IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 29, 2019, accepted May 28, 2019, date of publication June 7, 2019, date of current version June 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921676

GP-ICP: Ground Plane ICP for Mobile Robots

HYUNGJIN KIM, (Member, IEEE), SEUNGWON SONG, AND HYUN MYUNG, (Senior Member, IEEE)

Urban Robotics Lab, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Hyun Myung (hmyung@kaist.ac.kr)

This work was supported in part by the Industrial Convergence Core Technology Development Program (Development of Robot Intelligence Technology for Mobility with Learning Capability Toward Robust and Seamless Indoor and Outdoor Autonomous Navigation) funded by the Ministry of Trade, Industry and Energy (MOTIE), South Korea, under Grant 10063172, in part by the Industry Core Technology Development Project (Development of Artificial Intelligence Robot Autonomous Navigation Technology for Agile Movement in Crowded Space) funded by the MOTIE, South Korea, under Grant 20005062, and in part by BK21+.

ABSTRACT In this paper, we propose a robust point cloud registration method for ground vehicles. Given the vast developments in the field of autonomous vehicles, the use of point cloud data has increased. The simultaneous localization and mapping (SLAM) algorithm is typically used to generate sophisticated point cloud maps. In the SLAM algorithm, the quality of the map depends on the performance of loop closure algorithms. The iterative closest point (ICP) algorithm is widely used for loop closure of the point cloud. However, the ICP algorithm might not work well for ground vehicles because it was originally developed for 3D reconstruction in computer vision field. Therefore, this paper proposes a method to find robust matching correspondences in the ICP algorithm on ground vehicle conditions. The performance of the proposed method is compared with other conventional methods by using KITTI open datasets. The source code is publicly released on the Github website.

INDEX TERMS Iterative closest point, point cloud, autonomous vehicle, ground plane condition.

I. INTRODUCTION

Thanks to the tremendous improvements brought about by light detection and ranging (LiDAR) sensor technology over the last decades, LiDAR sensors have been utilized in various robotics applications, including simultaneous localization and mapping (SLAM) [1], localization [2]–[4], map generation [5], object and pedestrian detection [6], etc. The DARPA Grand/Urban Challenge [7] verified the possibility of fully autonomous driving using LiDAR sensors. Although LiDAR sensors can obtain high-precision information from the environment, not all information can be procured because of the limitations of the scanning range. Therefore, data registration is required for localization and map generation.

The scan matching algorithm is widely utilized for the registration of 3D shapes. In the early 1990s, the iterative closest point (ICP) technique was first proposed for scan matching [8], [9]. The ICP algorithm iteratively calculates the transformation between two 3D shapes by finding correspondence pairs at each step. Besl and McKay [8] proposed a ''point-to-point'' ICP algorithm, which searches for the correspondence pairs from the geometrically closest point in the 3D shape. Chen and Medioni [9] introduced a ''point-to-plane'' ICP algorithm that can be used for range data by estimating the target model as a plane. Zhang [10] introduced a robust ICP algorithm that addresses outliers, occlusion appearance, and disappearance.

After the ICP algorithm was proposed, many alternative studies have been undertaken. Lu and Milios [11] proposed iterative dual correspondence (IDC) to improve robot pose estimation in unknown environments by matching 2D range scans. The metric-based ICP (MbICP) [12] algorithm was also devised to handle a large initial orientation error by minimizing geometric distance including the translation and orientation of sensors simultaneously. Unlike the ICP algorithm, the normal distributions transform (NDT) algorithm [13] was proposed as a new approach to laser scan matching without the need for explicit correspondences. This approach optimizes the transformation using the normal distribution of each cell from the probabilities of measuring a point. Some researches proposed a 3D registration method based on Fast Point Feature Histograms (FPFH) [14], [15]. These methods have advantages for global registration, but the 3D model should be of high density to extract good features. Since these

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

traditional methods try to match dense point cloud models, it is difficult to apply them directly to 3D LiDARs, with which autonomous vehicles are mostly equipped.

The data provided by 3D LiDAR sensors, also known as multi-layer LiDAR, is sparse due to the limitations of the sensor coverage area. Therefore, other researchers have focused on studying matching with multi-layer LiDAR sensor data. The generalized ICP (G-ICP) method [16] and multi-channel G-ICP [17] take advantage of sparse data matching by considering covariance of each point in the rigid transformation. Tazir *et al*. [18] proposed cluster iterative closest point (CICP) for sparse-dense point cloud registration. Agamennoni *et al*. [19] introduced probabilistic data association for data registration. The 3D [20] and multi-layered NDT [21] algorithms were also devised to match 3D multi-layer sensors. Das *et al*. [22] and Das and Waslander [23] introduced 3D scan registration using the NDT with ground segmentation. Similarly, Pandey *et al*. [24] introduced a 3D point cloud alignment for a ground plane-dominant environment. Zaganidis *et al*. proposed the semantic-assisted Normal Distributions Transform (SE-NDT) by considering semantic category during optimization [25]. These algorithms consider only the characteristic of the sensor system or environment, but do not take into account the movement of the sensor system or vehicle.

Most studies reported improved performance by analyzing data in various ways. However, these methods are not well-suited to autonomous vehicles as the original purpose of the data registration algorithms is to match two 3D shapes for 3D reconstruction. Therefore, this paper proposes a robust scan matching algorithm with a ground constraint. The ground vehicles can only move on the ground plane that lies along the horizontal direction. As the relative transformation of 3D LiDAR data is mainly related to the ground plane direction, the transformations of $\mathbf{SE}(2)$ elements (x, y, yaw) are large, while other elements (z, roll, pitch) are small. Thus, we suggest an efficient data association method to find the best correspondences in the ground constraint.

The main contribution of this paper is its introduction of an efficient searching method to find the best correspondences for ground vehicle applications. In ICP iteration, the general search algorithms depend on the geometric distance without any constraints. Thus, this paper proposes a ground plane search algorithm. Since the ground plane usually lies on the x-y plane, the proposed algorithm finds the correspondences through a boundary search based on height information (z). Unlike the 2D ICP algorithm, 6 degree-of-freedom (DoF) transformation is estimated at every iteration step to overcome ground slope conditions.

II. GROUND PLANE ICP

A. G-ICP

The main concept of the standard ICP algorithm consists of two steps. First, the matched correspondences are computed using geometric distances from two scan data. Second, the transformation between the two scan data is calculated by minimizing the distance of matched correspondences. To obtain the final transformation, these two steps are repeated until convergence. Some researchers focus on the first step by computing better correspondences than the standard [11], [12]. Others target the second step of data processing while calculating the transformation [16]. In this paper, we focus on the first step to improve the matching qualities for the ground condition.

Algorithm 1 Standard ICP

Input: $A = \{a_i\}$: target point cloud
$B = \{b_i\}$: query point cloud
T_0: initial transformation
Output: T: the aligned transformation with A and B

1 $T \leftarrow T_0$
2 $K^A \leftarrow k\text{-dTreeGeneration}(A)$
3 **while** *not converged* **do**
4 **for** $i \rightarrow 1$ **to** N **do**
5 $m_i \leftarrow \texttt{SearchClosestPoint}\left(K^A, T \cdot b_i\right)$
6 **end**
7 $T \leftarrow \underset{T}{\operatorname{argmin}} \sum_i \|T \cdot b_i - m_i\|^2$
8 **end**

Algorithm 1 represents the standard "point-to-point" ICP. The inputs are two point clouds, A and B, and an initial transformation T_0. The output is the final transformation between these two point clouds, T. In the first step of the ICP algorithm, each matched correspondence, m_i, is computed by finding the closest pair depending on the current transformation, T (line 5), using the k-d tree search algorithm that is generated in advance in line 2. In the second step, a current transformation is computed by minimizing the distance between the matched correspondences (line 7). For robustness, outliers can be rejected by checking the distance between m_i and $T \cdot b_i$. The point-to-plane ICP is implemented by simply changing line 7 of Algorithm 1 as follows:

$$T \leftarrow \underset{T}{\operatorname{argmin}} \sum_i \|\eta_i \cdot (T \cdot b_i - m_i)\|^2 \tag{1}$$

where η_i denotes the surface normal vector of m_i.

This paper focuses on scan matching utilized in ground vehicles. G-ICP [16] often yields acceptable results to match multi-layer LiDAR sensor data [26], but it does not consider a ground condition. G-ICP takes advantage of sparse data by taking into account the tendency of data during the optimization process. In the G-ICP algorithm, the optimization step is changed to the probabilistic-based model on line 7 in Algorithm 1. When the matched correspondence point sets of two scan data are denoted as $A = \{a_i\}_{i=1,\ldots,N}$ and $B = \{b_i\}_{i=1,\ldots,N}$, the probabilistic model of point sets can be represented by $\hat{A} = \{\hat{a}_i\}$ and $\hat{B} = \{\hat{b}_i\}$ where $a_i \sim \mathcal{N}(\hat{a}_i, C_i^A)$ and $b_i \sim \mathcal{N}(\hat{b}_i, C_i^B)$, respectively, $\hat{a}_i$ and $\hat{b}_i$ are means, and C_i^A, C_i^B are covariance matrices calculated using the distribution of the surrounding points. If $\mathbf{T}^*$ is the aligned transformation,

we can write

$$\hat{b}_i = \mathbf{T}^* \hat{a}_i. \tag{2}$$

Let the distance error of an arbitrary rigid transformation be defined as $d_i^{(\mathbf{T})} = b_i - \mathbf{T}a_i$. Then, $d_i^{(\mathbf{T}^*)}$ is drawn as follows:

$$\begin{aligned} d_i^{(\mathbf{T}^*)} &\sim \mathcal{N}(\hat{b}_i - \mathbf{T}^* \hat{a}_i, C_i^B + (\mathbf{T}^*) C_i^A (\mathbf{T}^*)^T) \\ &= \mathcal{N}(0, C_i^B + (\mathbf{T}^*) C_i^A (\mathbf{T}^*)^T) \end{aligned} \tag{3}$$

by applying $\hat{b}_i - \mathbf{T}^* \hat{a}_i = 0$ from (2). To compute $\mathbf{T}^*$, the simplified optimization method is performed as follows:

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_i d_i^{(\mathbf{T})^T} (C_i^B + \mathbf{T} C_i^A \mathbf{T}^T)^{-1} d_i^{(\mathbf{T})}. \tag{4}$$

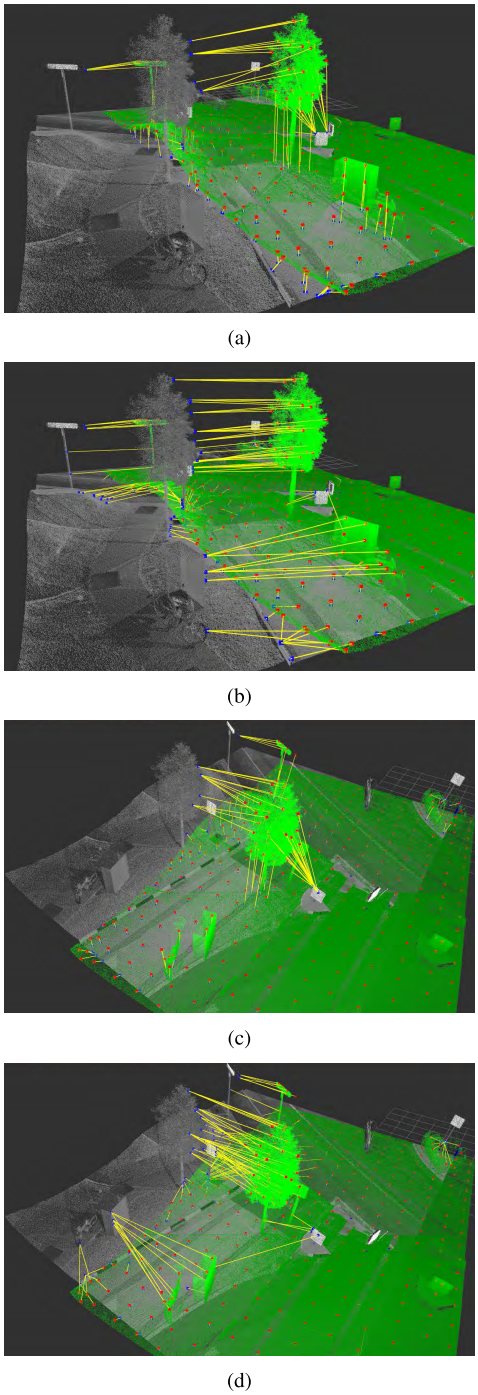
The details are shown in [16]. Our proposed method also calculates the transformation of the correspondence pairs using the G-ICP method.

B. GP-ICP

This paper proposes an efficient search method to find the best correspondences for ground vehicles. The proposed method changes the search step based on line 5 of Algorithm 1. The conventional method finds correspondences based on the closest point in the 2D or 3D geometric distance. Most ego-motions of ground vehicles have slight changes in roll, pitch, and height direction (z), and large changes in forward direction (x), lateral direction (y), and yaw. In the conventional method, the correspondence points are found according to the Euclidean distance in x, y, and z-axes without considering the ground condition, so the mismatch might happen. For ground vehicles, some conditions never occur such as a large displacement in the z-axis. Therefore, this paper proposes a height-limit conditioned search algorithm by changing line 5 of Algorithm 1 as follows:

$$\begin{aligned} m_i \leftarrow \texttt{SearchClosest} \\ \texttt{Point}\left(K^A, T \cdot b_i\right) |_{|(m_i)_z - (T \cdot b_i)_z| \le \epsilon}, \end{aligned} \tag{5}$$

where ϵ denotes a pre-specified small value and the subscript z denotes the height value of the defined vector. When searching for a correspondence as in (5), the mismatch pair is minimized by the z-axis condition. Examples of the proposed and conventional matching methods are shown in Fig. 1. Fig. 1 represents matching correspondence results of the proposed method and conventional method between a query and a target point cloud at a random initial transformation of a certain distance. According to the result in Fig. 1, the proposed method has the advantage of matching vertical information to a better extent than the conventional method as the matching condition is limited to a certain height range. Since the proposed method can also estimate 6-DoF transformation due to a limited height range, it can handle a inclined condition unlike 2D ICP.



(a) (b) (c) (d)

FIGURE 1. Comparison of the proposed and conventional matching methods between a query point cloud (green) and a target point cloud (gray). The yellow lines denote the result of matching correspondences. (a) Matching result of the conventional method in view #1. (b) Matching result of the proposed method in view #1. (c) Matching result of the conventional method in view #2. (d) Matching result of the proposed matching method in view #2.

C. ACCELERATION OF GP-ICP

This section introduces an acceleration method of the proposed GP-ICP algorithm. The conventional ICP algorithm finds the correspondence point by referring to the nearest

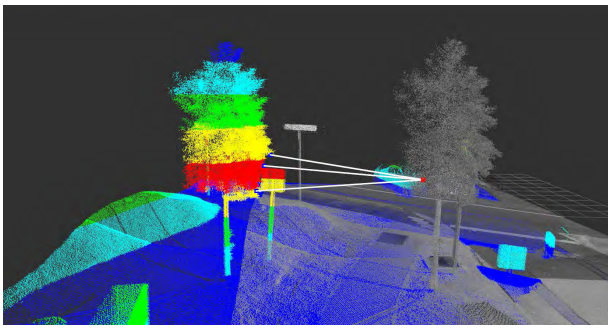


FIGURE 2. Acceleration concept to find matching correspondences using the multi-layer scheme. The colored layers denote each multi-layer point cloud, and the white lines refer to the results of each matching correspondence with each multi-layer point cloud.

neighbor point using the k-d tree algorithm. On the other hand, the proposed method searches for the correspondence point by checking the height value sequentially until the point is in a certain height range. In this case, the complexity increases if many neighbor points are out of the height range. The average complexity of the conventional k-d tree search algorithm is $O(\log(n))$, where n is the number of points in the search space. On the other hand, the average complexity of the proposed searching method is $O(p \cdot \log(n))$, where p denotes the number of search trials until neighbor points are within the height range. In usual cases, 80% $\sim$ 90% of the nearest points satisfy the height range (in this case $p = 1$), while others do not satisfy the height range. In the unsatisfied case, p is often very large. Inspired from this, we propose an acceleration method to search for (5) using multi-layer scheme. The target point cloud is divided evenly in vertical direction by a pre-specified small value, ϵ from (5), as shown in the colored point cloud layers in Fig. 2. The query point cloud searches the closest point in each layer according to the height value, and then, the correspondences are found from the closest point in the height range among the searched points. Since the k-d tree search algorithm is performed three times, i.e., for the relevant layer, its upper layer, and its lower layer, the average complexity of the proposed search algorithm is $O(3 \cdot \log(n/L))$, where L denotes the total number of layers. L is determined by ϵ and the maximum height of point cloud. This height-limited search algorithm is performed only if the height condition is not satisfied ($p \neq 1$). While the computation time of non-accelerated search algorithm depends on the p value, the accelerated search algorithm has a constant computation time. The accelerated method is faster than the non-accelerated method when the p value is greater than 3 even in the worst case of $L = 1$. Since p is often very large in unsatisfied cases, the accelerated method is expected to be much faster than the non-accelerated method.

The accelerated GP-ICP algorithm is shown in Alg. 2. The multi-layer k-d trees, $K^A_{1,\dots,h}$, where h denotes the number of layers, are generated by a certain height interval ϵ (line 3). The line 7 checks the hight condition. In line 8 of

Algorithm 2 Accelerated GP-ICP

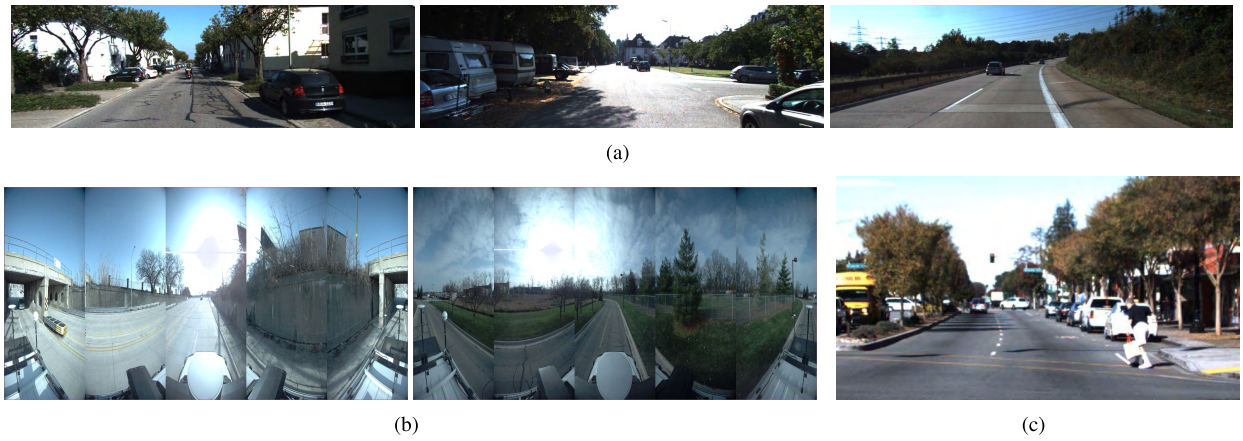
Input: $A = \{a_i\}$: target point cloud
$B = \{b_i\}$: query point cloud
T_0: initial transformation
Output: T: aligned transformation with A and B

1 $T \leftarrow T_0$
2 $K^A \leftarrow k\text{-dTreeGeneration}(A)$
3 $K^A_{1,\dots,h} \leftarrow k\text{-dTreeGeneration}(A_{1,\dots,n})$
4 **while** *not converged* **do**
5 **for** $i \rightarrow 1$ **to** N **do**
6 $m_i \leftarrow \texttt{SearchClosestPoint}\left(K^A, T \cdot b_i\right)$
7 **if** $|(m_i)_z - (T \cdot b_i)_z| > \epsilon$ **then**
8 $l \leftarrow \texttt{SearchLayerByZ}\,(T \cdot b_i)$
9 $m_i^+ \leftarrow \texttt{SearchClosestPoint}\left(K^A_{l+1}, T \cdot b_i\right)$
10 $m_i^0 \leftarrow \texttt{SearchClosestPoint}\left(K^A_{l}, T \cdot b_i\right)$
11 $m_i^- \leftarrow \texttt{SearchClosestPoint}\left(K^A_{l-1}, T \cdot b_i\right)$
12 $m_i \leftarrow \texttt{MinDistance}\left(T \cdot b_i, \{m_i^+, m_i^0, m_i^-\}\right) |_{|(m_i^*)_z - (T \cdot b_i)_z| \leq \epsilon}$
13 **end**
14 **end**
15 $T \leftarrow \underset{T}{\operatorname{argmin}} \sum_i \|T \cdot b_i - m_i\|^2$
16 **end**

Algorithm 2, the relevant layer, l, of each point is selected by the z value of $T \cdot b_i$ in every iteration step. The closest points in the relevant layer (line 9), its upper layer (line 10), and its lower layer (line 11) are found by the k-d tree search of the corresponding layer. The final matched point is selected using the closest point among the three points (m_i^+, m_i^0, and m_i^-) under the height condition $|(m_i^*)_z - (T \cdot b_i)_z| \leq \epsilon$. The same optimization method to G-ICP is applied in line 15.

III. EXPERIMENTS

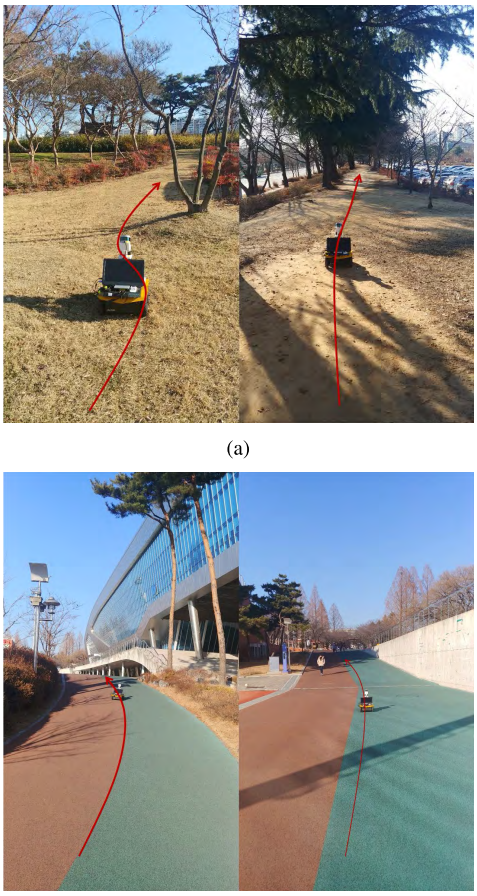
Since the proposed method focuses on matching with sparse data for a ground vehicle, the algorithm is verified using a KITTI [27], [28] (sequences 00 and 01), Ford campus [29] collected on a vehicle equipped with Velodyne 64E [31], and Udacity [30] datasets with Velodyne 32E. These datasets were collected in various urban, residential, and campus areas as shown in Fig. 3. Ford campus data partially contains off-road environment and Udacity dataset contains many inclined environment. To test extreme cases such as off-road and inclined environments, our own robot system equipped with Velodyne VLP-16 is utilized as shown in Fig. 4. In this experiment, only the horizontal Velodyne sensor is used for matching. The off-road environment consists of glass and dirt road as shown in Fig. 5a and the inclined environment slants around 5° $\sim$ 10° as shown in Fig. 5b. To demonstrate superiority, GP-ICP is compared with conventional methods



(a) (b) (c)

FIGURE 3. Various test environments from open dataset. (a) KITTI, (b) Ford Campus, (c) Udacity.

FIGURE 4. The robot system used in our experiment.



(a) (b)

FIGURE 5. Experimental environments of our own datasets. (a) Off-road environment and robot path. (b) Inclined environment and robot path.

such as the G-ICP [16], point-to-point ICP [8], point-to-plane ICP [9], and NDT [13] algorithms.

The experiment is conducted by matching two point clouds collected at different positions. To compare the matching performance of each algorithm, 2,646, 1,039, 1,256, and 1,503 pairs from sequence 00 and 01 of KITTI, Ford campus, and Udacity datasets, respectively, are randomly selected by two point clouds of Velodyne with a distance difference of 7-10m. We also select 1,000 pairs each from our own off-load and inclined datasets are also randomly selected by two point clouds of Velodyne VLP-16 with a distance difference of 2-4m, respectively. Since the overlapping part of Velodyne VLP-16 is smaller than the one of Velodyne 64E, the distance difference of Velodyne VLP-16 datasets set smaller than the one of Velodyne 64E. Since the proposed method has advantages for robust matching in ground conditions, each pair is tested to match in various initial transformations with x (-8 m $\sim$ 8 m), y (-8 m $\sim$ 8 m), and theta ($-40° \sim 40°$) where the direction of the x-axis and y-axis denote the front direction and the lateral direction of the vehicle, respectively. To represent the performance of each algorithm, the success rate in each direction with the absolute initial transformation is plotted in Fig. 6. The success or failure is determined by the overlapping area of the matched point clouds. The overlapping points are judged by the distance between matching correspondences which are selected as the

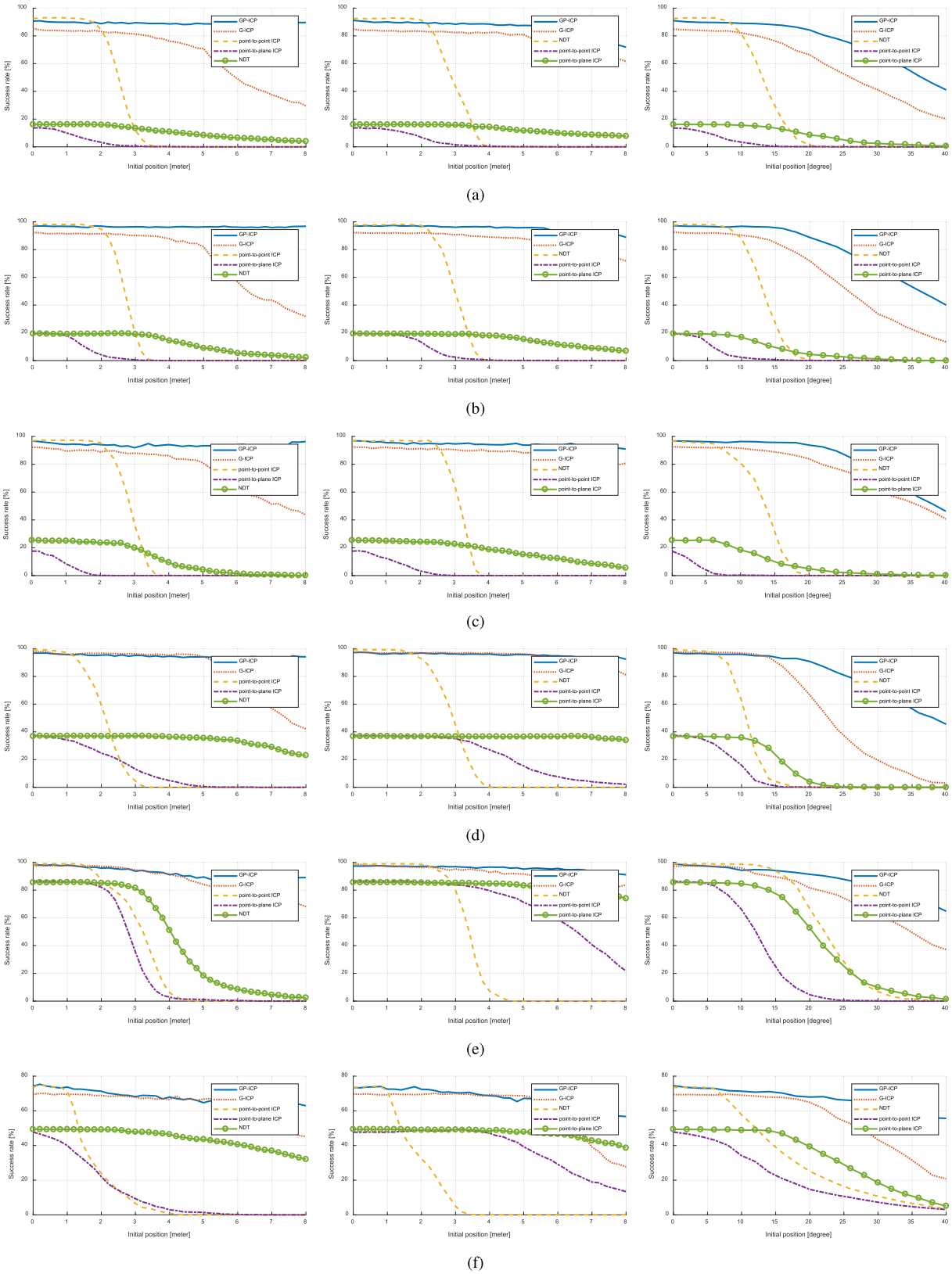


FIGURE 6. Success rate plots with Velodyne 64E for various initial transformation; (left) initial transformation in x-axis direction, (middle) y-axis direction, (right) initial yaw angle. (a) KITTI dataset sequence 00, (b) sequence 01, (c) Ford campus dataset, (d) Udacity dataset, (e) off-road dataset, and (f) slope dataset.

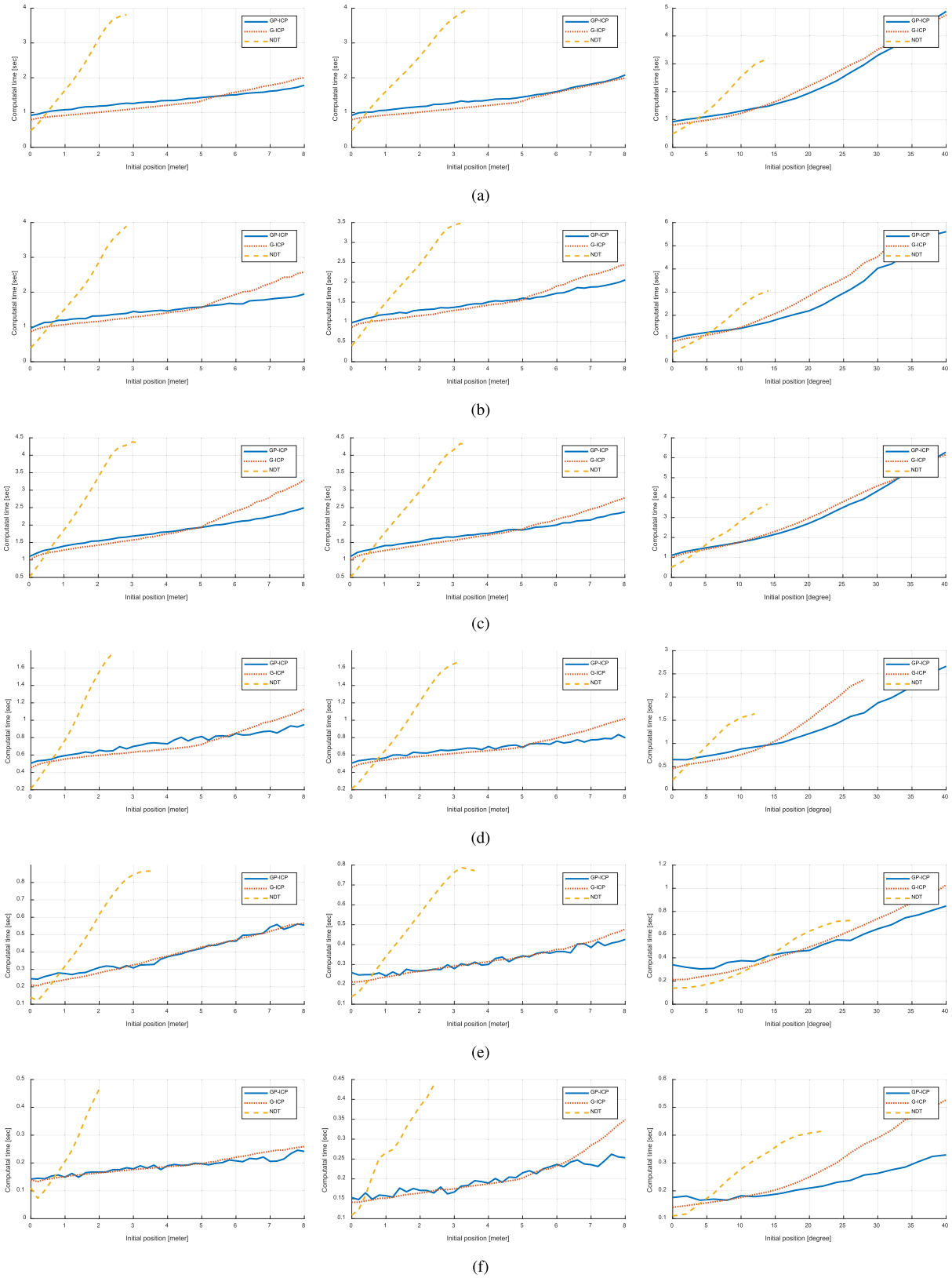
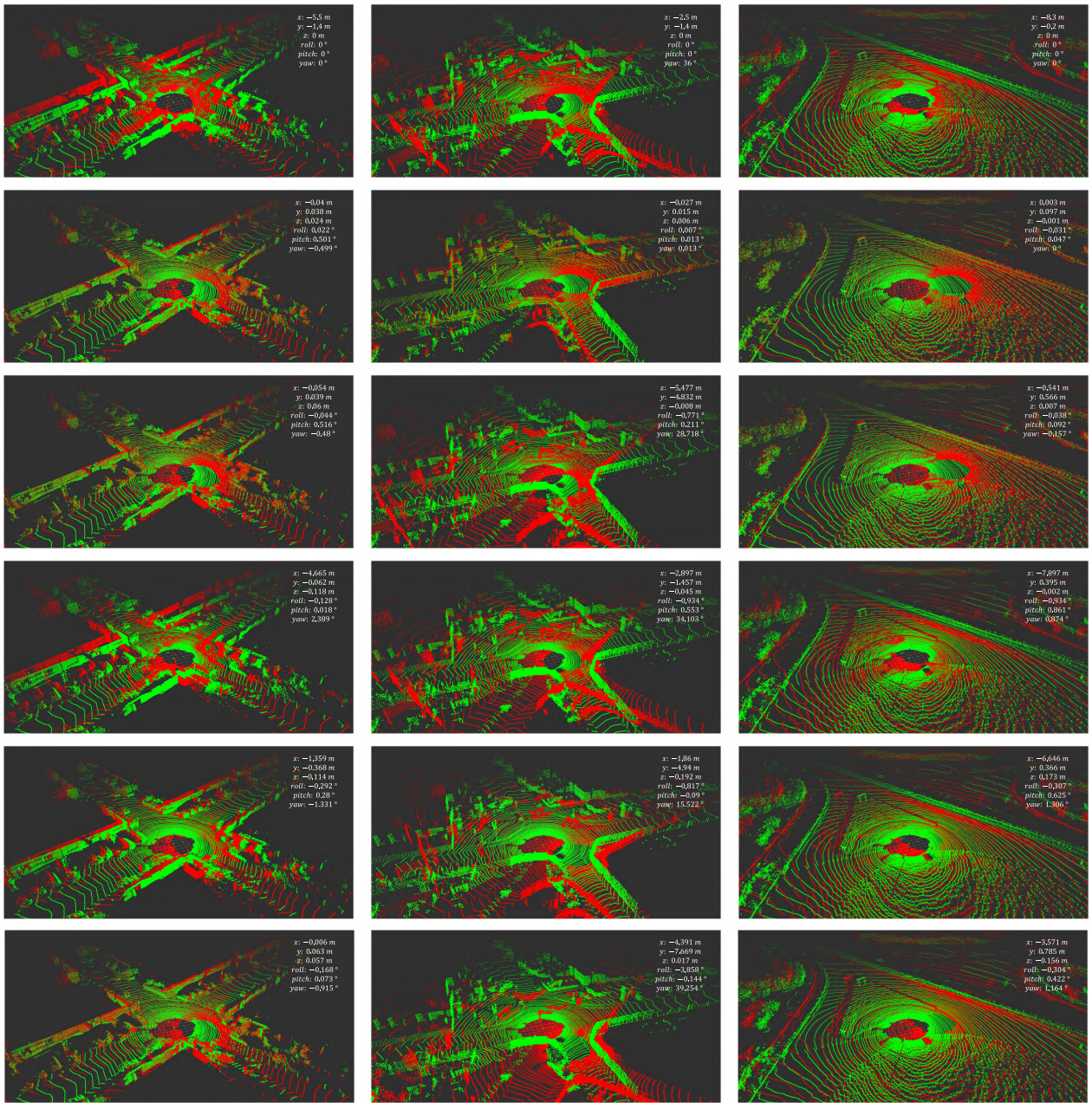


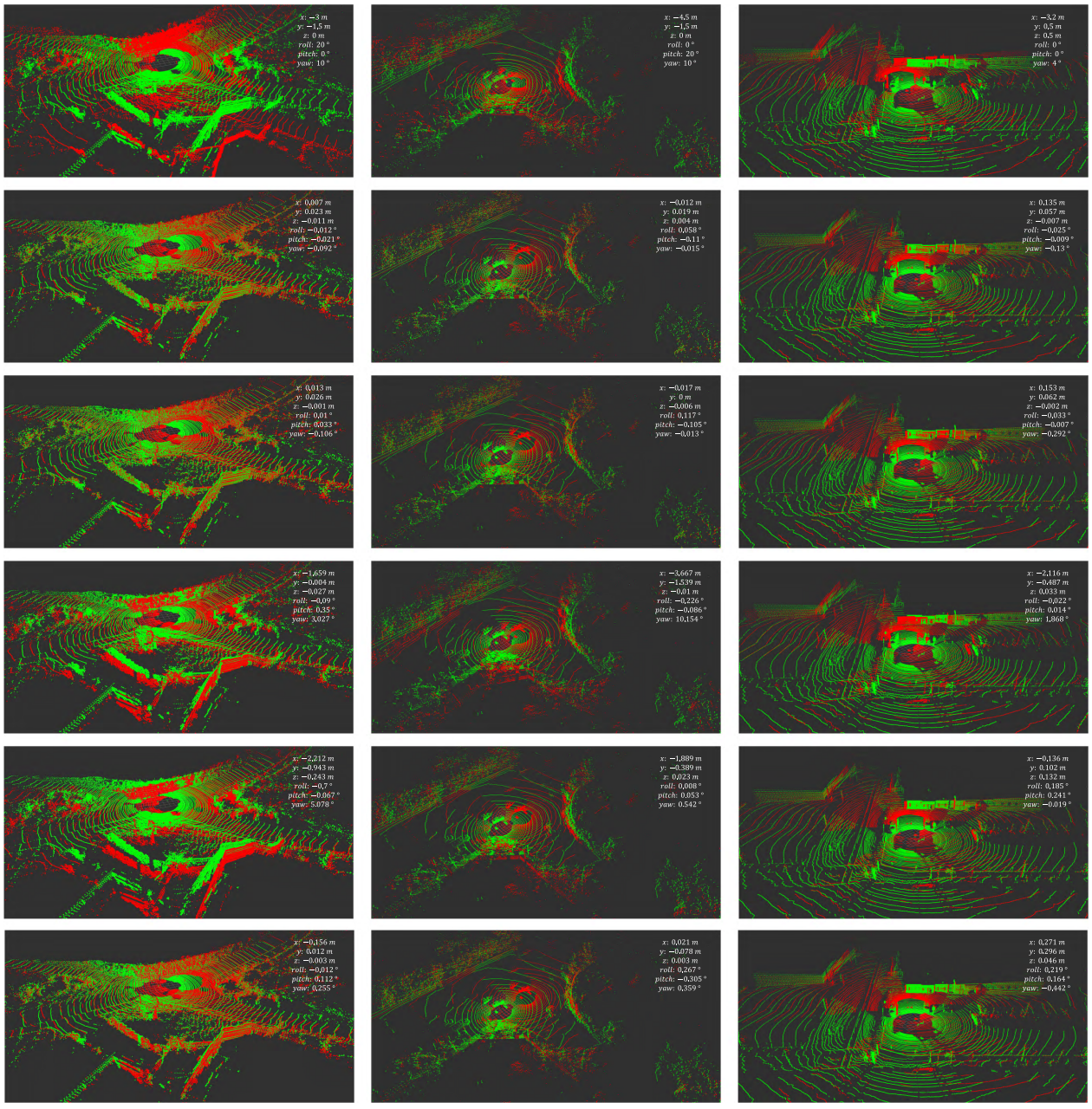
FIGURE 7. Success rate plots with Velodyne 64E for various initial transformation; (left) initial transformation in x-axis direction, (middle) y-axis direction, (right) initial yaw angle. (a) KITTI dataset sequence 00, (b) sequence 01, (c) Ford campus dataset, (d) Udacity dataset, (e) off-road dataset, and (f) slope dataset.



(a) (b) (c)

FIGURE 8. Point cloud matching result in various initial transformation; (top - bottom) initial transformation, GP-ICP, G-ICP, NDT, point-to-point ICP, and point-to-plane ICP. (a) An arbitrary pair of initial transformation (x: -5.5m, y: -1.4m, z: 0, roll: 0, pitch: 0, yaw: 0) from KITTI dataset sequence 00. (b) An arbitrary pair of initial transformation (x: -2.5m, y: -1.2m, z: 0, roll: 0, pitch: 0, yaw: 35°) from KITTI dataset sequence 00. (c) An arbitrary pair of initial transformation (x: -8.3m, y: -0.2m, z: 0, roll: 0, pitch: 0, yaw: 0) from KITTI dataset sequence 01.

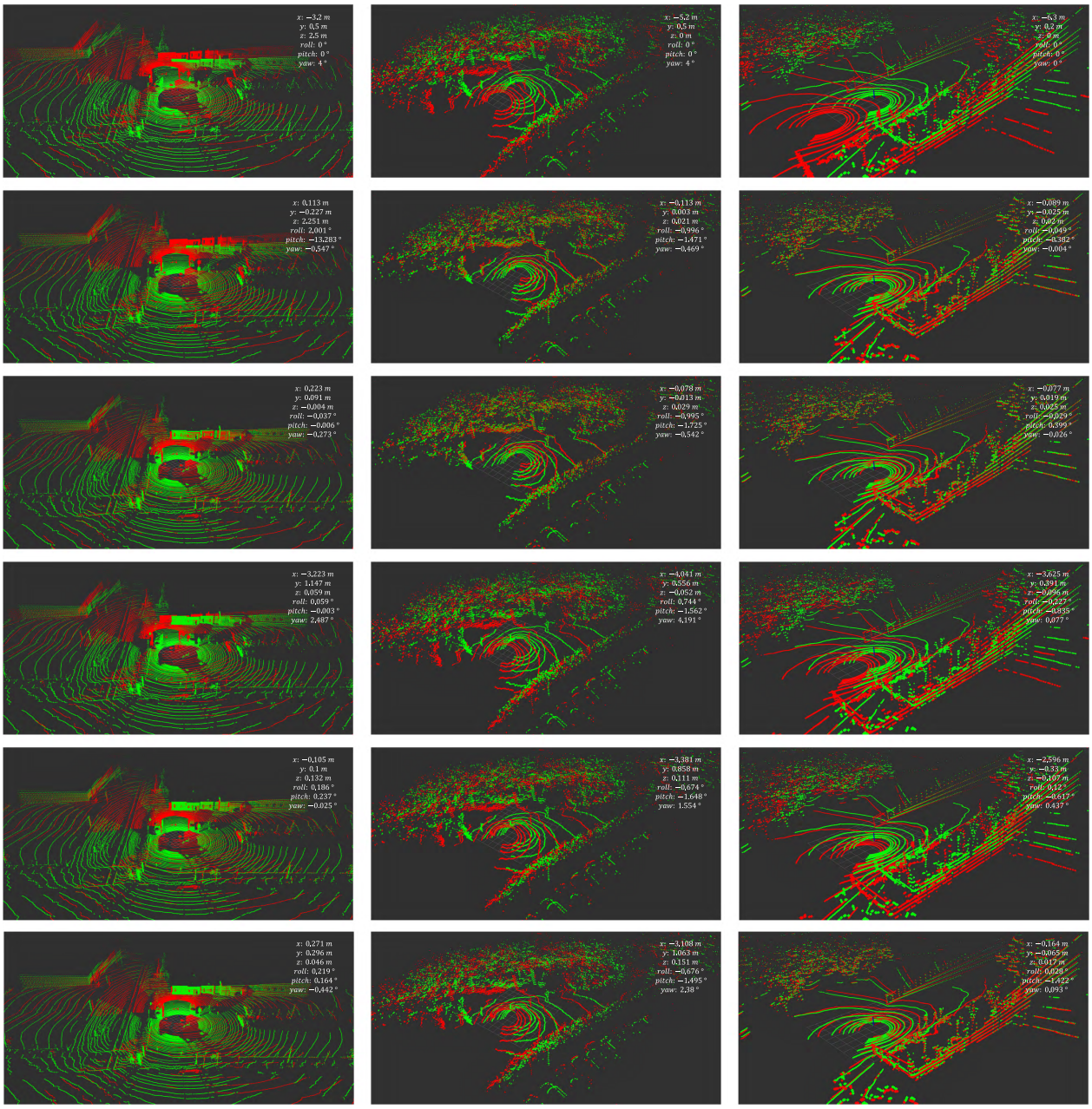
closest points between point clouds. In this paper, the distance threshold is set to 10cm and the overlapping area over 50% is regarded as a success. Fig. 6 shows that the proposed method provides superior results, particularly with regard to large initial transformations, than the other conventional methods. As mentioned in [16], the matching performances of point-to-point ICP and point-to-plane ICP are significantly degraded. The NDT algorithm shows acceptable results for small initial transformations only. The graph of the G-ICP algorithm is slightly similar to that of the GP-ICP algorithm, but the performance deteriorates sharply as the initial transformations in the direction of the x-axis and yaw angle increase. When considering the movements of the ground vehicle, matching in large initial transformations of the x-axis and yaw angle are more important than that of the other axes. Therefore, the proposed method is of considerable value as its superiority is verified using a public dataset for large initial transformations, especially in the x-axis and yaw angle.



(a) (b) (c)

FIGURE 9. Point cloud matching result in various initial transformation; (top - bottom) initial transformation, GP-ICP, G-ICP, NDT, point-to-point ICP, and point-to-plane ICP. (a) An arbitrary pair of initial transformation (x: -3.0m, y: -1.5m, z: 0, roll: 20°, pitch: 0, yaw: 0) from KITTI dataset sequence 01. (b) An arbitrary pair of initial transformation (x: -4.5m, y: -1.5m, z: 0, roll: 0, pitch: 20°, yaw: 10°) from Udacity dataset. (c) An arbitrary pair of initial transformation (x: -3.2m, y: 0.5m, z: 0.5, roll: 0, pitch: 0, yaw: 4°) from Ford campus dataset.

KITTI datasets offer the 6-DoF ground truth for a benchmark, while other datasets only offer the 3-DoF ground truth from RTK-GPS. Therefore, the root mean square error (RMSE) of datasets are compared with the ground truth as shown in Table 1. The comparison results are generated from the calculation using only the successfully matched pairs in Fig. 6. In Table 1, the errors of point-to-point ICP and point-to-plane ICP are larger than those of the other methods, which is as expected [13], [16]. The result of the NDT algorithm is slightly worse than those of its G-ICP and GP-ICP counterparts. Given that GP-ICP utilizes the same optimization method as G-ICP, the errors of G-ICP and GP-ICP are similar, as expected. The average computation time for successful matching according to the initial transformations is shown in Fig 7. Since the performance of point-to-point and point-to-plane ICPs is poor, the computation time of the proposed method is only compared with G-ICP and NDT. The average computation time of GP-ICP is slower than



(a) (b) (c)

FIGURE 10. **Point cloud matching result in various initial transformation; (top - bottom) initial transformation, GP-ICP, G-ICP, NDT, point-to-point ICP, and point-to-plane ICP. (a) An arbitrary pair of initial transformation (x: -3.2m, y: 0.5m, z: 2.5, roll: 0, pitch: 0, yaw: 4°) from Ford campus dataset as a failure case. (b) An arbitrary pair of initial transformation (x: -5.2m, y: 0.5m, z: 0, roll: 0, pitch: 0, yaw: 4°) from offroad dataset. (c) An arbitrary pair of initial transformation (x: -6.3m, y: 0.2m, z: 0, roll: 0, pitch: 0, yaw: 0) from Udacity dataset.**

that of G-ICP at small initial transformations while GP-ICP is much faster at large initial transformations. According to Section II.C, the search method of GP-ICP is 1.33 times slower than that of G-ICP when $n = 100,000$, $L = 20$, and $|(m_i)_z - (T \cdot b_i)_z| > \epsilon$ is 15%. Nevertheless, the overall processing time of GP-ICP is faster than that of G-ICP at large initial transformations. The computation time of NDT is usually low and significantly increases in large initial transformations.

The results of point cloud matching are shown in Figs. 8-10. The green dots denote the target point cloud and the red dots represent a transformation of the query point cloud with (a) the initial transformation, (b) GP-ICP, (c) G-ICP, (d) NDT, (e) point-to-point ICP, and (f) point-to-plane ICP.

The matching results with various initial transformations in x, y, and yaw are shown in Fig. 8. In this case, GP-ICP shows superior results than the other conventional algorithms

TABLE 1. Root mean square error of each algorithm compared with the ground truth (units: meter, degree).

(a)

	x	y	z	roll	pitch	yaw
GP-ICP	**0.049**	**0.060**	0.036	**0.094**	**0.061**	0.079
G-ICP	0.056	0.061	0.036	0.136	0.077	**0.071**
NDT	0.048	0.076	**0.034**	0.112	0.063	0.078
Point-to-point ICP	0.245	0.235	0.064	0.372	0.163	0.382
Plane-to-plane ICP	0.153	0.102	0.057	0.361	0.158	0.249

(b)

	x	y	z	roll	pitch	yaw
GP-ICP	**0.060**	0.088	0.031	**0.037**	**0.061**	0.061
G-ICP	0.064	0.087	0.027	0.037	0.067	**0.060**
NDT	0.060	**0.085**	**0.024**	0.041	0.066	0.070
Point-to-point ICP	0.371	0.262	0.135	0.154	0.096	0.331
Plane-to-plane ICP	0.224	0.162	0.127	0.120	0.116	0.256

(c)

	x	y	z
GP-ICP	**0.044**	0.039	**0.026**
G-ICP	0.054	**0.036**	0.030
NDT	0.045	0.050	0.028
Point-to-point ICP	0.215	0.201	0.054
Plane-to-plane ICP	0.145	0.092	0.045

(d)

	x	y	z
GP-ICP	0.069	**0.055**	**0.018**
G-ICP	**0.066**	0.057	0.022
NDT	0.073	0.053	0.021
Point-to-point ICP	0.325	0.264	0.078
Plane-to-plane ICP	0.203	0.109	0.084

in environments with plentiful vertical geometric information such as walls of buildings, trucks of trees, or poles. However, the performance of GP-ICP is similar to that of G-ICP in environments such as empty place or bushes area where there are little vertical geometry. Therefore, GP-ICP is better than G-ICP in overall performance. Although the movements in roll, pitch, and z-axis for ground vehicle are not large in general cases, some cases such as inclined, off-road, or unexpected situations might result in certain differences to roll, pitch, and z-axis. Therefore, small initial transformations in roll, pitch, and z-axis are tested as shown in Fig. 9. From the results, the proposed method is also verified to match well in small unexpected movements (roll, pitch, and z-axis). However, the large initial transformation in z-axis results in poor matching while the other methods (G-ICP, point-to-plane ICP) are matched well as shown in Fig. 10(a). Since this paper employs horizontal search method, it is hard to handle a large initial transformation in z-axis. However, this case rarely happens in ground vehicle conditions. The matching results with Velodyne VLP-16 for off-road and inclined environments are shown in Fig. 10. As a result, the superiority of the proposed algorithm is verified in various environments, initial transformation, and sensor system.

IV. CONCLUSION

This paper proposed a new matching method with two-point cloud for ground vehicles. We termed this method as GP-ICP. While the conventional methods utilize only geometric information of point cloud data, GP-ICP can consider a ground condition in the process of point cloud matching. Notably, GP-ICP provides a big advantage in terms of large initial transformations. The proposed method was validated by testing an open public data set, by comparing with other state-of-the-art methods for matching performance, error, and computation time. The proposed method was also tested in inclined and off-road environments using our own robot system. The results of this paper are bound to be very useful to the fields of mobile robots, and it can be extended to autonomous vehicles as well. We publicly release the source code of the GP-ICP algorithm in the following website: https://github.com/hyungjinkim0508/gpicp, and we expect that the method will be implemented for various applications in several industries. In future works, the proposed method will be tested to generate vast point cloud map data.

REFERENCES

[1] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 163–168.

[2] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita, "Lidar scan feature for localization with highly precise 3-D map," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 1345–1350.

[3] I. Baldwin and P. Newman, "Road vehicle localization with 2D push-broom LIDAR and 3D priors," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp. 2611–2617.

[4] H. Kim, B. Liu, C. Y. Goh, S. Lee, and H. Myung, "Robust vehicle localization using entropy-weighted particle filter-based data fusion of vertical and road intensity information for a large scale urban area," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1518–1524, Jul. 2017.

[5] Mitsubishi Electric Corporation. *Mobile Mapping System—High-Accuracy GPS Mobile Measuring Equipment*. Accessed: Sep. 2018. [Online]. Available: http://www.mitsubishielectric.com/bu/mms/

[6] C. Premebida, O. Ludwig, and U. Nunes, "LIDAR and vision-based pedestrian detection system," *J. Field Robot.*, vol. 26, no. 9, pp. 696–711, Sep. 2009.

[7] C. Urmson, J. Anhalt, M. Clark, T. Galatali, J. P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, Y. H. Kato, P. Koon, K. Peterson, B. Smith, S. Spiker, E. Tryzelaar, and W. R. Whittaker, "High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004," Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-04-37, 2004.

[8] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[9] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[10] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, 1994. doi: 10.1007/BF01427149.

[11] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans," *J. Intell. Robot. Syst.*, vol. 18, no. 3, pp. 249–275, 1997.

[12] J. Minguez, F. Lamiraux, and L. Montesano, "Metric-based scan matching algorithms for mobile robot displacement estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 4, Apr. 2005, pp. 3557–3563.

[13] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, Oct. 2003, pp. 2743–2748.

[14] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 766–782.

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

[16] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robot. Sci., Syst.*, vol. 2, no. 4, p. 435, Jun. 2009.

[17] J. Servos and S. L. Waslander, "Multi-channel generalized-ICP: A robust framework for multi-channel scan registration," *Robot. Auton. Syst.*, vol. 87, pp. 247–257, Jan. 2017.
[18] M. L. Tazir, T. Gokhool, P. Checchin, L. Malaterre, and L. Trassoudaine, "CICP: Cluster iterative closest point for sparse-dense point cloud registration," *Robot. Auton. Syst.*, vol. 108, pp. 66–86, Oct. 2018.
[19] G. Agamennoni, S. Fontana, R. Y. Siegwart, and D. G. Sorrenti, "Point clouds registration with probabilistic data association," in *Proc. EEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4092–4098.
[20] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.
[21] C. Ulas and H. Temeltas, "3D multi-layered normal distribution transform for fast and long range scan matching," *J. Intell. Robot. Syst.*, vol. 71, no. 1, pp. 85–108, 2013.
[22] A. Das, J. Servos, and S. L. Waslander, "3D scan registration using the normal distributions transform with ground segmentation and point cloud clustering," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2013, pp. 2207–2212.
[23] A. Das and S. L. Waslander, "Scan registration using segmented region growing NDT," *Int. J. Robot. Res.*, vol. 33, no. 13, pp. 1645–1663, 2014.
[24] G. Pandey, S. Giri, and J. R. Mcbride, "Alignment of 3D point clouds with a dominant ground plane," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2143–2150.
[25] A. Zaganidis, L. Sun, T. Duckett, and G. Cielniak, "Integrating deep semantic segmentation into 3-d point cloud registration," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2942–2949, Oct. 2018.
[26] F. Moosmann and C. Stiller, "Velodyne SLAM," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 393–398.
[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[28] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, pp. 1231–1237, Aug. 2013.
[29] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *Int. J. Robot. Res.*, vol. 30, no. 13, pp. 1543–1552, 2011.
[30] Udacity. *Udacity Self-Driving Car Dataset*. Accessed: Jan. 2019. [Online]. Available: https://github.com/udacity/self-driving-car
[31] Velodyne LiDAR. *Velodyne LiDAR HDL-64E*. Accessed: Jan. 2019. [Online]. Available: https://velodynelidar.com/hdl-64e.html

HYUNGJIN KIM received the B.S. degree in electrical engineering from Kyunghee University, Suwon, South Korea, in 2012, and the M.S. degree from the Department of Civil and Environmental Engineering (Robotics Program), Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include simultaneous localization and mapping, high-definition map generation, autonomous vehicle, and robot navigation.

SEUNGWON SONG received the B.S. degree in mechanical engineering and the M.S. degree in robotics program from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2016, and 2018, respectively, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering. His research interests include structural health monitoring using simultaneous localization and mapping, robotics, and robot navigation.

HYUN MYUNG received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1992, 1994, and 1998, respectively. He was a Senior Researcher with the Electronics and Telecommunications Research Institute, Daejeon, from 1998 to 2002, the CTO and the Director of the Digital Contents Research Laboratory, Emersys Corporation, Daejeon, from 2002 to 2003, and a Principle Researcher with the Samsung Advanced Institute of Technology, Yongin, South Korea, from 2003 to 2008. Since 2008, he has been a Professor with the Department of Civil and Environmental Engineering, KAIST, where he is currently the Head of the KAIST Robotics Program. Since 2019, he has been a Professor with the School of Electrical Engineering. His current research interests include simultaneous localization and mapping, robot navigation, machine learning, artificial intelligence, deep learning, swarm robot, and structural health monitoring using robotics.