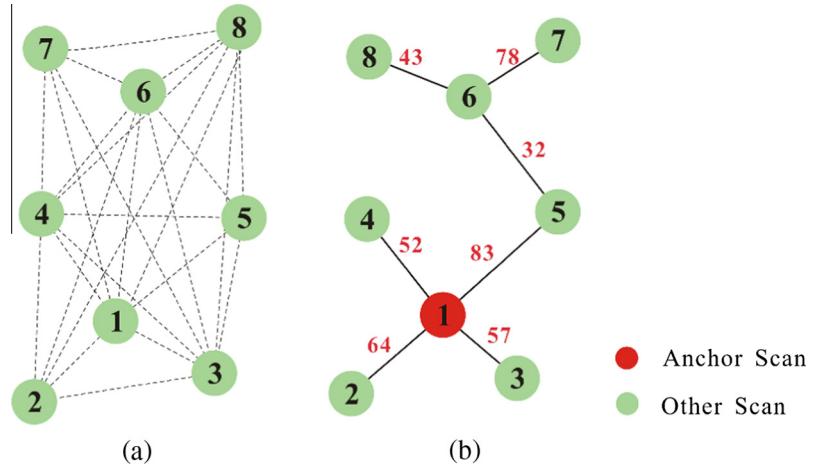


Fig. 7. Pairwise registration and multi-view registration graphs: (a) exhaustive pairwise registration graph, and (b) the MST of undirected weighted graph. The red numbers on the edges are the numbers of corresponding triangles between the scans. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Descriptions of the TLS datasets.

		Data set A	Data set B	Data set C
Number of scans		32	18	9
Number of points (million)		349	282	133
Resolution	Horizontal (degree)	0.04	0.05	0.04
	Vertical (degree)	0.04	0.05	0.04
Accuracy	Range (mm)	5	30	5
	Angle (mrad)	0.3	0.4	0.3
Dimensions (m)		1450 * 650 * 65	600 * 400 * 30	300 * 450 * 10
Point density (points/m ²)		442	326	673
Average overlap		40%	20%	50%

Constraint 3. The corresponding vertices of the triangle pair T_i^S and T_j^t should belong to the same type of semantic feature points (V_{PFL} , V_{BBL} , or V_{BIL}). Moreover, suppose that corresponding vertices are associated with the semantic feature points V_{BIL} or V_{BBL} , the rotation angle around the z-axis between the triangle pair T_i^S and T_j^t can be calculated by the three corresponding vertices or by the difference of $Pl_{direction1}$ or $Pl_{direction2}$ of the semantic feature points. Both calculated solutions should result in a consistent rotation angle. As shown in Fig. 5, α is the rotation angle around the z-axis calculated from the corresponding vertices of the triangle pair T_i^S and T_j^t , $\beta_1 = Pl_{direction1} - Pl'_{direction1}$ and $\beta_2 = Pl_{direction2} - Pl'_{direction2}$. β_1 and β_2 should be approximately equal to the rotation angle α .

If the three conditions between T_i^S and T_j^t are satisfied, then a correspondence of triangle is established and added to the set of correspondences between P_s and P_t .

2.2.2. Mismatches rejection

In light of Constraints 1–3, the corresponding triangles between the source and target point clouds are searched. In a further effort to eliminate wrong corresponding triangles, the geometric consistency test (Aldoma et al., 2012) is used to partition the potential corresponding triangles into different groups. Given a list of corresponding triangle pairs $\{C_1, C_2, \dots, C_{NC}\}$, for any two correspondences triangle pairs $C_K = \{T_i^S, T_j^t\}$ and $C_L = \{T_n^S, T_m^t\}$, the

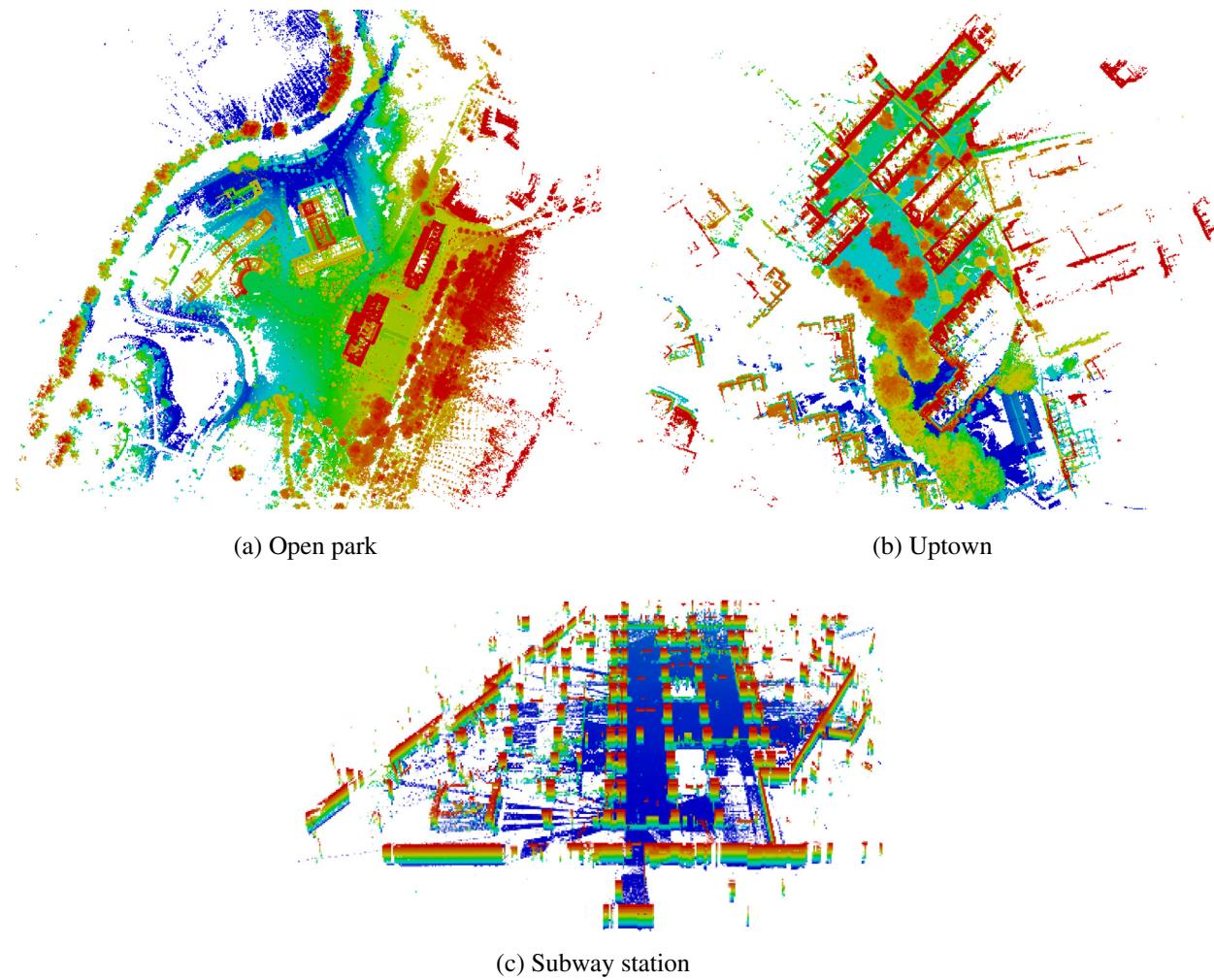


Fig. 8. Overview of the three TLS datasets: (a) dataset A (top view), (b) dataset B (top view), and (c) dataset C (the ceiling was removed for a better visual effect).

Table 2

Parameter setting for experimental datasets.

Procedure		Parameter	Descriptor	Value
Semantic feature points extraction	Point cloud segmentation	$CS_{interval}$	Height interval of the cross sections	0.4 m
		$CS_{thickness}$	Thickness of the cross sections	0.2 m
	Vertical feature line extraction	R_{min}	Min radius of fitted cylinder	0.05 m
		R_{max}	Max radius of fitted cylinder	0.5 m
		L_{min}	Min length of fitted line	1.0 m
Point cloud registration	Correspondence matching	$U_{matching}$	The ratio between the best matching and second best matching	1.1
	Mismatches rejection	$GC_{constraint}$	The maximum distance of geometric constraint	0.3 m

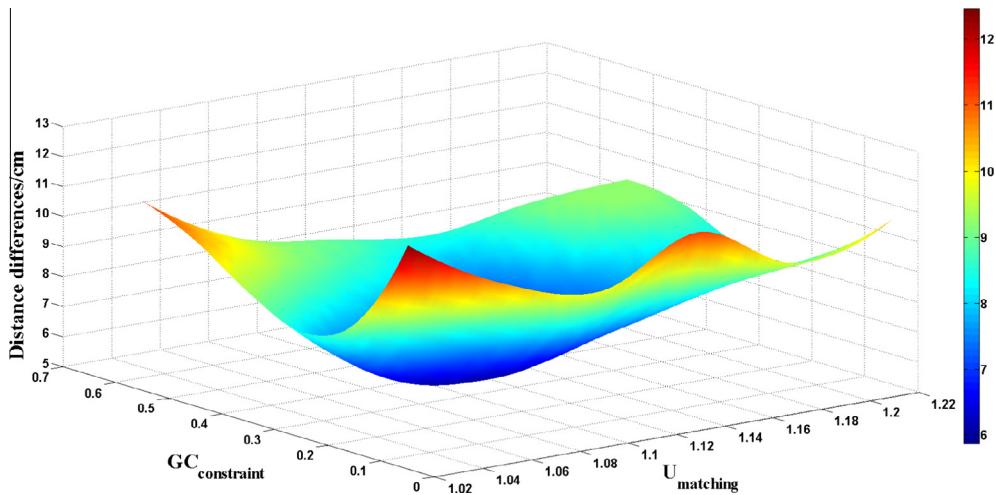


Fig. 9. Registration errors with different parameters values.

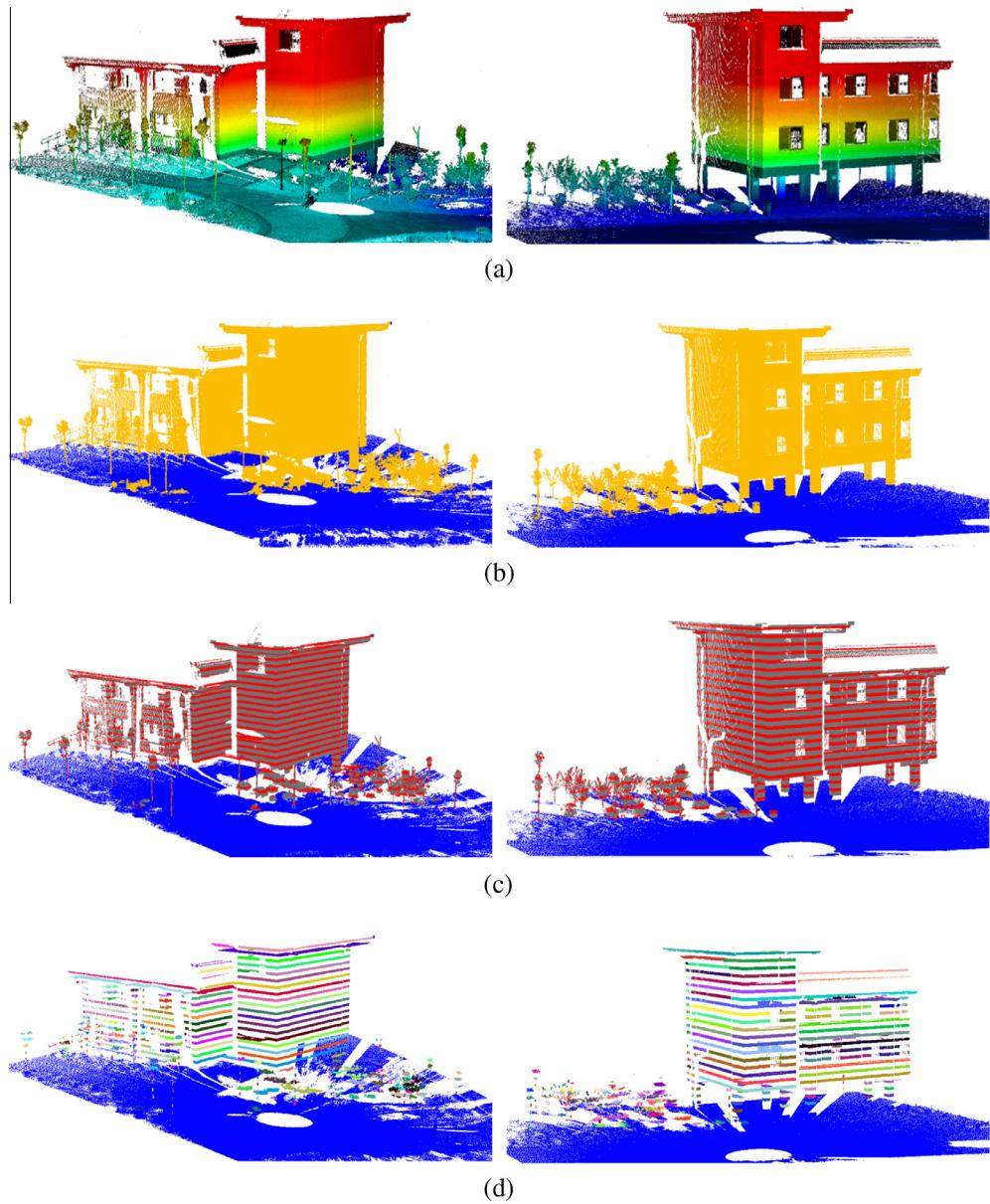


Fig. 10. Semantic feature point extraction: (a) original data, (b) segmentation of ground and non-ground points, (c) generated cross sections of non-ground points, (d) segments of a cross section, and (e) and (f) vertical feature lines and semantic feature points extraction.

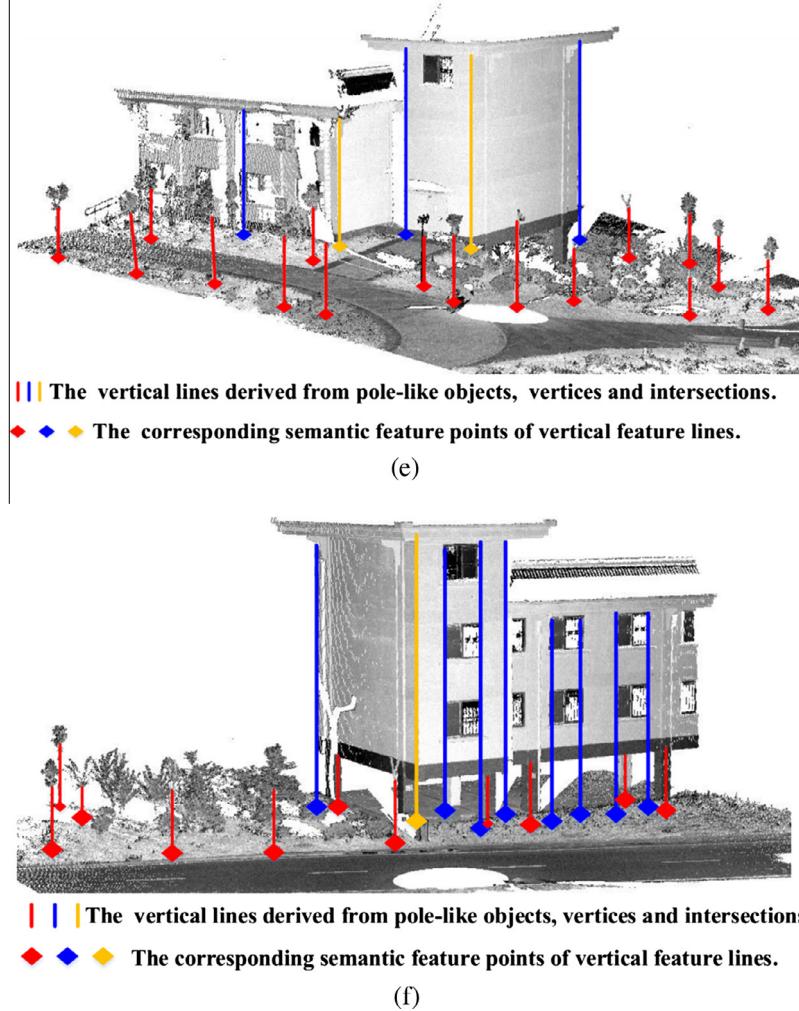


Fig. 10 (continued)

correspondences triangle pairs are partitioned into the same group if they satisfy Eq. (7). Repeat the same procedure until all the corresponding triangle pairs are grouped.

$$\text{abs}(\|T_i^S - T_n^S\| - \|T_j^t - T_m^t\|) < GC_{\text{constraint}}, \quad (7)$$

where $\|\cdot\|$ is the Euclidean distance between the centroids of two triangles, T_i^S, T_n^S are the triangles in the source point cloud P_s and T_j^t, T_m^t are the corresponding triangles in the target point cloud P_t , and $GC_{\text{constraint}}$ is a small value.

The larger the group is, the more likely it contains the true corresponding pairs, as shown in Fig. 6. Hence, the corresponding vertices of the matching triangles in the largest group are used to calculate the pairwise registration between two scans by minimizing Eq. (8):

$$\delta = \sum_{l=1}^m \|P_{l,\text{target}} - T(P_{l,\text{source}})\|, T(P_{l,\text{source}}) = A * P_{l,\text{source}} + B, \quad (8)$$

where A is a 3×3 rotation matrix, B is a 3D translation vector, $P_{l,\text{source}}, P_{l,\text{target}}$ are the corresponding semantic feature points in the source and target point clouds respectively, m is the number of corresponding points, and δ is calculated as the sum of the Euclidean distances between the corresponding points in the source and target point clouds.

2.2.3. Multi-view registration

The registration of multiple scans (multi-view registration) is difficult because of the large nonlinear search space and the huge quantity of TLS data involved (Kang et al., 2009) and should be effectively dealt with to reduce accumulation errors introduced by pairwise registration (Huber and Hebert, 2003). To perform multi-view registration, the method should solve the three following problems: (1) determining which scans are adjacent or the view order of the scans and calculating the transform matrixes between them; (2) determining which scan is the anchor scan; and (3) calculating transformation matrixes from any other point clouds to anchor scan point cloud.

First, the exhaustive pairwise registration is carried out by the above proposed method, and an undirected weighted graph of exhaustive pairwise scans is constructed, denoted by $G(V, E, W)$, where V represents the multiple scans, E indicates the transformation between two scans, and each transformation E is assigned one weight W , as shown in Fig. 7a. The weight W equals one divided by the number of corresponding triangles between the two scans. The weight W indicates that the larger the number of corresponding triangles between two scans is, the more likely the two scans are adjacent and the transform matrix calculated based on the corresponding triangles is more reliable. Therefore, the minimum spanning tree (MST) of the undirected weighted graph is generated using Kruskal algorithm (Kruskal, 1956) as shown in Fig. 7b.

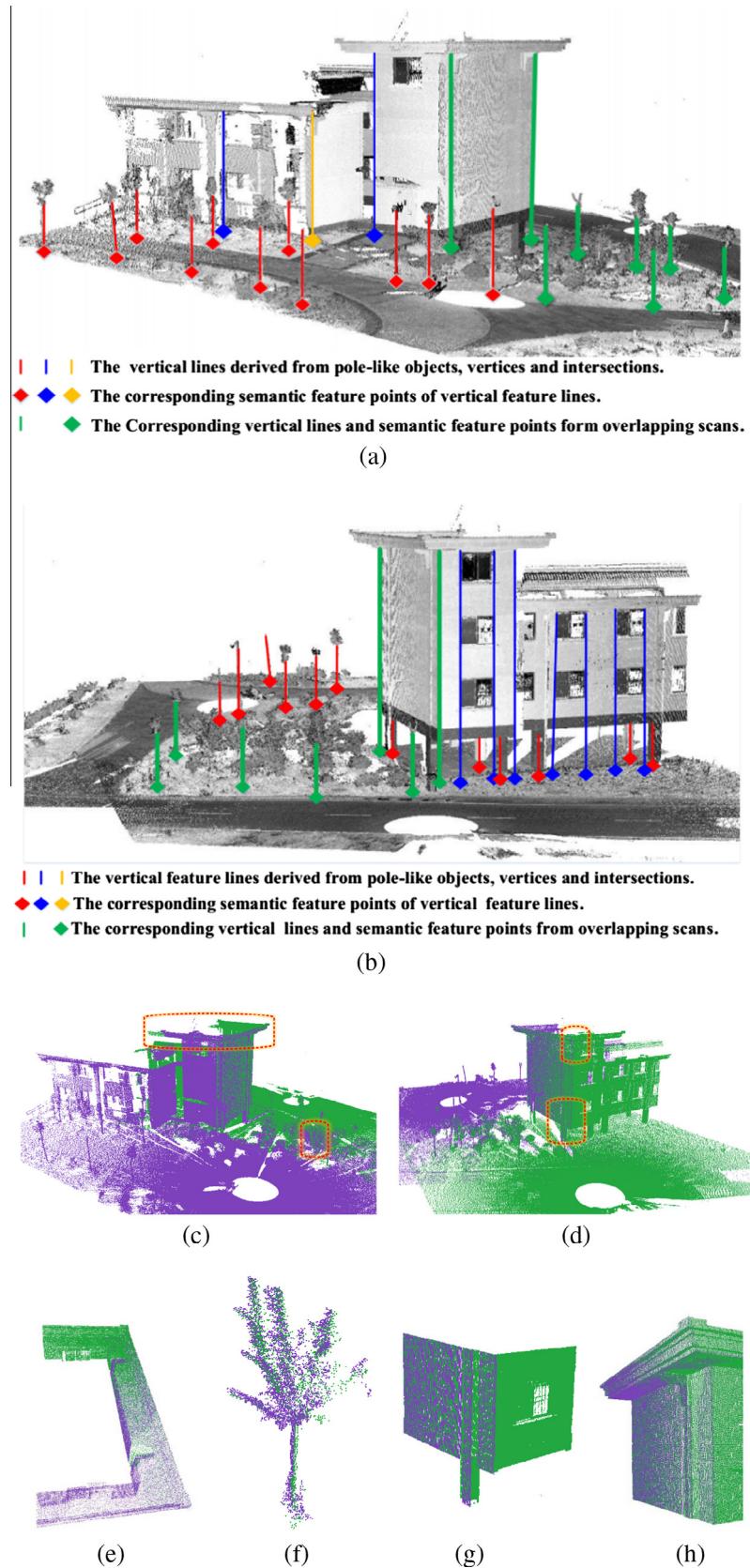


Fig. 11. Pairwise registration, (a) and (b) corresponding vertical feature lines and semantic feature points between a pair of overlapping scans, (c) and (d) pairwise registration result, and (e)–(h) details of registration result.

According to the MST, we can reject the paths of pairwise transformation that contain registrations of lower confidence, and determine the adjacent relations as well as the transformation matrixes between scans. So the paths of the MST indicate high probability of reliable registration between scans. Second, in order to reduce accumulation errors, the method takes the scan with the largest degree as the anchor scan, as shown in Fig. 7b. Third, the method refines the adjacent scans in the MST using the pairwise level ICP (Besl and McKay, 1992) and updates the transformation matrixes between them; next, the transformation matrixes from any other point clouds to anchor scan point cloud are acquired

based on the registration paths in MST and the refined transformation matrixes between adjacent scans.

3. Results and analysis

3.1. Data description

The efficiency, accuracy, and robustness of the proposed method were checked using three TLS datasets of different scenes. Datasets A and C were captured using the RIEGL VZ-400 laser scanner system with a field of view of 360° in the horizontal direction

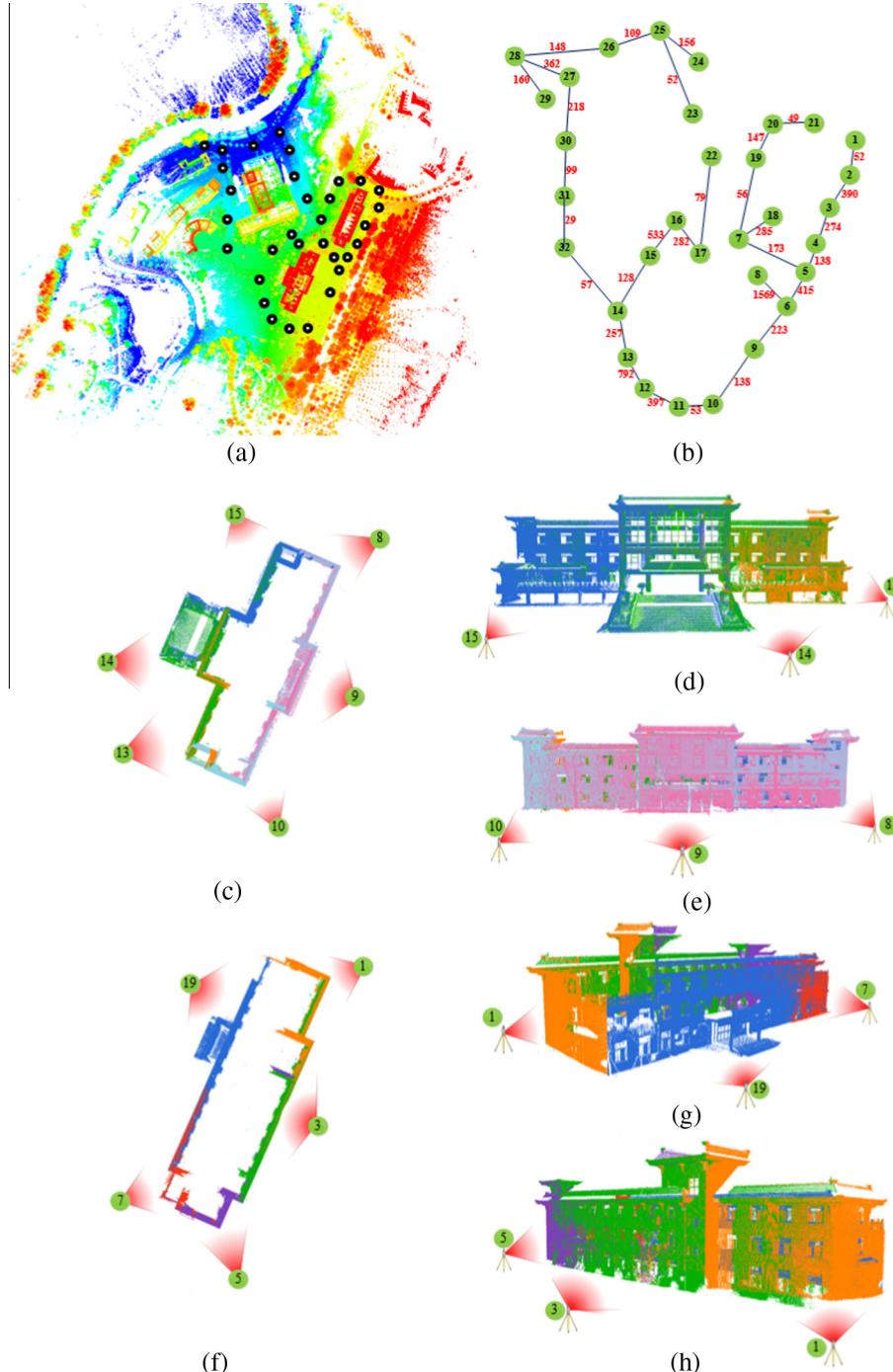


Fig. 12. Multi-view registration results for dataset A: (a) dataset A as aligned from all 32 scans, with shading representing differences in height; (b) the multi-view registration path; (c) top view of first selected building; (d) front view of first selected building; (e) back view of first selected building; (f) top view of second selected building; (g) left view of second selected building; (h) right view of second selected building.

and 100° in the vertical direction, a maximum range of 600 m, and a maximum measurement rate of 300,000 points/s. Dataset B was captured using a Stonex X300 laser scanner with a field of view of 360° in the horizontal direction and 180° in the vertical direction, a maximum range of 300 m, and a maximum measurement rate of 40,000 points/s. Table 1 provides a description of the datasets, and Fig. 8 shows an overview of the three scenes.

It can be seen from Fig. 8 that the point clouds contain a wide variety of objects (e.g., trees, buildings, cars, poles) with significant disparity in size, shape, level of structure complexity, and point density. Dataset A represents an open park with few buildings and more pole-like objects. Dataset B represents an uptown with dense buildings and pole-like objects, resulting in a narrow field of view and few overlaps due to occlusion. Dataset C is an underground subway station with many symmetric artificial structures and noise caused by specular reflection.

3.2. Point cloud registration results

3.2.1. Generation parameters analysis

The parameters for semantic feature point extraction and point cloud registration for the three datasets are listed in Table 2.

Parameters $CS_{interval}$ and $CS_{thickness}$ are the height interval and thickness of the cross section, controlling the trade-off between the efficiency and accuracy of point cloud segmentation. When $CS_{interval}$ is large and $CS_{thickness}$ is small, the point cloud segmentation procedure generates less cross sections and each cross section has less points, so the process is more efficient. Otherwise, the method can acquire more segments with higher accuracy at the expense of more additional operation cost. To check a balance between efficiency and accuracy, $CS_{thickness}$ was specified as five of average point span and the $CS_{interval}$ was specified as double of $CS_{thickness}$ through trial and error. Parameters R_{min} and R_{max} are the minimum and maximum radius of the fitted cylinder, controlling the number of vertical feature lines derived from pole-like objects. The minimum and maximum of the radius were chosen after visual analyses considering the smaller poles (e.g. thin small sign poles (0.05 m)) and larger posts (e.g. big trees (0.5 m)). Parameter $U_{matching}$ is the minimum ratio between the best matching triangle pairs and second best matching triangle pairs, and $GC_{constraint}$ is the maximum distance threshold in geometric consistency test. The performance of registration method against different settings of parameters $U_{matching}$ and $GC_{constraint}$ was tested on the three dataset using the criterion of the registration error. And the error is calculated as the

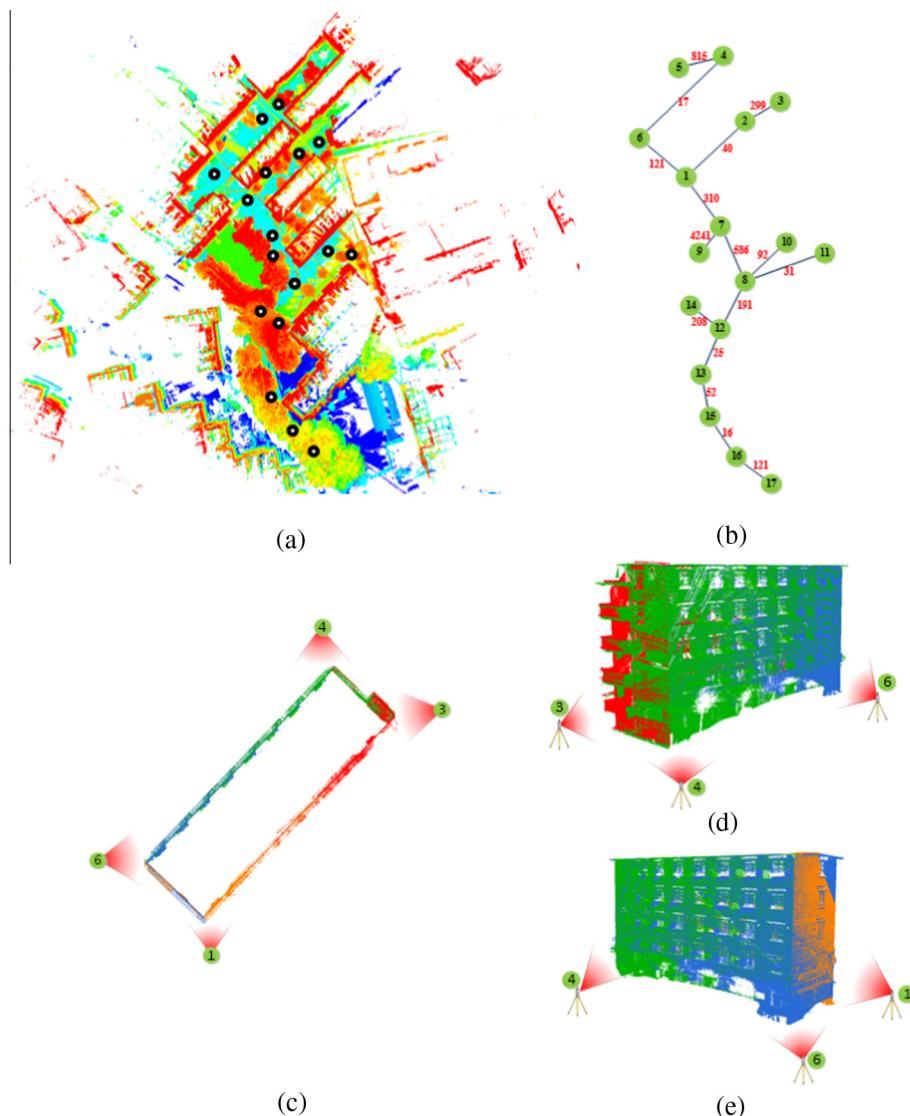


Fig. 13. Multi-view registration results for dataset B: (a) dataset B as aligned from all 18 scans, with shading representing differences in height; (b) the multi-view registration path; (c) top view of selected building; (d) left view of selected building; (e) right view of selected building.

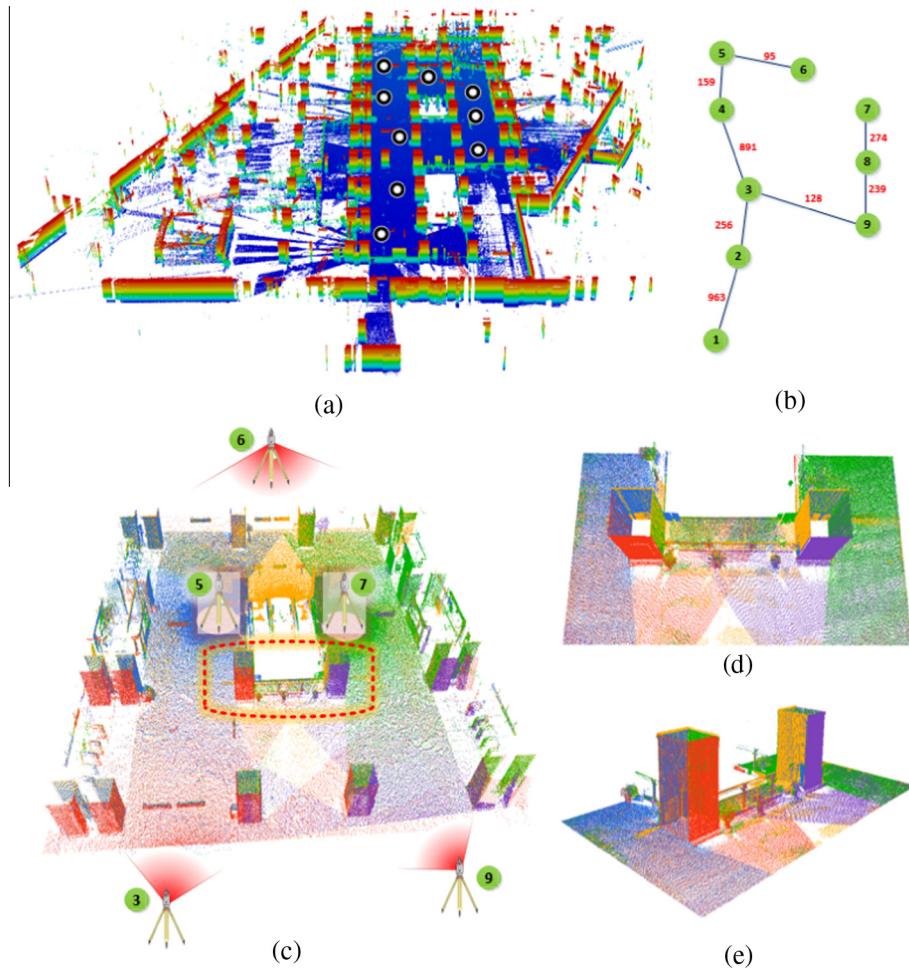


Fig. 14. Multi-view registration results for dataset C (the ceiling was removed for a better visual impression): (a) dataset C as aligned from all 18 scans, with shading representing differences in height; (b) the multi-view registration path; (c) top view of selected region; (d) top view of selected details; (e) side view of selected details.

average Euclidean distance of corresponding markers in the overlapping area. We specified U_{matching} ranging from 1.02 to 1.2 with an interval of 0.02, and $GC_{\text{constraint}}$ ranging from 0.05 m to 0.5 m with an interval of 0.05 m in this experiment, and the results were illustrated in Fig. 9. Experiments showed that the proposed method was insensitive to the configuration of the parameters values in the reasonable range with the registration errors ranging from 12.4 cm to 5.9 cm. And the registration errors reached the minimum in the vicinity of (1.1, 0.3). Therefore, parameters U_{matching} and $GC_{\text{constraint}}$ are specified as 1.1 and 0.3 m in the following experiments.

3.2.2. Semantic feature point extraction

Fig. 10 illustrates the semantic feature point extraction. Fig. 10a shows the raw point clouds colored¹ to show the elevations of the scanning points. Fig. 10b illustrates the ground and non-ground points as segmented by the method of Hernández and Marcotegui (2009), dotted in blue and yellow respectively. In light of the framework, the cross sections of the non-ground points were generated, as shown in Fig. 10c as red dots. Fig. 10d shows the segments generated by the adaptive distance determination in point segmentation, in which each segment is dotted in one color. Fig. 10e and f illustrates

the extraction results of vertical feature lines and semantic feature points.

3.2.3. Point cloud registration

Fig. 11 shows the pairwise registration result for a pair of overlapping scans. Fig. 11a illustrates the registration between the overlapping scans based on the semantic feature points extracted. Fig. 11b illustrates the registration result with different color points distinguishing the individual scans, in which the source point clouds and the target point clouds are colored purple and green respectively. Fig. 11c shows more details of the registration results. These results show that the correspondence matching and geometric consistency test between triangles are a robust way to measure the similarities between the triangles constructed from different scans, resulting in stable registration.

Figs. 12–14 show the multi-view registration results of datasets A, B, and C, respectively. Figs. 12a, 13a, and 14a show all the registered point clouds for each datasets, in which black points represent the scan positions and different colors indicate the points from different scans. Figs. 12b, 13b, and 14b show the generated registration path based on MST. The red number on each edge is the number of corresponding triangles between the two sites. Other figures show the registered results for selected regions and details of different views, with different colors standing for points from different scans.

¹ For interpretation of color in Figs. 10–14, the reader is referred to the web version of this article.

3.3. Evaluation of the proposed method

3.3.1. Time performance

Registration was performed on a computer with 8 GB RAM and an Intel (R) Core (TM) i7-4500U @ 1.80 GHz CPU. **Table 3** lists the time performance of the proposed method for registration of multiple scans, including the time costs of each step. It takes about 0.5 h to register the point cloud from 32 scans containing 282 million points. The time costs show that the proposed method has high time efficiency in aligning multiple scans, particularly for the large data volumes in urban TLS point clouds. The high time efficiency of the proposed method in aligning large-scale urban scenes is attributable to the following factors. The proposed method deals with unorganized points in the cross sections rather than the whole set of non-ground points, resulting in low computing cost for segmentation and semantic feature point extraction. Additionally, the method discards unstable triangles and indexes the remaining triangles into a hash table, thus reducing the number of triangles and speeding up retrieval of matching triangles.

3.3.2. Registration accuracy evaluation of the proposed method

To evaluate the registration accuracy of the proposed method for registering the TLS point clouds of large-scale urban scenes, the registration results for the data sets were compared with reference values based on manual alignment using artificial targets placed in the scene. **Table 4** lists the distance error (the distance between corresponding markers after registration) and transformation parameters errors in the registration. This experimental result demonstrates that the proposed method performs well in registering the TLS point clouds of complex urban scenes, with accuracy at the centimeter level, which can satisfy the requirements of object extraction and 3D reconstruction.

3.3.3. Performance comparison

A quantitative comparison was undertaken between the proposed method and that of [Dold and Brenner \(2006\)](#), and the

percentage complete as well as distance error are shown in **Table 5**. The distance error is defined as the distance between corresponding markers after registration, and the percentage complete is calculated as Eq. (9).

$$\text{Percentage complete} = \frac{N_{as}}{N_a} \quad (9)$$

where N_a is the number of total pairwise registration between adjacent scans, and N_{as} is the number of correct pairwise registration between adjacent scans.

Because the feature plane-based methods (e.g. [Dold and Brenner, 2006](#)) mainly employs the plane patches derived from buildings, the methods are difficult to deal with data sets in the suburb or open park with few buildings (e.g. data set A). Our method employs the features derived both from buildings and pole-like objects, enriching the categories of features and multiplying the number of features. Meanwhile, the proposed method adopts both the geometrical constraints (3-point scheme) and the semantic information (category and direction) for correspondence matching, reducing the probability of mismatching. All of these factors make the proposed method have better percentage complete and accuracy of registration in various urban and suburb environments than the feature plane-based method ([Dold and Brenner, 2006](#)).

3.3.4. Registration robustness analysis

It is clear that the point densities, overlapping areas, and levels of noise in the point clouds have an important impact on registration performance. To check the robustness of the proposed method, a number of experiments were undertaken on point clouds mixed with different levels of noise, varying overlapping areas, and different point densities. [Fig. 15a–c](#) illustrates the registration results achieved by the proposed method under different levels of noise, varying point densities, and different overlapping areas.

Table 3
Time performance of the proposed method.

	Semantic feature point extraction(s)		Point cloud registration(s)			Total costs (min)
	Point cloud segmentation	Vertical feature line extraction	Matching	Geometric consistency	Multi-view registration	
Data set A	376	436	150	126	130	20.3
Data set B	507	654	310	268	206	32.4
Data set C	125	201	25	18	60	7.1

Table 4
Quantitative description of registration errors.

	Transformation parameters errors						Distance error (cm)		
	Translation (cm)			Rotation angle (°)			Average	Max	RMSE
	Δx	Δy	Δz	$\Delta\phi$	$\Delta\omega$	$\Delta\kappa$			
Data set A	-5.0	2.4	1.5	0.06	0.09	0.04	6.8	12.9	2.9
Data set B	3.6	3.2	4.2	-0.02	-0.03	0.04	7.2	14.5	3.5
Data set C	-1.8	2.3	2.6	0.08	-0.04	-0.03	6.9	13.8	3.2

Table 5
Performance comparison between the proposed method and that of [Dold and Brenner \(2006\)](#).

	Dold and Brenner (2006)			The proposed method		
	Percentage complete (%)		Distance error (cm)	Percentage complete (%)		Distance error (cm)
	Ave	Max	RMSE	Ave	Max	RMSE
Data set A	83.9	12.3	25.2	4.5	100.0	6.8
Data set B	94.1	11.8	19.9	5.5	100.0	7.2
Data set C	100.0	13.1	24.5	3.9	100.0	6.9

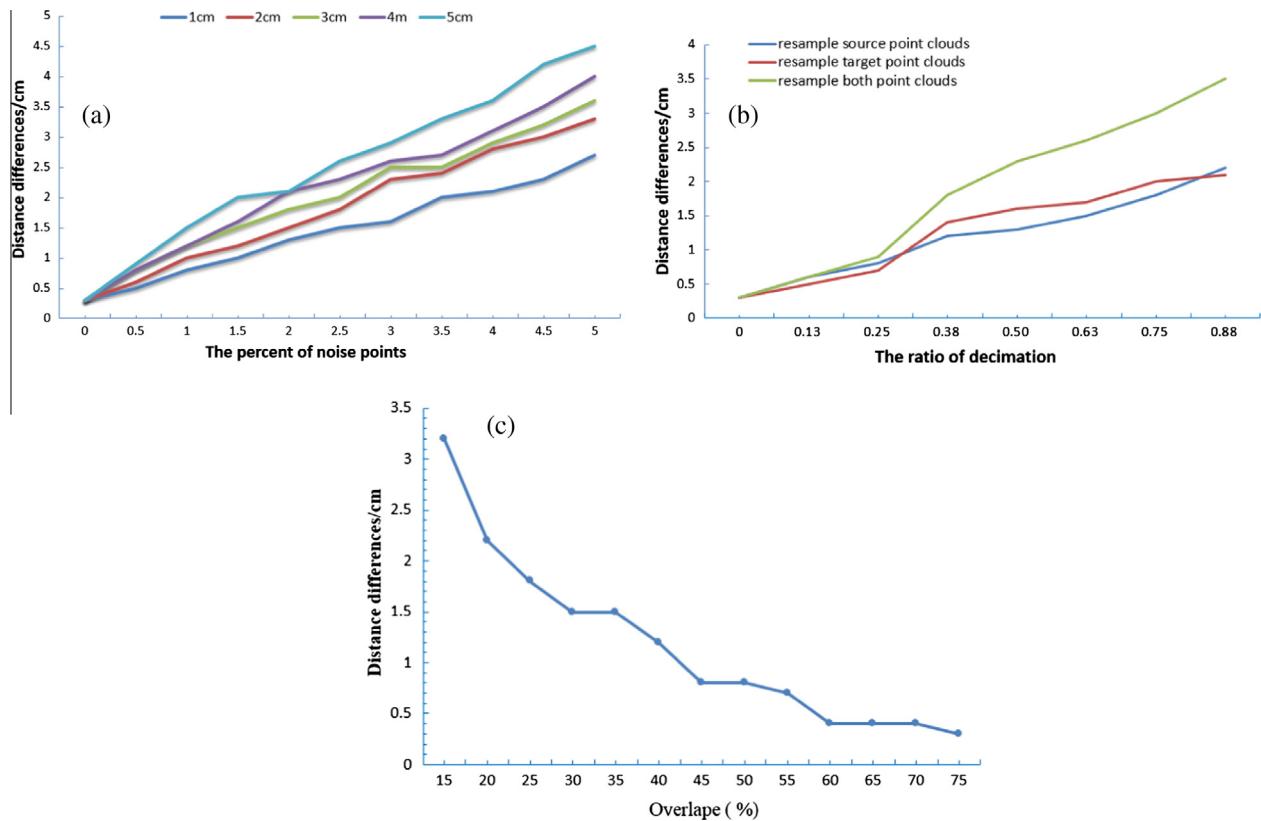


Fig. 15. Registration robustness analysis: (a) robustness to noise, (b) robustness to varying point density, and (c) robustness to overlapping.

3.3.4.1. Robustness to noise. To evaluate the robustness of the proposed method to noise, Gaussian noise with standard deviations of 1, 2, 3, 4, and 5 cm was added to the point cloud data. The distance differences of the corresponding points in the overlapping areas under different levels of noise are presented in Fig. 14a. As shown in Fig. 15a, the proposed method achieved good coarse registration accuracy, even when 5% Gaussian noise with a standard deviation of 5 cm was added to the point clouds. This indicates that the proposed method is very robust to different levels of noise.

3.3.4.2. Robustness to varying point density. To evaluate the robustness of the proposed method to varying point density, the point clouds were down sampled to 1/8, 2/8...7/8 of their original point density. And the raw laser scan points were down sampled in an organized way by keeping the regular angular measurement principle and removing a certain amount of points. The distance differences of corresponding points in the overlapping area under different levels of point density are presented in Fig. 15b. It was found that the proposed method performed well even at one-eighth of the original point density, indicating the robustness of the proposed method to varying point density.

3.3.4.3. Robustness to overlap. To test further the robustness of the proposed method to overlapping areas, a new dataset with the base length between adjacent scans ranging from 10 m to 20 m, 30 m...100 m was collected, and the overlap between them was calculated by Eq. (10). The new dataset with different degrees of overlaps are registered using the proposed method. The Euclidean distance of corresponding markers in the overlapping area under different degrees of overlap are presented in Fig. 15c. It was found that the proposed method requires few stringent overlapping conditions and works well in cases where the overlap is only 15%

between neighboring scans, thus increasing the efficiency of TLS point cloud collecting.

$$\text{overlap} = \frac{2 * \text{the corresponding point numbers in overlapping areas}}{\text{the total point numbers in both scans}} \quad (10)$$

The comprehensive comparison of the registration results under different conditions demonstrates that the proposed method is robust and reliable for registering point clouds in the urban environment. The strong robustness of the proposed method to noise, varying point density, and overlapping areas is attributable to the following factors. The proposed method adaptively determines the neighborhood size of each point, resulting in good segmentation and laying a good foundation for rapid and robust semantic feature line extraction. Additionally, the method searches for corresponding vertices by measuring the similarity between the triangles built by the vertices and generates an optimized registration path for multiple scans, leading to globally consistent registration.

4. Conclusions

Automatic and accurate registration of multiple scans is a precondition for object extraction, 3D scene analysis, and 3D model reconstruction. This paper has proposed a marker-free and multi-view registration method for large-scale urban scene point clouds. The main contribution of the proposed method are the robust algorithm for the semantic feature points extraction and the novel matching strategy by using both geometrical constraints (3-point scheme) and their semantic information (category and direction). Comprehensive experiments were undertaken to evaluate the accuracy and robustness of the proposed method. And the experiments have demonstrated that the proposed method contributes

to good registration results in terms of accuracy, time cost, and robustness to noise, point densities, and overlapping areas. However, semantic feature point extraction scheme is relatively complex, and there remains room for further improvement. And also, the future work will further investigate the approaches which provide global consistent alignments by using redundancy from multiple scans.

Acknowledgements

Work described in this paper was jointly supported by the Key NSFC project (No. 41531177), National Basic Research Program of China (No. 2012CB725301), and the NSFC project (No. 41071268).

References

- Aldoma, A., Tombari, F., Di Stefano, L., Vincze, M., 2012. A global hypotheses verification method for 3D object recognition. In: European Conference on Computer Vision, ECCV, pp. 511–524.
- Barnea, S., Filin, S., 2008. Keypoint based autonomous registration of terrestrial laser point-clouds. *ISPRS J. Photogramm. Rem. Sens.* 63 (1), 19–35.
- Besl, Paul J., McKay, Neil D., 1992. Method for registration of 3-D shapes. *Int. Soc. Opt. Photon., Robot. – DL Tentative*, 586–606.
- Böhm, J., Becker, B., 2007. Automatic marker-free registration of terrestrial laser scans using reflectance. In: Proceedings of 8th Conference on Optical 3D Measurement Techniques, Zurich, Switzerland, July 9–12, 2007, pp. 338–344.
- Dold, C., Brenner, C., 2006. Registration of terrestrial laser scanning data using planar patches and image data. *Int. Arch. Photogramm., Rem. Sens. Spatial Inform. Sci.* 36 (5), 78–83.
- García, M., Gajardo, J., Riaño, D., Zhao, K., Martin, P., Ustin, S., 2015. Canopy clumping appraisal using terrestrial and airborne laser scanning. *Rem. Sens. Environ.* 161, 78–88.
- Garnieri, A., Remondino, F., Vettore, A., 2006. Digital photogrammetry and TLS data fusion applied to Cultural Heritage 3D modeling. *Int. Arch. Photogramm., Rem. Sens. Spatial Inform. Sci.* 36 (5).
- Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., 2014. 3D object recognition in cluttered scenes with local surface features: a survey. *Pattern Anal. Mach. Intell., IEEE Trans.* 36 (11), 2270–2287.
- Habib, A., Ghanma, M., Morgan, M., Al-Rouq, R., 2005. Photogrammetric and LiDAR data registration using linear features. *Photogramm. Eng. Rem. Sens.* 71 (6), 699–707.
- Hernández, J., Marcotegui, B., 2009. Point cloud segmentation towards urban ground modeling. *IEEE, Urban Rem. Sens. Event 2009*, 1–5.
- Huber, D.F., Hebert, M., 2003. Fully automatic registration of multiple 3D data sets. *Image Vis. Comput.* 21 (7), 637–650.
- Kang, Z., Li, J., Zhang, L., Zhao, Q., Zlatanova, S., 2009. Automatic registration of terrestrial laser scanning point clouds using panoramic reflectance images. *Sensors* 9 (4), 2621–2646.
- Kankare, V., Holopainen, M., Vastaranta, M., Puttonen, E., Yu, X., Hyppä, J., Vaaja, M., Hyppä, H., Alho, P., 2013. Individual tree biomass estimation using terrestrial laser scanning. *ISPRS J. Photogramm. Rem. Sens.* 75, 64–75.
- Kruskal, J.B., 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.* 7 (1), 48–50.
- Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60 (2), 91–110.
- Montuori, A., Luzi, G., Stramondo, S., Casula, G., Bignami, C., Bonali, E., Bianchi, M.G., Crosetto, M., 2014. Combined use of ground-based systems for Cultural Heritage conservation monitoring. In: *IEEE International, Geoscience and Remote Sensing Symposium (IGARSS) 2014*, pp. 4086–4089.
- Pieraccini, M., Noferini, L., Mecatti, D., Atzeni, C., Teza, G., Galgaro, A., Zaltron, N., 2006. Integration of radar interferometry and laser scanning for remote monitoring of an urban site built on a sliding slope. *IEEE Trans. Geosci. Rem. Sens. (TGARS)* 44 (9), 2335–2342.
- Prokop, A., Panholzer, H., 2009. Assessing the capability of terrestrial laser scanning for monitoring slow moving landslides. *Nat. Hazards Earth Syst. Sci.* 9 (6), 1921–1928.
- Pu, S., Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS J. Photogramm. Rem. Sens.* 64 (6), 575–584.
- Restrepo, M.I., Ulusoy, A.O., Mundy, J.L., 2014. Evaluation of feature-based 3-d registration of probabilistic volumetric scenes. *ISPRS J. Photogramm. Rem. Sens.* 98, 1–18.
- Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm. In: *3-D Digital Imaging and Modeling, Third International Conference on, IEEE, 2001, Proceedings*, pp. 145–152.
- Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M., 2008. Aligning point cloud views using persistent feature histograms. In: *Intelligent Robots and Systems, 2008. IEEE/RSJ International Conference on, IEEE*, pp. 3384–3391.
- Salvi, J., Matabosch, C., Fofi, D., Forest, J., 2007. A review of recent range image registration methods with accuracy evaluation. *Image Vis. Comput.* 25 (5), 578–596.
- Stamos, I., Leordeanu, M., 2003. Automated feature-based range registration of urban scenes of large scale. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, IEEE, 2003, pp. II-555–II-561*.
- Theiler, P.W., Wegner, J.D., Schindler, K., 2014. Keypoint-based 4-points congruent sets-automated marker-less registration of laser scans. *ISPRS J. Photogramm. Rem. Sens.* 96, 149–163.
- Theiler, P., Schindler, K., 2012. Automatic registration of terrestrial laser scanner point clouds using natural planar surfaces. *ISPRS Ann. Photogramm. Rem. Sens. Spatial Inf. Sci.* 2012 (1), 173–178.
- Von Hansen, W., 2006. Robust automatic marker-free registration of terrestrial scan data. *Proc. Photogramm. Comput. Vision* 2006 (36), 105–110.
- Vosselman, G., Gorte, B.G., Sithole, G., Rabban, T., 2004. Recognising structure in laser scanner point clouds. *Int. Arch. Photogramm., Rem. Sens. Spatial Inform. Sci.* 46 (8), 33–38.
- Weber, T., Hänsch, R., Hellwich, O., 2015. Automatic registration of unordered point clouds acquired by Kinect sensors using an overlap heuristic. *ISPRS J. Photogramm. Rem. Sens.* 2015 (102), 96–109.
- Weinmann, M., Weinmann, M., Hinz, S., Jutzi, B., 2011. Fast and automatic image-based registration of TLS data. *ISPRS J. Photogramm. Rem. Sens.* 66 (6), S62–S70.
- Yang, B., Dong, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS J. Photogramm. Rem. Sens.* 2013 (81), 19–30.
- Yang, B., Zang, Y., 2014. Automated registration of dense terrestrial laser-scanning point clouds using curves. *ISPRS J. Photogramm. Rem. Sens.* 2014 (95), 109–121.