Article

A Robust Registration Method for Autonomous Driving Pose Estimation in Urban Dynamic Environment Using LiDAR

Rendong Wang [1], Youchun Xu [1,*], Miguel Angel Sotelo [2], Yulin Ma [3], Thompson Sarkodie-Gyan [4], Zhixiong Li [5,*] and Weihua Li [5]

[1] Department of Military Vehicles, Military Transportation University, Tianjin 300161, China; rendongwang.army@gmail.com

[2] Department of Computer Engineering, University of Alcalá, 28801 Alcalá de Henares (Madrid), Spain; miguel.sotelo@uah.es

[3] Suzhou Automotive Research Institute, Tsinghua University, Suzhou 215134, China; myl@itsc.cn

[4] Laboratory for Industrial Metrology and Automation, College of Engineering, University of Texas, El Paso, TX 79968, USA; tsarkodi@utep.edu

[5] School of Mechanical, Materials, Mechatronic and Biomedical Engineering, University of Wollongong, Wollongong, NSW 2522, Australia; weihuali@uow.edu.au

* Correspondence: youchunxu.army@gmail.com (Y.X.); zhixiong_li@uow.edu.au (Z.L.); Tel.: +1-515-203-5578 (Z.L.)

check for updates

Abstract: The registration of point clouds in urban environments faces problems such as dynamic vehicles and pedestrians, changeable road environments, and GPS inaccuracies. The state-of-the-art methodologies have usually combined the dynamic object tracking and/or static feature extraction data into a point cloud towards the solution of these problems. However, there is the occurrence of minor initial position errors due to these methodologies. In this paper, the authors propose a fast and robust registration method that exhibits no need for the detection of any dynamic and/or static objects. This proposed methodology may be able to adapt to higher initial errors. The initial steps of this methodology involved the optimization of the object segmentation under the application of a series of constraints. Based on this algorithm, a novel multi-layer nested RANSAC algorithmic framework is proposed to iteratively update the registration results. The robustness and efficiency of this algorithm is demonstrated on several high dynamic scenes of both short and long time intervals with varying initial offsets. A LiDAR odometry experiment was performed on the KITTI data set and our extracted urban data-set with a high dynamic urban road, and the average of the horizontal position errors was compared to the distance traveled that resulted in 0.45% and 0.55% respectively.

1. Introduction

Pose estimation is one of the key technologies for autonomous driving in the urban environment. Considering the fact that the GPS is unable to keep high accuracy in urban environments, a very common solution to this problem has emerged in recent years. Currently, the most common solution is based on the pose provided by GPS/INS, where real time data acquired from LiDAR or camera are registered with a priori map to achieve accurate localization [1,2]. In fact, in an environment that does not have a map, pose transform can be estimated by registration between the front and back frames in

which the vehicle's global pose is calculated by the accumulation of the previous transforms, that is odometry or the front-end of SLAM (simultaneous localization and mapping) [3,4].

The key challenges of point cloud registration in the urban environment involve two aspects: In the first instance, there are large numbers of moving objects, such as vehicles and pedestrians, and various static surroundings (e.g., the number and position of vehicles parking on the roadside vary at different times, and trees or green belts will vary greatly in different seasons) whose changes and shadows greatly increase the difficulty of registration. The second instance involves tall buildings, overpasses, and trees that may lead to multipathing and shadowing. These attributes characterize the most important factors of GPS error, whereby should the precision of INS be not high enough, will lead to the easy generation of large pose error in order to make registration hard.

ICP and its improved algorithms have very high registration accuracy, but they can easily fall into a local optimum with large noise and high initial error. Therefore, it is necessary to use coarse registration to eliminate the outliers and provide a better initial pose for accurate registration. A common coarse registration method is the RANSAC [5], which is mainly used to solve model estimation problems of large amounts of outliers in data sets. There have been many contributions towards its improvement from feature extraction and the RANSAC algorithm itself [5–7].

In this paper, the authors propose a robust solution with the generation of more efficient sample sets and employing a novel framework of the RANSAC. In Section 2 of this paper, we discuss related work and how our work is unique compared to the state of the art. In Section 3, our segmentation strategy and its merits are introduced and the principle of our registration method is analyzed and a novel framework of the RANSAC is proposed. Experimental results are shown in Section 4, and the conclusion and future work are discussed in Section 5.

2. Related Work

Up to today, there has been tremendous contribution towards the further development of pose registration based on LiDAR registration. A state-of-the-art methodology of the LiDAR odometry and SLAM has emerged as the 'LiDAR odometry and mapping in real-time' (LOAM) method [3]. This method is capable of achieving fast and accurate 3D motion estimation and works well in static and low dynamic urban scenes, but exhibits some difficulty in obtaining good results in highly dynamic scenes. In order to overcome the interference of dynamic objects, the registration is combined with detection and tracking of moving object (DATMO). Moosmann and Stiller [8] used an object tracking framework, while Yang et al. [9] used a registration framework (multiple-model RANSAC) as contribution towards this methodology. Although their methods are different, both of them integrated dynamic and static objects into one framework, and obtained good results in sequential frames. However, as time intervals between frames increase, the match of the corresponding objects becomes more difficult [10]. Furthermore, satellite positioning results are usually used as the initial pose of point cloud registration. However, satellite positioning will drift and fluctuate due to the occlusion of GPS signals from buildings and trees in cities, resulting in larger initial position and attitude errors [11]. This will make it more difficult to search and register the corresponding points in the point cloud. All of these factors will make the frame registration and object tracking worse. Particularly, this method cannot be employed in localization using priori maps.

In aspects of map-based localization, Levinson et al. [1] used the LiDAR intensity information on the road to construct high precision map in order to realize the vehicle's localization. This method can effectively overcome dynamic interference, but it is hard to adapt to rainy or snowy weather. Wolcott et al. [2] proposed a multiresolution Gaussian mixture map based on z-height information to realize the accurate localization in different weather conditions. The method may also be used in low dynamic scenes but the a priori map is required to be static. Therefore, the feature extraction of static objects is usually needed during the process of map construction and real-time registration. There are two main methods that may be considered; one method involves the application of machine learning methods for the recognition of moving and static objects [12], whilst the other involves the static object

detection by extracting features such as the vertical corner features of buildings [13], and line features of curbs [14] and road lanes [15], respectively. However, in urban areas where moving objects are crowded and without a prior map, large amounts of occlusions between objects and more sparse point clouds (e.g., using LiDAR with fewer beams or detecting objects at longer distances) will make both recognition and feature detection difficult. Although deep learning can achieve good results, it needs a lot of labeled data to train and consumes a lot of computing resources at runtime.

The above research contributions have mainly focused on the problem of dynamic interference. However, in order to achieve robust pose estimation of autonomous driving, both odometry and map-based localization may be required, and the interference of both dynamic objects and high initial pose error may be considered.

Although 4PCS [16,17] and its improved methods can overcome larger initial error, in the environment where large numbers of dynamic objects exist, it is difficult to find four pairs of corresponding static objects from two frame point clouds by random sampling, because of the fact that occlusion makes static and dynamic objects hard to distinguish, and the fact that the point clouds are sparse and have large density differences.

One method to overcome high initial pose error is usually dealt with in most localization systems by keeping the vehicle immobile (no movement) and running a particle filter to launch several potential solutions around the initial one. The algorithm is run until there is convergence. Since it uses hundreds or thousands of particles, it usually converges to a very accurate result even if the initial error is high. Upon localization convergence, the vehicle can start to move.

However, two problems exist in particle filtering. First, when the initial error is high, the convergence time to the right pose is long. In practical applications, it becomes impossible to tolerate long waiting times. Second, in changing dynamic scenes, the surroundings change all the time, and are different from the historical map data. When the particle filter is applied, there arises the risk of converging to the wrong pose. This phenomenon in urban roads is particularly prominent, such as matching to the opposite lane.

In this paper, both the dynamic interference and high initial pose error are considered. We improve the traditional methods from the object segmentation and registration. In object segmentation, three constraint conditions are adopted to optimize region growing method to get more stable object centroid. In registration, all objects are processed in a same framework irrespective as to whether they are dynamic or not. A novel framework with "RANSAC-iteration-RANSAC" three nested layers is proposed as an improvement of the RANSAC algorithm. The proposed registration method can provide higher quality solutions to any localization system, even in cases of high initial errors. Thus, particle filter-based localization methods will greatly benefit from this method both in terms of localization accuracy and computation time (i.e., less particles will be needed and convergence time will consequently decrease).

3. Proposed Method

Currently, the point cloud scanned by a LiDAR on a self-driving car is usually a sparse point cloud with large scale, and point density of each beam decreases with the increase of distance to the sensor. In addition, the distribution and density of point clouds in the same region will be different due to the change of observation pose, which further increases the difficulty of registration. Compared with local features of each scanning point, registration based on object features may be more robust. Because the number of objects is far less than the number of scanning points, the amount of calculation will be much smaller. Therefore, first of all, a complete frame of point cloud data needs to be divided into independent object point clouds.

In order to overcome the moving objects, occlusions and large initial pose errors, we improve the point cloud segmentation method to obtain more stable observation centroid of the object. A more robust coarse registration based on these centroids is achieved by improving the RANSAC algorithm.

Through rough registration, the outliers can be eliminated and the pose error can be reduced greatly. On this basis, we use ICP to accurately register the point cloud data corresponding to the interior centroids.

3.1. Object Segmentation

A point cloud scanned by a multi-beam LiDAR usually contains ground and obstacle data. In this paper, the method expressed in [18] is firstly used to remove the ground. Then, the obstacle point cloud is clustered and segmented. A simple, yet effective method is region growing. Its basic idea is to merge similar areas together, likely to DBSCAN, which can cluster point clouds of arbitrary shape. However, compared with DBSCAN, the advantage of region growing is that it uses neighborhood search does not need to build a search tree, and it does not need to search for the nearest point, so it is efficient, and is less affected by the change of point cloud density. The simplest region growing is to vote three-dimensional point clouds into two-dimensional grids containing only x and y information, and cluster them according to whether there are obstacle point clouds in the grids. The result is shown in Figure 1b, where different objects are distinguished with different colors. Obviously, when two objects close to each other, they are likely to be clustered into one object.

In this paper, we still use two-dimensional grids. The height of point cloud is regarded as an attribute of two-dimensional grid [19], and the region growing algorithm is optimized in two aspects based on this attribute:

- Unstable objects, such as low curbs and branches of trees, are eliminated directly by the threshold of both grid height and object height. The gird height is calculated by maximum height and minimum height difference of points in one grid, and the object height is calculated by maximum height and minimum height difference of grids in one object.
- Objects such as the trees, street lamps, and road signs in the environment are stable, feature-obvious, and not easily shadowed. They are important features in the registration. The height of such objects is usually very different from that of the surrounding objects, so they can be separated from other objects according to the height difference between adjacent grids.

There may be a lot of feature information about the object, but we hope to be able to complete the matching through as simple information as possible. Therefore, in this paper we use the observation centroid of the object.

The observation centroid of *i*th object (x_i, y_i) in a frame is computed by Equation (1), where n_i represents the number of points contained in the object.

$$x_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_j \text{ and } y_i = \frac{1}{n_i}\sum_{j=1}^{n_i} y_j, \tag{1}$$

Because the point cloud obtained by LiDAR through one scan is only a part of the surface of the object, to be exact, the part facing the sensor, which has an inconsistent spatial relation to the actual centroid of the object. This will lead to some new problems, for example, the observation centroid of an object may change because of the occlusion of other objects or different observation positions. In this paper, the large objects in the environment are compulsively split into multiple small objects, such as buildings, road barriers, green belts, etc. Here are three advantages: (1) The centroid of a large object is more prone to fluctuate because of occlusion generated by dynamic object, and with the movement of observation position, its centroid may also change greatly, whilst the centroid of the small object is more stable, and has less sensitivity to occlusion. (2) When a large object is divided into several small objects, it could correspond to more centroid. This will increase its proportion in the sample set. Thus, large objects will have greater weight in the registration. (3) In scenes where obvious features are deficient, such as expressways and highways, this will increase the static object ratio, which is helpful to improve the success rate and the reliability of the registration.
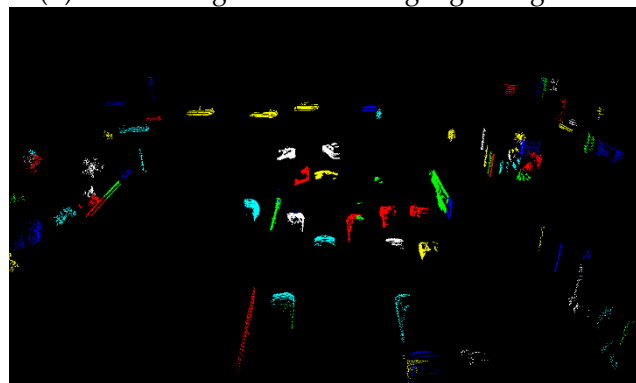
The improved segmentation results are shown in Figure 1c. After object segmentation and centroid extraction, the data quantity of the point cloud is significantly decreased. In most urban scenes, the quantity of the centroid in a frame segmented by our method is usually less than 200.
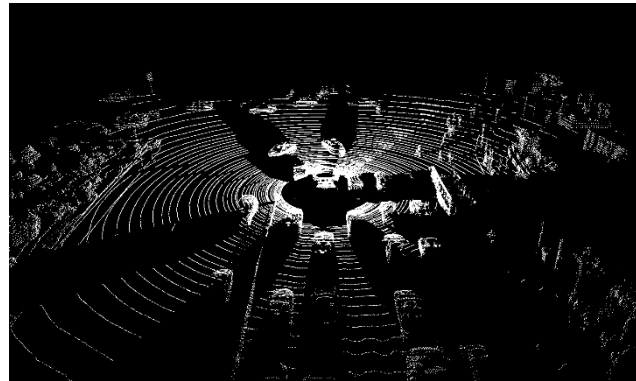
(**a**) Raw data
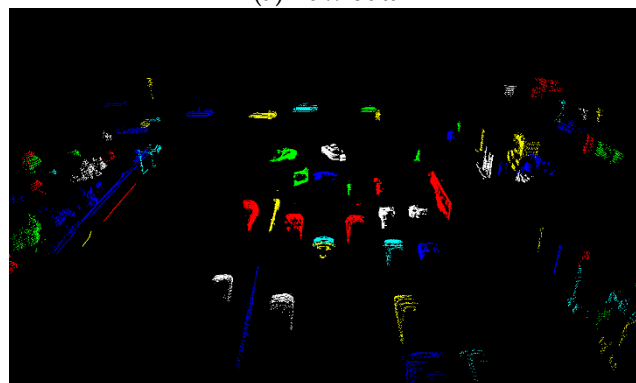
(**b**) Obstacle segmentation using regional growth

(**c**) Improved segmentation

Figure 1. Object segmentation. (**a**) Raw data. (**b**) Obstacle segmentation using regional growth. (**c**) Improved segmentation.

In summary, in order to obtain more stable observation centroids, we optimize the traditional two-dimensional region growing algorithm in three aspects:

- height information is introduced to denoise the point cloud according to the object height
- more reasonable point cloud segmentation using height difference between grids
- compulsive segmentation of large objects

3.2. Registration

In the process of registration-based localization and odometry, the vehicle's two-dimensional horizon position and heading angle are the most important, yet the most varying elements amongst the six degrees of freedom pose information. In this paper, the coarse registration of these three elements in the horizontal plane is first carried out, and then, accurate registration based on ICP is employed in inliers of 3D point clouds [20].

3.2.1. Overview

A 2D transform needs a minimum of two pairs of corresponding points. This implies that we should find two centroid points of static objects p_1, p_2 in the frame, A, and their corresponding points q_1, q_2 in the frame, B, respectively. However, it is difficult to distinguish between which objects are moving and which objects are static when the sequence of frames is weak and the initial pose is unreliable.

We use RANSAC to solve this problem, as shown in Figure 2. At first, we regarded all objects as moving. Under the consideration that both their moving speed and direction are different, the objects may be divided into several categories. (1) For static objects, the consensus of their moving status makes them classified into the same category. (2) The moving objects with different velocities and directions may be classified into different categories. Therefore, the main hypothesis of this manuscript is that even if the total number of moving objects is greater than the total number of static objects (in most situations the number of objects in the static category is still greater than the number of objects in the other categories), the transformation matrix that has the largest number of inliers found by RANSAC is the transformation matrix of the static class.
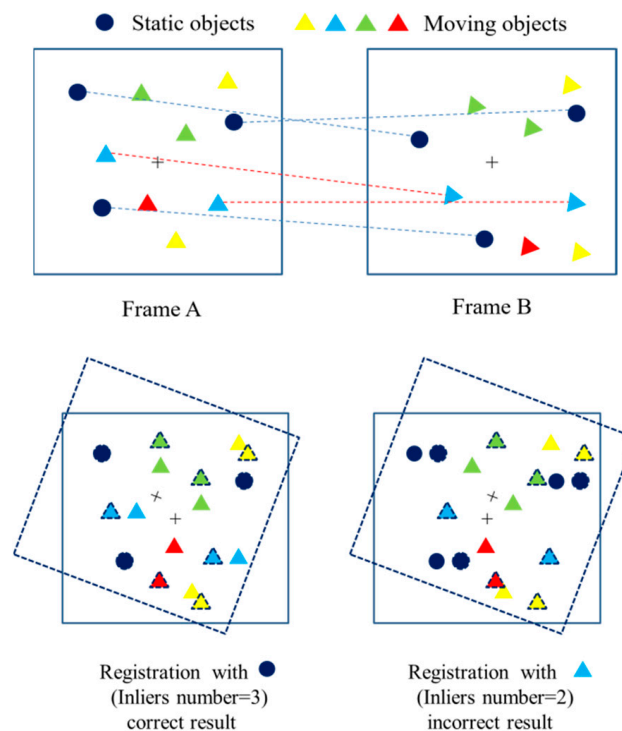

Figure 2. Schematic diagram of RANSAC registration in dynamic environment.

In this scenario, there are three static and seven dynamic objects, and the different colors represent different moving status. Although the total number of moving objects is more than that of static objects, the maximum inliers of the transformation with a set of moving objects which have the same moving status is still less than that with static objects.

3.2.2. Data Association

According to [16], the distance between two points in the same frame should be neither too close nor too far away. If the distance is too close, the error of the transformation matrix will probably increase. If the distance is too far, the corresponding points in the two frames with smaller overlapping may not be found. So, first of all, we choose the moderate two points p_1, p_2 in the frame A. In this paper, the distance between p_1 and p_2 is limited between 10 m and 40 m. Then, the K nearest neighbor points of p_1 in frame B are searched, forming a candidate corresponding points set Q_1 of p_1. Points satisfied with Equation (2) are searched to form candidate corresponding points set Q_2 of p_2.

$$Q_2 = \{q_2 | \|\|q_1 - q_2\| - \|p_1 - p_2\|\| < \varepsilon, \\ q_1 \in Q_1 \subset B, q_2 \in B, p_1 \in A, p_2 \in A\} \tag{2}$$

For each candidate pair, the transformation matrix is solved by the least squares method, and B is rotated and translated according to that. Then, the distance between each point in the transformed frame B and its nearest point in frame A is calculated, and the number of inliers whose distance is less than the threshold is counted. The candidate pair which has the greatest number of inliers is chosen as the corresponding pair.

3.2.3. Multi-Layer RANSAC

Assuming that p_A is the ratio of points that meet the transformation of frame A and B in frame A, then, a set of points (n points) is randomly selected in A, and the probability that the transformation between these points and their corresponding points is the real transformation is expressed as

$$p = p_A{}^n \tag{3}$$

After m times of the above sampling process, the probability of obtaining at least one correct result is

$$P_m = 1 - (1 - p)^m = 1 - (1 - p_A{}^n)^m \tag{4}$$

To make this probability greater than p_0, namely

$$1 - (1 - p)^m > p_0 \tag{5}$$

The number of sampling times required is

$$m > \log(1 - p_0) / \log(1 - p_A^n) \tag{6}$$

From Equation (6), it can be seen that the more the number of points needed to extract in a single random sample, the smaller the probability of getting the correct registration relationship, and the more the number of samples required. Therefore, on the premise that the model can be solved, the number of sampling points should be reduced as much as possible.

When initial pose error is small, using method in Part B is easy to get the true corresponding point, and the traditional RANSAC algorithm can register well. However, when the initial error—especially rotation error—increases, the ratio of the static points is constant; but the probability that the corresponding points exist in the set of K nearest neighbor points will significantly reduce, so the success rate of the registration using the traditional RANSAC will decrease. In order to obtain reliable registration results in the presence of high initial errors, a multi-layer nested framework of RANSAC is proposed. The framework consists of three nested layers, as shown in Algorithm 1.

The inner layer is the traditional RANSAC algorithm, where M is the sampling time in frame B. The middle layer of the algorithm is an iterative layer. The best result is selected to update the position of B, and then the RANSAC process is repeated in a new pose. It should be noticed that the aim of N iterations is to reduce the pose error rather than to filter the outliers, and instead of using the inliers,

the sample process in each iteration is performed on the whole point set of B. Therefore, the iteration only changes the pose of B. This is to avoid unsuitable filter eliminating useful points. The outer layer is a RANSAC process that nests the above two layers. It is repeated *L* times until the result satisfies the termination condition.

This framework involves two significant parameters, one is the termination condition ξ, and the other parameter is the search range *K*. ξ is the threshold of the inliers proportion which is used to determine the registration. If ξ is too large, the algorithm becomes time-consuming, while with too small ξ, the algorithm may not obtain good results. In this paper, we use the registration error and the algorithm running time to construct loss function, and take large numbers of the urban high dynamic scenes and low dynamic scenes as samples to train this parameter. The quasi Newton method is used to minimize the sample mean of the loss function, and the optimal solution $\xi = 0.3$ is obtained.

Algorithm 1 Multi-layer RANSAC

Step	Description of the implementation of the Multi-layer RANSAC algorithm
1	Start:
2	*K*: KNN search parameter
3	N_{cp}: Number of correspondences in set *A* and B_j
4	δ: Threshold of inliers ratio
5	**For** *i* = 0 **to** L **do**
6	*K* = *K*(*i*)
7	**For** *j* = 0 **to** N **do**
8	*M* = *M*(*j*)
9	**For** *k* = 0 **to** M **do**
10	RANSAC(*A*, B_j)
11	**End for**
12	Calculating T_j with the largest number of inliers
13	$T = T \cdot T_j$
14	$R_j = \max(n_k)/N_{cp}$
15	**If** $R_j > \delta$ **then**
16	**Return** *T*
17	**Else**
18	$B_{j+1} = T_j (B_j)$
19	**End if**
20	**End for**
21	**End for**
22	**RANSAC**(*A*, B_j):
23	Random sample p_i in *A*
24	Search correspondences of p_i in B_j by KNN
25	Pick the points that satisfy conditions
26	Calculate matrix of each corresponding pairs
27	Transform B_j and count the number of inliers
28	End.

K is related to the initial pose error of the registration and should be increased as the initial error increases. However, it is difficult to estimate the range of the initial error in practical applications just as the value of *K*. If *K* is too big, both the time consumption of the algorithm and the risk of the registration failure will increase. If *K* is too small, the correct registration results cannot be obtained when the initial error is high. In order to solve this problem, we dynamically adjust *K* by increasing it in the outer layer and decreasing it in the middle layer, respectively. The initial value of the middle layer is provided by the outer layer. Using this strategy, our algorithm can automatically adapt to various scenes and various initial errors in less time than the method of changing *K* artificially.

The variables in the algorithm are defined as follows.

S_A is the set of the static objects in set A; S_B is the set of the static objects in set B; a_i is any point in set A; b_i is any point in set B; B_k represents the set B after k-th iterations; $b_i^{(k)}$ is any point in B_k; in set B, q_i is the actual corresponding point of p_i in set A; $q_i^{(k)}$ is the candidate corresponding point in set B_k, Q is the set of candidates corresponding points; $\widetilde{q}_i^{(k)}$ is the optimum corresponding point that computed by Equations (7) and (8),

$$
n_j^{(k)} = \begin{cases} 1, & \|a_j - b_j^{(k)}\| < \varepsilon \\ 0, & else \end{cases}
\tag{7}
$$

$$
\widetilde{q}_i^{(k)} = \underset{q_i^{(k)}}{\mathrm{argmax}} \left\{ \sum_{a_j \in A} n_j^{(k)} \right\}
\tag{8}
$$

where $e^{(0)}$ is the initial pose error and $e^{(k)}$ is the registration error after kth iteration of the middle layer.

Compared with the traditional RANSAC algorithm, with the same number of samples, the probability of the improved algorithm to achieve the correct registration results is significantly improved, and the proof process is as follows:

According to Bayes formula,

$$
\begin{aligned}
& P(\widetilde{q}_i^{(k)} = q_i | e^{(k)}) \\
& = \frac{P(q_i \in Q | e^{(k)}) \cdot P(\widetilde{q}_i^{(k)} = q_i | q_i \in Q, e^{(k)})}{P(q_i \in Q | \widetilde{q}_i^{(k)} = q_i, e^{(k)})}
\end{aligned}
\tag{9}
$$

Because all the points in set Q will be iterated through to search the optimum corresponding point,

$$
\begin{aligned}
& P(\widetilde{q}_i^{(k)} = q_i | q_i \in Q, e^{(k)}) \\
& = P(q_i \in Q | \widetilde{q}_i^{(k)} = q_i, e^{(k)}) = 1
\end{aligned}
\tag{10}
$$

When the error $e^{(k)}$ increases, especially when the rotation error is higher, the probability of the existence of the corresponding points q_i in the KNN set Q will greatly decrease, so will the value of $P(\widetilde{q}_i^{(k)} = q_i | e^{(k)})$. Therefore, $e^{(k)}$ should be reduced as much as possible to increase the likelihood of finding a corresponding point, which can be achieved through iteration.

In the traditional RANSAC algorithm, the correct registration results can be obtained only when p_i is the static point and the optimal corresponding point $\widetilde{q}_i^{(k)}$ is real corresponding point q_i. Its probability is represented by P_0. However, in the iterative process of this method, we just need the transformation matrix calculated by p_i and $\widetilde{q}_i^{(k)}$ to reduce the initial error, i.e., $e^{(k)} < e^{(0)}$. In the first iteration,

$$
\begin{aligned}
P_1 &= P(e^{(1)} < e^{(0)} | e^{(0)}) \geq P(\widetilde{q}_i^{(k)} = q_i | e^{(0)}) \\
&> P(\widetilde{q}_i^{(k)} = q_i | e^{(0)}) \cdot P(p_i \in S_A | A) = P_0
\end{aligned}
\tag{11}
$$

From Equation (11), we can see that P_1 is bigger than P_0, especially in the environment with more moving objects. The advantage of this method is obvious because the value of $P(p_i \in S_A | A)$ is small.

According to Equation (6), the sampling times m_k in the kth iteration should be satisfied with Equation (12).

$$
P(e^{(k)} < e^{(0)} | m_k) = 1 - (1 - P_k)^{m_k} > P(p_i \in S_A | A)
\tag{12}
$$

And in the next iteration,

$$
\begin{aligned}
P_k &= P(e^{(k)} < e^{(k-1)}|e^{(k-1)}) \\
&> P(e^{(k)} < e^{(k-1)}, e^{(k-1)} < e^{(0)}|e^{(k-1)}) \\
&= P(e^{(k)} < e^{(k-1)}|e^{(k-1)}, e^{(k-1)} < e^{(0)}) \cdot P(e^{(k-1)} < e^{(0)}|m_{k-1}) \\
&\geq P(\widetilde{q}_i^{(k)} = q_i|e^{(k-1)}, e^{(k-1)} < e^{(0)}) \cdot P(e^{(k-1)} < e^{(0)}|m_{k-1}) \\
&> P(\widetilde{q}_i^{(k)} = q_i|e^{(0)}) \cdot P(e^{(k-1)} < e^{(0)}|m_{k-1}) \\
&> P(\widetilde{q}_i^{(k)} = q_i|e^{(0)}) \cdot P(p_i \in S_A|A) = P_0
\end{aligned}
\tag{13}
$$

Therefore, after kth iterations, the probability of registering success is

$$
P_s = 1 - \prod_{i=1}^{k}(1 - P_i)^{m_i} > 1 - (1 - P_0)^{\sum\limits_{i=1}^{k} m_i}
\tag{14}
$$

From the above deduction, it may be observed that within a certain range, the more moving objects, the greater the initial error; and the more iterations, the more obvious the advantages of this method than traditional methods.

The outer layer of our algorithm is a RANSAC process regarded as the middle layer as a unit. This layer can effectively improve the registration success rate when the initial error is high by dynamically adjusting the parameters. Although many iterations may be required during this process, the algorithm is still efficient in most cases due to the small amount of points used for registration.

3.2.4. Accurate Registration

The set B is transformed according to the coarse registration matrix T, and the data in the centroid set A and B are divided into inliers and outliers according to the threshold distance of the corresponding points. Then, the segmented point cloud in both P_A and P_B corresponding to the inliers of the centroid set is extracted to consist of P'_A and P'_B, respectively. Finally, weighted ICP is implemented for accurate 3D registration of point cloud in P'_A and P'_B with the initial transformation T. Different from the coarse registration, the small and high objects, which are always pole objects such as trees, street lamps, and road signs, are greatly weighted in accurate registrations. Because they are less changeable and the distribution of their point clouds is more concentrated, weighted ICP is more accurate than standard ICP, especially in the registration between the current frame and the historical frame for localization.

4. Experiments

The experiment includes two parts. Part A is the registration on typical urban dynamic scenes with various initial pose errors. It includes registration between sequence frames which is the basis of LiDAR odometry, and the registration between the current frame and the historical frame which is the basis of localization with an a priori map. Part B is the LiDAR odometry experiment in the public dataset provided by KITTI [21] and a data set of urban highly dynamic environments collected by ourselves. For more experimental result, please see the Supplementary video.

Our LiDAR data is collected by velodyne HDL-64E. During the experiments, the algorithms processing the LiDAR data run on a computer with 2.4 GHz quad cores and 8 G memory, on top of the ubuntu14.04 in Linux. The algorithm consumes only one thread.

4.1. Registration

This experiment consists of two groups: registration between sequence frames (short time interval, 0.5 s) and registration between current frame and historical frame (long time interval, three months, winter and spring). Experiments were carried out in large numbers of urban scenes. For each scene, the actual distance between two frames is about 5 m.

The position and heading of the vehicle provided by the RTK-GPS are regarded as the real coordinates, expressed in the form (x, y, θ). An offset $(\Delta x, \Delta y, \Delta \theta)$ is added to the coordinates of one frame as the initial pose of registration. By adjusting the offset distance $|\Delta d|$ ($\sqrt{\Delta x^2 + \Delta y^2}$, units:

meter) and offset angles $|\Delta\theta|$ (units: degree), we set four sets of initial errors: The ranges of distance offset are [0, 4], [6, 10], [14, 18], [24, 28]; and the ranges of the angle offsets are [0, 5], [5, 10], [0, 15], [15, 20]. They are denoted as (4 m, 5°), (10 m, 10°), (18 m, 15°), and (28 m, 20°), respectively.

To demonstrate the effectiveness of our work, both segmentation and registration algorithms are compared before and after our improvement. The segmentation is represented as seg0 and seg1, and the registration is represented as RANSAC0 and RANSAC1. They can form four methods, i.e., seg0 + RANSAC0, seg0 + RANSAC1, seg1 + RANSAC0, seg1 + RANSAC1 (proposed). For each scene and each set of initial error, each method is implemented 100 experiments. The initial offset in each experiment is randomly generated within the range of initial error. All the experiments in this paper use the same parameters.

Here, we selected four typical highly urban dynamic scenes in each group for the analysis. Figures 3–10 show the scene registration results at the initial error (28 m, 20°) using the proposed method. In Figures 3–10, (a) is the top view of two obstacle point clouds with an initial offset (they are respectively shown as red and green); (b) is the top view after segmentation and registration (points belong to inliers are shown as blue and most of them are points of overlapping static objects, with moving objects and non-overlapping static objects eliminated effectively); (c) is a local enlarged 3D view of (b), which shows that the static objects, such as buildings, trees and parked vehicles, are registered well.

The four scenes for registration between sequence frames are denoted as scene A1, A2, A3, and A4. A1 is a crossroad with the experimental car turning right; A2 is a crossroad with the experimental car turning left; A3 is a straight narrow road with numbers of static objects occluded by a bus; and A4 is a traffic congestion road with the experimental car changing lane.

The four scenes for registration between current frame and historical frame are denoted as scene B1, B2, B3, and B4. B1 is a straight urban road, with many parked cars whose position has changed greatly; B2 is a crossroad, and our experimental car collected data of these two frames in opposite lanes and directions; B3 is a ramp entrance, in which there are amounts of vegetation changing greatly from winter to spring, and fences existing in winter but not in spring; and B4 is an elevated road lacking geometric features.

Both groups of scenes have numbers of moving objects, such as vehicles and pedestrians.

The registration effectiveness is determined by the deviation between the registration pose and its true pose. However, in practical applications, the true pose of a frame is unknown. In this paper, the ratio of the inliers obtained by the RANSAC is used as an evaluating indicator for the success of the registration. When the ratio of the inliers is greater than the threshold ξ, the coarse registration is considered to be successful.
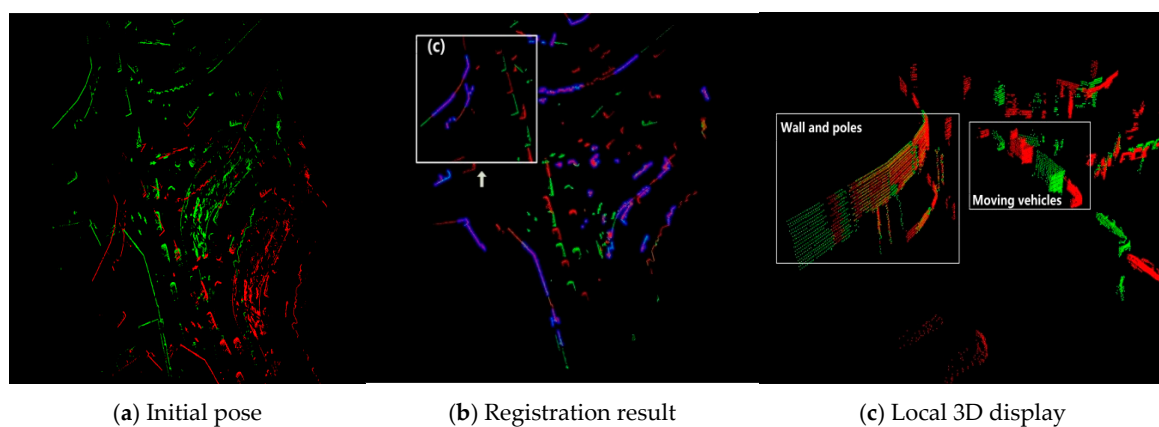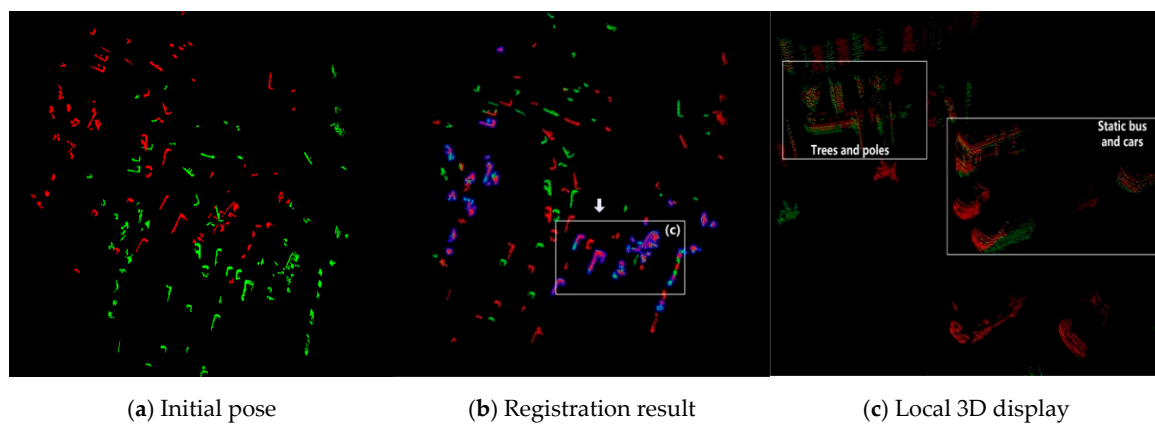

(**a**) Initial pose   (**b**) Registration result   (**c**) Local 3D display

Figure 3. Scene A1.

(**a**) Initial pose　　　　(**b**) Registration result　　　　(**c**) Local 3D display

Figure 4. Scene A2.

(**a**) Initial pose　　　　(**b**) Registration result　　　　(**c**) Local 3D display

Figure 5. Scene A3.

(**a**) Initial pose　　　　(**b**) Registration result　　　　(**c**) Local 3D display
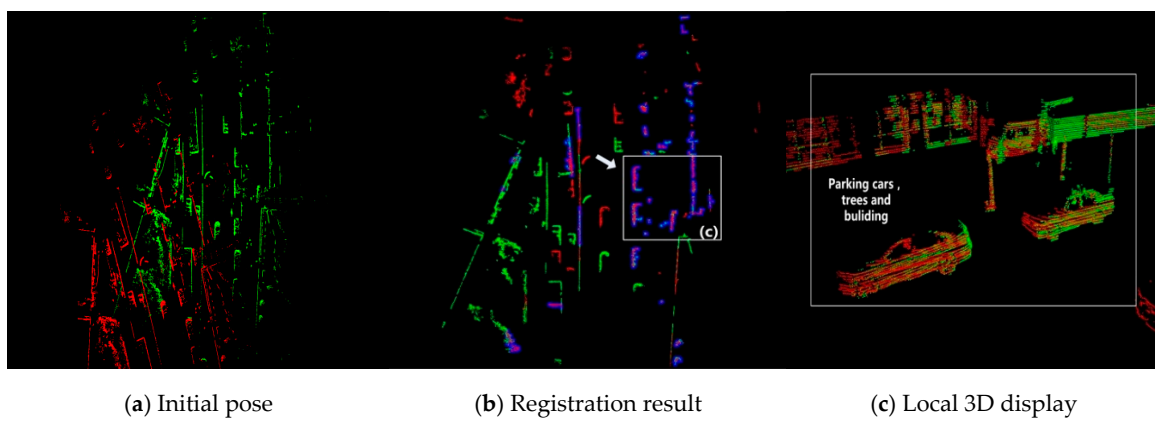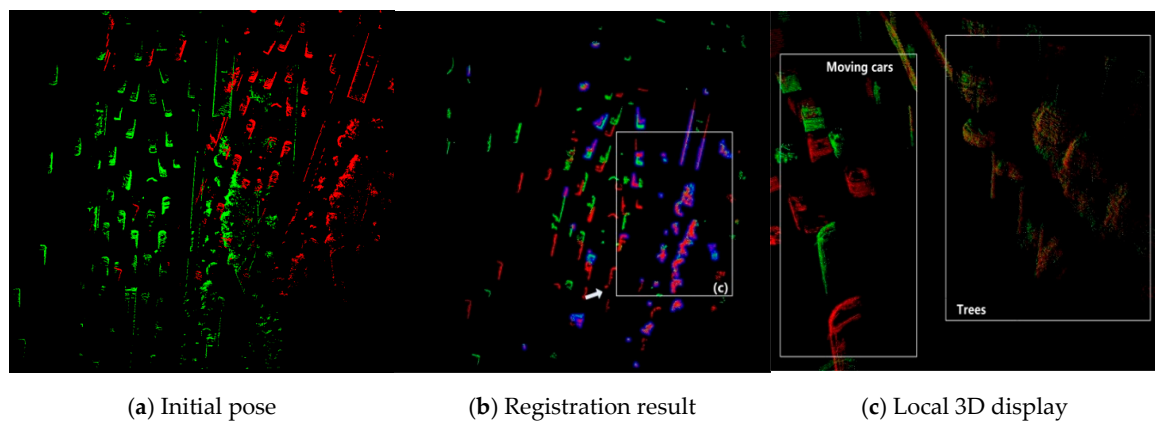
Figure 6. Scene A4.
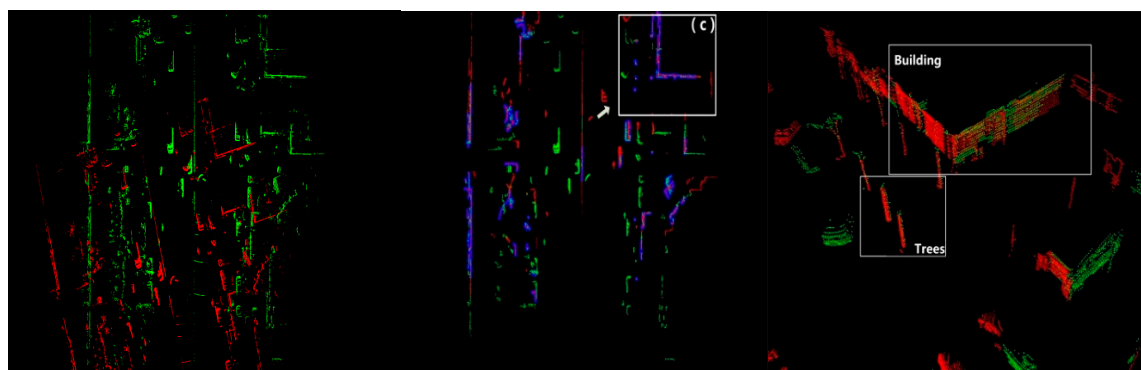
(**a**) Initial pose   (**b**) Registration result   (**c**) Local 3D display

Figure 7. Scene B1.
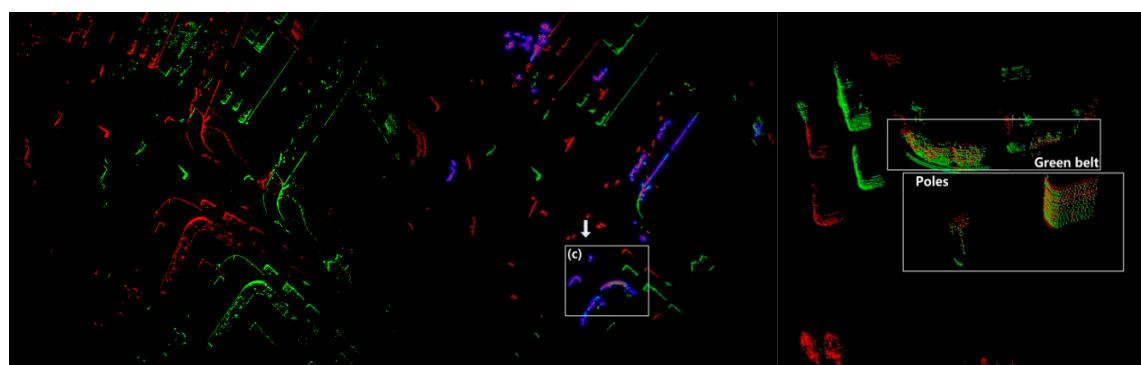
(**a**) Initial pose   (**b**) Registration result   (**c**) Local 3D display

Figure 8. Scene B2.
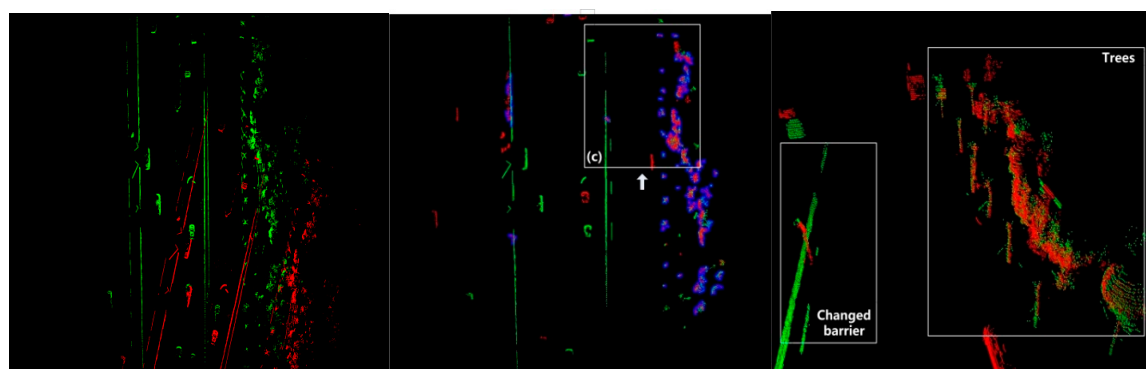
(**a**) Initial pose   (**b**) Registration result   (**c**) Local 3D display

Figure 9. Scene B3.

(**a**) Initial pose   (**b**) Registration result   (**c**) Local 3D display
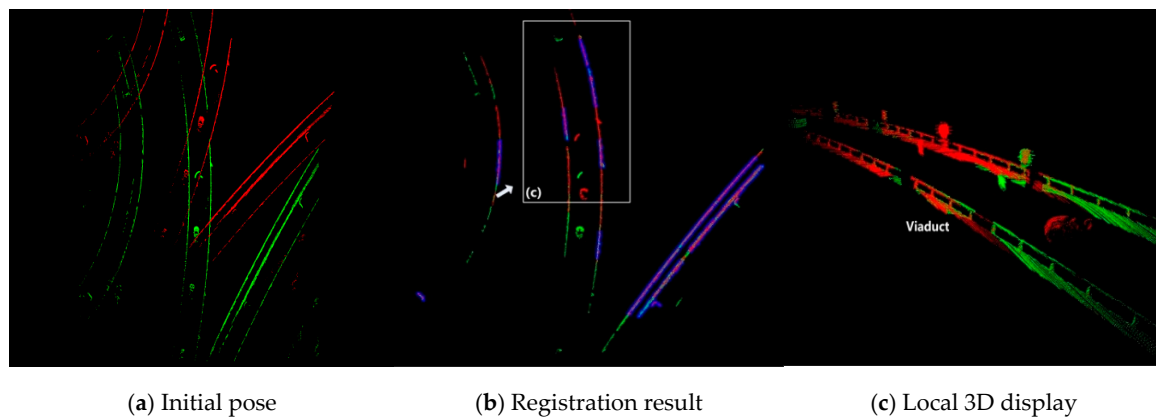
Figure 10. Scene B4.

In order to verify the rationale of the evaluation method, the 12,800 experimental results of both groups are statistically analyzed, and their error distributions are demonstrated in Figure 11. The *x*-axis is the distance error between the registration result and the real position, whereas the *y*-axis is the heading error. In order to represent the distribution of the small error data more clearly, a logarithmic coordinate system is applied. Even scenes, initial offsets and registration algorithms are all different, the error of the registrations recognized as success are obviously smaller than the errors of registrations recognized as failure. Table 1 indicates the relationship between the error and the classification of the registration result in all tests. It may be observed that the majority of the registrations judged as success have errors less than (0.2 m, 0.2 m, 0.5°), and the majority of registrations judged as failures have errors more than (0.2 m, 0.2 m, 0.5°), respectively. There are only a few results that are misjudged. This includes results whose registration error is high whilst the algorithm considers it as a successful registration. In fact, most of these misjudgments occur in the experiments of the high initial error of the scene B4. This is because the static features in this scene are single, and the number of features is very few, which results in the algorithm considering the wrong matching as the correct result, which leads to the big error of our pose estimation. This is also a shortcoming of our method.
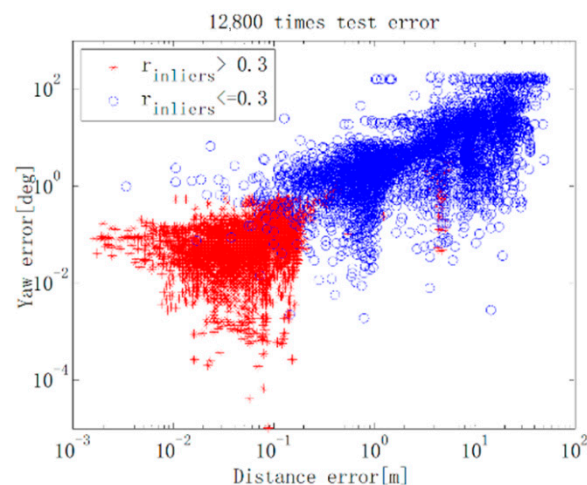

Figure 11. Error distribution and classification of registration results.

Table 1. Statistics of experiment times with result in different error and classification.

	Test A (Short-Term Registration)		Test B (Long-Term Registration)	
	Success	**Fail**	**Success**	**Fail**
Error less than (0.2 m, 0.2 m, 0.5°)	4126	40	3561	17
Error more than (0.2 m, 0.2 m, 0.5°)	29	2205	451	2371

Therefore, we make a statistical comparison of the success rate of the algorithm before and after improvement, as shown in Figure 12. The improved algorithm has a very high success rate in different scenes and different initial errors, which is greatly better than the traditional method. Even in the case of initial offset (28 m, 20°), the success rate of our method in the two sets of experiments can achieve 94% and 98%.
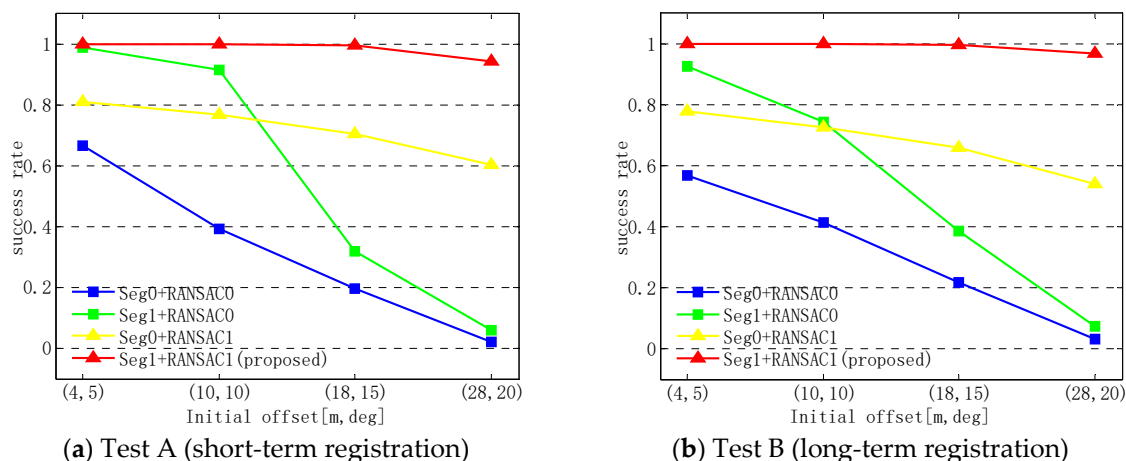

(**a**) Test A (short-term registration) (**b**) Test B (long-term registration)

Figure 12. Comparison of algorithm registration success rates in different scenes.

In addition, the registration errors of each method are also analyzed. Figure 13 shows the proportion of the experiments whose registration errors (lateral, longitudinal, yaw) are less than (0.1 m, 0.1 m, 0.25°) and (0.2 m, 0.2 m, 0.5°) in the two set of experiments with different initial offsets. It can be seen that the proportion of the precise results obtained by the proposed method is higher than that of the traditional methods. In both groups A and B, the proportion of the number of registration errors within (0.2 m, 0.2 m, 0.5°) is kept above 95% by using the proposed method. In the A group, the ratio of the resulting error within (0.1 m, 0.1 m, 0.25°) remained at about 90%. In the B group, due to some changes in the static scene, the proportion of registration errors within (0.1 m, 0.1 m, 0.25°) is much lower, but still more than 70%.
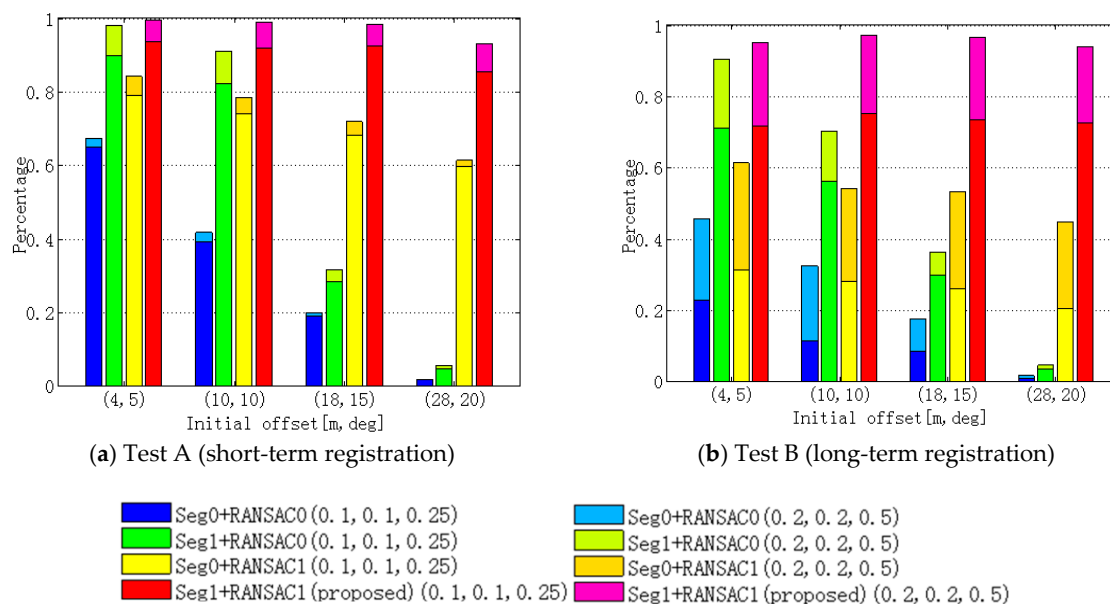

(**a**) Test A (short-term registration) (**b**) Test B (long-term registration)

Figure 13. The proportion of high-precision registration results.

According to both Figures 12 and 13, comparing method 1(seg0 + ransac0) and method 2(seg1 + ransac0), it may be observed that the new segmentation method can effectively improve the registration quality of the complex dynamic scene with small initial error. However, as initial error increases, the registration effects of the two methods decline greatly. The comparison between method 1 (seg0 + ransac0) and method 3 (seg0 + ransac1) shows that the improved RANSAC method can work well with greater initial error. As initial error increases, the registration effect of method 3 has only decreased slightly while method 1 decreased greatly. Therefore, using both the improved segmentation and the improved RANSAC, the robustness of the registration is greatly improved in high dynamic environments with high initial error, which can be seen from the contrast between method 1 (seg0 + ransac0) and method 4 (seg1 + ransac1).

Two sets of experiments show that the algorithm can achieve accurate and stable registration results, whether it is short interval registration or long interval registration. The RMSE of the successful registration in different scenes are shown in Table 2. It may be seen that the location accuracy of our algorithm has reached the centimeter level, and the RMSE of the heading angle is within 0.3°. In the scenes with rich features such as crossroads, the angular error of the algorithm is smaller, and it can almost be controlled within 0.1°.

Figure 14 shows the comparison of time consumption of coarse registration between the proposed method and the traditional RANSAC method with different initial offsets. When the offset is small, the average time consumption of our method can be kept at about 80 ms, which is obviously superior to the traditional RANSAC method. With the increase of the initial offset, the average time-consumption of the proposed method increases, but it is still less than 150 ms. In the experiment of the traditional RANSAC method, with more sampling process becoming unavailable due to the failure of corresponding search points, more registration steps are skipped, which results in the decrease of time-consuming as the initial offset increasing.

Table 2. Registration error of our method in different scenes

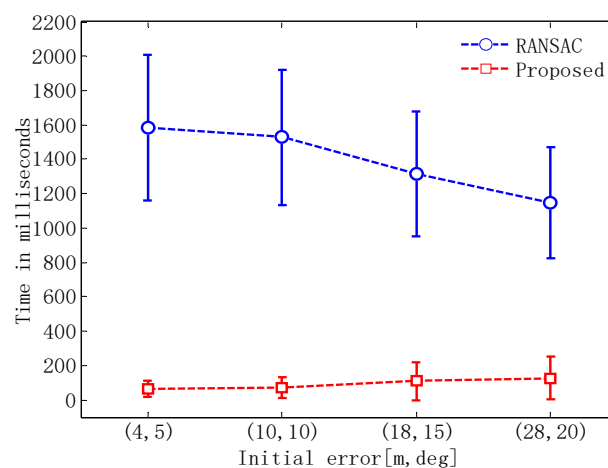| Scene	RMSE		
	Lateral (m)	Longitudinal (m)	Yaw(°)
A1 turning right	0.0148	0.0354	0.0954
A2 turning left	0.0298	0.0348	0.1113
A3 bus occlusion	0.0193	0.0753	0.2322
A4 traffic jam	0.0291	0.0249	0.0419
B1 straight road	0.076	0.1561	0.2751
B2 cross road	0.0342	0.044	0.0409
B3 entrance ramp	0.0546	0.0642	0.2328
B4 elevated road	0.0221	0.0373	0.047

Figure 14. Time consumption of coarse registration.
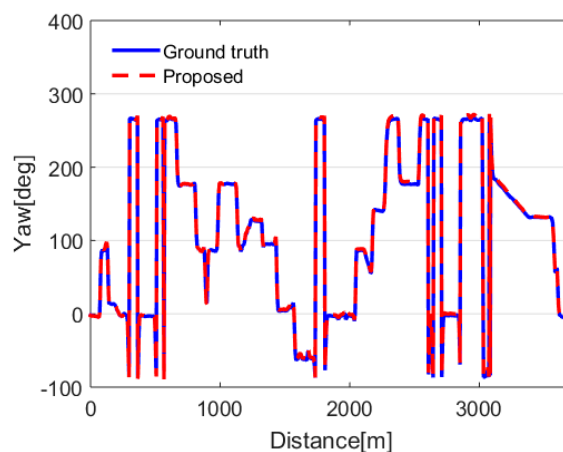
4.2. LiDAR Odometry

LiDAR odometry based on point cloud registration is one of the key technologies of autonomous driving. It is used to further demonstrate the accuracy and stability of our method.

In this experiment, the starting point is given by GPS, and then the following poses are all calculated by point cloud registration, without any assistance from other sensors and algorithm of smoothing and optimization. To achieve good ego-motion results, each frame for registration is extracted from every five frames on both KITTI dataset and our own dataset. Because the acquisition frequency of LiDAR is 10 Hz, the data acquisition time of the two frames for the registration is 500 ms, which is much larger than the time consumption of the registration calculation, so that the real-time performance of the algorithm can be ensured.

KITTI provides urban public data sets and error evaluation methods that are used specifically for LiDAR odometry testing. The advantages of this method are mainly in the complex urban roads that exist in large numbers of moving objects, but unfortunately, the urban scene for the odometry test in this dataset is mainly static, with only a few moving cars in it. However, to verify the adaptability of the algorithm to different environments, we still do experiments in this data set. Figure 15 is the experimental result of the data set numbered 00, with a total length 3.7 km. Figure 15a shows the location error of the odometer based on the registration method, and the Figure 15b shows the error of the heading angle. Some large fluctuations in Figure 15b are caused by the critical value of the angle just at 360 and 0 degrees.
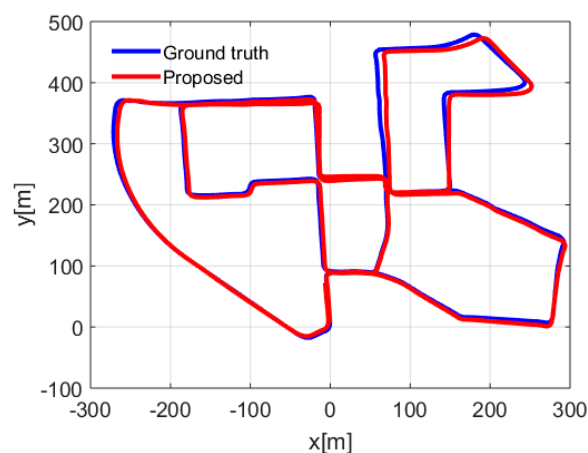
(**a**) Localization result.

(**b**) Yaw result.

Figure 15. LiDAR odometry result in KITTI dataset.

In addition, experiments were conducted on our urban dataset with large numbers of moving objects in Tianjin. It is shown in Figure 16a, with the full length of 1.6 km. We captured several typical pictures taken by vehicle mounted cameras, including a crossroad, a S curved road, a road with numbers of buses (the bus runs in the opposite lane, causing almost all the static data on this side to be occluded) and a very urgent U-turn road. There are a large number of vehicles and pedestrians in the whole course of the experiment. In this experiment, our method is compared with the MM-RANSAC [9], and the experimental results are shown in Figure 16b,c. Because the frame sequence is weak, the MM-RANSAC algorithm has high errors at many positions. Especially at turning points, such as the S turn and U turn in our experimental dataset, great changes of the heading angle in a short time result in registration failures. Thus, the error of the heading angle calculated by MM-RANSAC based registration is high, causing serious errors in the odometry. However, because the proposed method does not rely on the sequence between the frames and can work well in the case of high initial error, the odometry using our registration method remained stable and accurate, whether on straight road or turning road.

According to the evaluation method provided by KITTI, the average horizontal position error of the LiDAR odometry based on our registration method is 0.45% on the KITTI training dataset sequence 00 with low dynamic scene and 0.55% on our high dynamic scene dataset. We have submitted our horizontal results to KITTI for test.

At present, high precision map has been rapidly developed and has become a necessary module for automatic driving, but it is still difficult to ensure real-time updating. In some sections with no map or map updates, we can use the LiDAR odometry based on the proposed method to achieve reliable localization, even if there are lots of dynamic obstacles in the scene.

It should be mentioned that although our experimental data are all from the LiDAR with 64 beams, our algorithm is not only limited to this LiDAR, but also applies to the LiDAR with 16 beams, 32 beams, and so on.

The experiment shows that our method has strong robustness and can be applied to many complex urban road scenes. The LiDAR odometry based on our method can help to get rid of the dependence on high precision inertial navigation system to reduce cost. In addition, as we can register real-time data with the historical data long before, the requirement for the real-time performance of high precision maps can be reduced.

However, in some cases, the LiDAR localization based on this method is still likely to fail. For example, in a tunnel with single surrounding features, the LiDAR localization is easy to produce a high longitudinal error. Because our method can work in the case of high error, when a vehicle travels from a feature deficient scene to a feature of rich scene, although the pose error may accumulate due to localization failure, the proposed method can still quickly converge to the correct location.
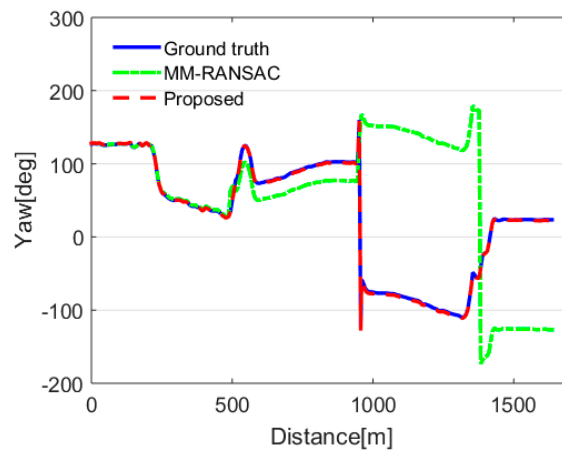
More results can be found in the Supplementary video attached to this paper in the web-site of journal of *Electronics*.
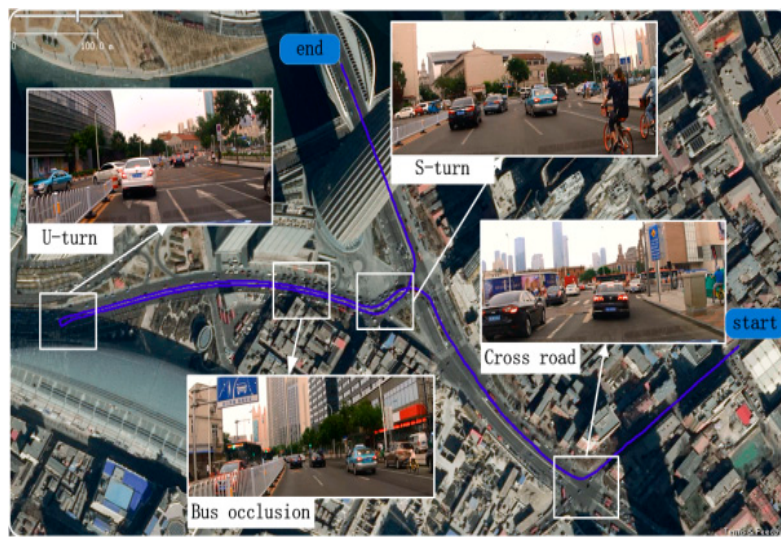
(**a**) Overview of test road
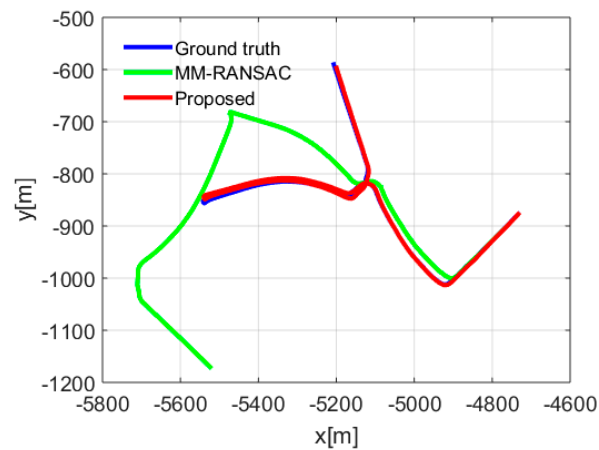
(**b**) Localization result

(**c**) Yaw result

Figure 16. LiDAR odometry result in highly dynamic environment.

5. Conclusions

Aiming at point cloud registration in urban complex dynamic environment for autonomous driving, this paper proposed a more robust registration method by optimizing the segmentation and improving the RANSAC algorithm with a novel framework, named 'multi-layer RANSAC'.

Experimental results demonstrate the robustness of our algorithm. It can achieve fast and accurate registrations in high dynamic scenes with large numbers of moving objects, serious occlusions and static environmental changes; even in case of high initial pose error. The key contributions of our paper are:

(1) The algorithm can solve the urban scene registration with numbers of moving objects without the aid of any other techniques such as object tracking and detection. Therefore, it is fit for both long and short time interval registrations. It can be used not only for LiDAR odometry, but also for precise localization with a priori map.

(2) The algorithm can adapt to higher initial pose error, so it can solve some difficult problems in localization, such as the poor initial localization accuracy caused by poor GPS signal at the vehicle start position and the high accumulated error due to long distance ego-motion estimation.

Our method still has a small amount of over-segmentation and under-segmentation, and our registration is more likely to be wrong due to misjudgment in scenes with sparse features. Next, we will try to extract semantic features from scenes for better segmentation, and set various object weights for better registration. In addition, in view of the fact that our method can provide the initial segmentation of dynamic and static objects, we will try to apply our method to the static map construction in dynamic environments and dynamic object detection and tracking

Supplementary Materials: The research results presented in this paper can be found in the supplementary video at http://www.mdpi.com/2079-9292/8/1/43/s1, Video S1: results.

Author Contributions: R.W. and Y.X. presented the algorithm; R.W., Y.M., and Z.L. wrote the paper; M.A.S. and R.W. provided analysis tool; R.W. and Y.X. analyzed the data; T.S.-G. and W.L. reviewed and revised the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Levinson, J.S. Automatic Laser Calibration, Mapping, and Localization for Autonomous Vehicles. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2011.

2. Wolcott, R.; Eustice, R. Fast LIDAR Localization using Multiresolution Gaussian Mixture Maps. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; IEEE: New York, NY, USA, 2015; pp. 2814–2821.

3. Zhang, J.; Singh, S. Low-drift and real-time lidar odometry and mapping. *Auton. Robots* **2017**, *41*, 401–416. [CrossRef]

4. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]

5. Chum, O.; Matas, J.; Kittler, J. Locally Optimized RANSAC. In *Proceedings of Joint Pattern Recognition Symposium*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 236–243.

6. Sattler, T.; Leibe, B.; Kobbelt, L. SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter. In Proceedings of the 2009 IEEE International Conference on Computer vision, Kyoto, Japan, 29 September–2 October 2009; IEEE: New York, NY, USA, 2009; pp. 2090–2097.

7. Pankaj, D.; Nidamanuri, R. A robust estimation technique for 3D point cloud registration. *Image Anal. Stereol.* **2016**, *35*, 15–28. [CrossRef]

8. Moosmann, F.; Stiller, C. Joint Self-Localization and Tracking of Generic Objects in 3D Range Data. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: New York, NY, USA, 2013; pp. 1146–1152.

9.   Yang, S.; Wang, C. Multiple-model RANSAC for Ego-Motion Estimation in Highly Dynamic Environments. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 18 August 2009; IEEE: New York, NY, USA, 2009; pp. 3531–3538.

10.   Wang, C.; Thorpe, C.; Thrun, S.; Hebert, M.; Durrant-Whyte, H. Simultaneous localization, mapping and moving object tracking. *Int. J. Robot. Res.* **2007**, *26*, 889–916. [CrossRef]

11.   Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]

12.   Schlichting, A.; Brenner, C. Vehicle localization by lidar point correlation improved by change detection. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 703–710. [CrossRef]

13.   Im, J.; Im, S.; Jee, G. Vertical corner feature based precise vehicle localization using 3D LIDAR in urban area. *Sensors* **2016**, *16*, 1268. [CrossRef] [PubMed]

14.   Hata, A.; Wolf, D. Road Marking Detection using LIDAR Reflective Intensity Data and its Application to Vehicle Localization. In Proceedings of the 2014 IEEE International Conference on Intelligent Transportation Systems, Qingdao, China, 8–11 October 2014; IEEE: New York, NY, USA, 2014; pp. 584–589.

15.   Hata, A.; Osorio, F.; Wolf, D. Robust Curb Detection and Vehicle Localization in Urban Environments. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 8–11 June 2014; IEEE: New York, NY, USA, 2014; pp. 1257–1262.

16.   Aiger, D.; Mitra, N.; Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. *ACM Trans. Gr.* **2008**, *27*, 85. [CrossRef]

17.   Mellado, N.; Aiger, D.; Mitra, N. Super 4pcs fast global point cloud registration via smart indexing. *Comput. Gr. Forum* **2014**, *33*, 205–215. [CrossRef]

18.   Su, Z.; Xu, Y.; Peng, Y. Enhanced detection method for structured road edge based on point clouds density. *Automot. Eng.* **2017**, *39*, 833–838.

19.   Dimitrievski, M.; Van Hamme, D.; Veelaert, P.; Philips, W. Robust matching of occupancy maps for odometry in autonomous vehicles. In Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016), Rome, Italy, 27–29 February 2016; Volume 3, pp. 626–633.

20.   Anderson, S.; Barfoot, T.D. RANSAC for Motion-Distorted 3D Visual Sensors. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; IEEE: New York, NY, USA, 2013; pp. 2093–2099.

21.   Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]