



## Cluster ICP: Towards Sparse to Dense Registration

Mohamed Lamine Tazir, Tawsif Gokhool, Paul Checchin, Laurent Malaterre,  
Laurent Trassoudaine

### ► To cite this version:

Mohamed Lamine Tazir, Tawsif Gokhool, Paul Checchin, Laurent Malaterre, Laurent Trassoudaine.  
Cluster ICP: Towards Sparse to Dense Registration. 15th International Conference on Intelligent  
Autonomous Systems, Jun 2018, Baden-Baden, Germany. hal-01863896

**HAL Id: hal-01863896**

**<https://hal.science/hal-01863896>**

Submitted on 29 Aug 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Cluster ICP: Towards Sparse to Dense Registration

Mohamed Lamine Tazir<sup>1</sup>, Tawsif Gokhool<sup>2</sup>, Paul Checchin<sup>1</sup>, Laurent Malaterre<sup>1</sup>, and  
Laurent Trassoudaine<sup>1</sup>

<sup>1</sup> Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000  
Clermont-Ferrand, France,

tazir.med@gmail.com, paul.checchin@uca.fr,  
laurent.malaterre@uca.fr, laurent.trassoudaine@uca.fr

<sup>2</sup> Université de Picardie Jules Vernes, France,  
tawsif.gokhool@u-picardie.fr

**Abstract** Normal segmentation of geometric range data has been a common practice integrated in the building blocks of point cloud registration. Most well-known point to plane and plane to plane state-of-the-art registration techniques make use of normal features to ensure a better alignment. However, the latter is influenced by noise, pattern scanning and difference in densities. Consequently, the resulting normals in both a source point cloud and a target point cloud will not be perfectly adapted, thereby influencing the alignment process, due to weak inter surface correspondences. In this paper, a novel approach is introduced, exploiting normals differently, by clustering points of the same surface into one topological pattern and replacing all the points held by this model by one representative point. These particular points are then used for the association step of registration instead of directly injecting all the points with their extracted normals. In our work, normals are only used to distinguish different local surfaces and are ignored for later stages of point cloud alignment. This approach enables us to overcome two major shortcomings; the problem of correspondences in different point cloud densities, noise inherent in sensors leading to noisy normals. In so doing, improvement on the convergence domain between two reference frames tethered to two dissimilar depth sensors is considerably improved leading to robust localization. Moreover, our approach increases the precision as well as the computation time of the alignment since matching is performed on a reduced set of points. Finally, these claims are backed up by experimental proofs on real data to demonstrate the robustness and the efficiency of the proposed approach.

**Keywords:** registration, dense to sparse, selection, clustering, matching

## 1 Introduction

In a generic representation of the environment, point clouds can be viewed as a collection of 3D point entities bearing a color or intensity information depending on the acquisition sensor (LiDAR, RGBD or time of flight cameras). To obtain a more meaningful information about the semantic structure of the environment, it is more instructive to rather consider a collective set of points representing the same surface. From there, several surface indices can be extracted such as their normals, curvatures and region bounds for example. Moreover, due to limitation of the field of view of 3D sensors,

coupled with the complex geometry of the scanned surrounding, registration methods are required to be more robust in order to deal with data taken from large viewpoints as well as different sensor resolution.

Extraction of surface normals is a double-edged tool, which guarantees good results if accurately exploited, but can also lead to divergence of the alignment process if badly used. Since normal features are based essentially on the estimation of neighbouring points attributed to the same surface in general, they are however subjected to sensor noise, resolution and scanning patterns. Consequently, this reverberates on the alignment process due to weak inter surface correspondences between the source and the target point cloud. Furthermore, with the advancement of 3D sensors in the market, the problematic of sparse to dense registration has emerged out [1]. In this trend, software packages such as PCL [25] has made the identification and the treatment of the above mentioned problem more accessible. Eventually, an elegant solution provided by a successful sparse to dense point cloud alignment, results in interesting robotics applications such as the case of a monocular camera localization in a 3D model [33,7] or augmenting the environment with more consistent data as in [17].

In order to support the claims stated above, an illustration of sparse to dense registration is given in Fig. 1. A dense point cloud is obtained from a 3D LiDAR Leica P20<sup>3</sup> scanner, whilst the sparser one is extracted from an HDL-32E Velodyne<sup>4</sup>. Figures 1 (g), (h), (i), (j) are samples of various places in a scene. The 3D points of the source and target clouds are represented in blue and green respectively, whilst their normals are in white and red. Because of the large discrepancies in density between the two point clouds, registration methods based on the classical point-to-point ICP metrics fail to provide a good pose estimate. The difficulty lies in the fact that there are no direct correspondences between the source and the target point clouds. Moreover, these figures also depict the dissimilarity between normals pertaining to the same surface, which theoretically should have the same orientations. This change is due mainly to the presence of wide amount of noise, pattern scanning, distortion and varying resolutions. This is the major problem of the methods that use geometric features according to [28,13].

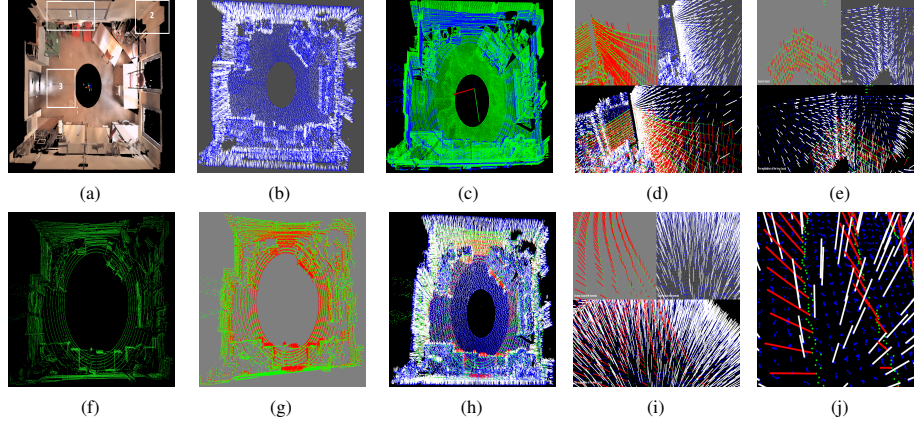
In this paper, a novel registration method is introduced, exploiting normals differently. It does not seek for each point its nearest neighbor sharing the same normal, nor introducing normals in the error function or in the minimization process. To remediate for the disturbances in the registration framework, hence inaccuracies in the final result, a voxelization is performed on both clouds to maintain the topological details of the scene. Then for each voxel, a normal-based classification of points is done. Thereafter, only one point of each local surface is maintained for the association step. This process results into points which offers better compatibility in terms of surface representation between the source and the target point clouds. This approach is not about points sampling, but rather an improved selection of points is achieved for the matching stage. Thereon, the process evolves in a classical ICP like framework for pose estimation.

The contributions of this paper are threefold. The first main contribution is to perform an efficient voxelization method operating on top of a clustering technique for electing one representative point for each local surface. This process aims to establish

---

<sup>3</sup> Leica P20: <http://leica-geosystems.com/>

<sup>4</sup> Velodyne LiDAR: <http://velodynelidar.com/>



**Figure 1.** Dense to sparse registration: (a) point cloud obtained from the Leica P20 LiDAR with 88 556 380 points; (f) point cloud obtained from an HDL-32E Velodyne with 69 984 points; (b) and (g) are their corresponding point clouds with normal vectors; (c) registration result of point cloud (a) and (f) using our proposed method; (h) normal vectors corresponding to (c); (d), (e) and (i) are exploded views of places indicated in (a); (j) is a close up view of (i).

two sets of points which are most likely to be matched, hence providing good correspondences. It consists of three stages: voxelization, clustering and matching. As a result, it:

1. reduces the amount of data to be processed during the matching phase,
2. improves the matching robustness by allowing only the association of compatible points,
3. avoids wrong associations that decrease alignment accuracy,
4. does not use unnecessary points that do not provide further information more than the ones used,
5. improves convergence and accuracy simultaneously.

The second contribution is that the proposed method is totally independent of the density (number of points, scanning resolution) of the two clouds, scanning patterns (nature of sensors). It takes as input point clouds of different resolution, gathered by different sensors, or with the same sensor. It is also based only on the geometric characteristics of the points, which makes it independent of weather and illumination conditions. Thirdly, normals are computed once before starting the process and are used only to distinguish the different local surfaces. They are not used in the alignment process.

The rest of this paper is organized as follows: in Section 2, an overview of the state of the art of registration methods is given. Section 3 details the proposed method. This is followed by experiments and a comparison with the state-of-the-art methods in order to evaluate the proposed approach. Finally, conclusions and suggestions of future works are discussed in Section 5.

## 2 Related Work

Registration algorithms assemble two representations of an environment in a single reference frame. The problem of registration has been dealt with extensively in several



studies over the last 25 years. This started with geometric approaches leading to the appearance of the Iterative Closest Point (ICP) algorithm [4,8]. ICP is used to calculate the optimal transformation fitting two point clouds by a two-step process: matching of points and minimizing a metric describing the misalignment [19]. These two-steps iterate to minimize the matching error and thus improve alignment. In the literature, two main groups of registration methods are identified:

- feature-based methods (approaches based on features extraction);
- dense methods (approaches exploit all the points in the cloud).

## 2.1 Feature-based Approaches

Feature-based methods are generally used in outdoor environments [17]. They are based on the use of features, which may be points that are easily identified by their apparent character (position, local information contents, mathematical definition, etc.) with respect to the other points. A good feature requires stability and distinctiveness [28]. In other words, detected features should be consistent in all the frames. They should be robust to noise and invariant to rotation, perspective distortion and changes of scale [28,9] [11].

Features extraction from point cloud representation is well documented literature. One can find the 3D Scale Invariant Feature Transform (3DSIFT), which is an extension of the 2D version proposed by Lowe in 1999 [15]. The 3D version was adapted by the PCL [25] community using the curvature of points instead of the intensity of pixels [12]. The method uses a pyramidal approach to reach the scale invariance characteristic of features. To achieve invariance against rotation, it assigns orientations to keypoints. This adds to an incomplete list of features such as FPFH [23], VFH (Viewpoint Feature Histogram) [24], CVFH (clustered viewpoint feature histogram) [2] to name a few.

However, feature extraction techniques are often cumbersome to determine and pose a problem to real-time applications [9], making them unsuitable for applications that require efficiency. Furthermore, the necessity of very dense clouds are required in order to obtain good features, which compromises with the use of sparse clouds [1,28,34,31]. More importantly, these methods are environment specific [6], which may result in the rejection of good data [20].

## 2.2 Dense Approaches

Dense approaches make use of all the points from both clouds, and require an initial guess (transformation) between the two clouds, which makes them sensitive to wrong initialization [35,27,9]. Despite the use of all the points, these methods are generally faster than feature-based approaches [27].

Dense techniques are however well adapted to a classical ICP framework. As pointed out by Pomerleau [21], its easy implementation and simplicity, are both its strength and its weakness. This led to the emergence of many variants of the original solution, adapted in many ways, throughout the years. At the very outset, Chen [8], improved the standard ICP by using point-to-plane metric instead of the Euclidean distance error. This approach takes advantage of surface normal information to reject wrong pairing. However, this approach fails when dealing with clouds of different densities, since

normals computation are affected by the change in resolution, presence of noise and distortion [10,13].

The Normal Distributions Transform (3D-NDT) [18] discretizes the 3D points with their normals in cells, where each one is modeled by a matrix, representing the probability of occupation of its points (linear, planar and spherical). Then, a non-linear optimization is performed to calculate the transformation between the two clouds. However, according to [10], the NDT is not suitable for systems with low computing power capability.

An efficient approach for dense 3D data registration was presented in [26]. This probabilistic version of ICP called Generalized ICP (GICP) is based on a Maximum Likelihood Estimation (MLE) probabilistic model. It exploits local planar patches in both point clouds, which leads to plane-to-plane concept. Since this algorithm is point-to-plane variant of ICP, it has similar drawbacks, especially those related to normals computation. For instance in [13], it is shown that the non-uniform point densities cause inaccurate estimates, which degrade the performance of the algorithm. Moreover, Agamennoni [1] affirmed that the GICP does not work well in outdoor unstructured environment.

Serafin [27] extended the GICP algorithm by using the normals in the error function and in the selection of correspondences, which according to the authors, increases the robustness of the registration.

Our approach, called CICP for Cluster Iterative Closest Point, uses an (NDT and NICP)-like representation, however, it is different from the NDT in the way it uses the points of each voxel to determine local surfaces and get one representative point from each local surface to the matching process. In contrast, NDT computes a Gaussian distribution in points of each voxel using the vicinity of each point. Whereas, NICP uses an image projection of the voxel grid representation to compute statistics, and considers each point with the local features of the surrounding surface. These features, namely normal and curvature, are calculated for each point from its neighboring points, with a computational complexity of  $\mathcal{O}(K \times N)$ , where  $K$  is the number of the neighboring points used to compute each normal and  $N$  the total number of points. Additionally, these features are used later in the process of point matching between the two clouds, as opposed to our method, that does not use normals in the matching process. Because of the difference in density, pattern scanning, and presence of noise, will lead to noisy normals and, hence, inaccurate results.

### 3 Proposed Method

In this paper, a novel registration method exploiting normals is introduced. We adopt the Rusinkiewicz [22] decomposition and propose a new selection strategy, which aims to improve the pairing process. Figure 2 illustrates the pipeline of the proposed method.

#### 3.1 Surface Normal Segmentation

CICP starts with the estimation of normals of the source and target point clouds using Principal Component Analysis (PCA) [14] follows:

$$C = \frac{1}{k} \sum_{i=1}^k (p_i - \bar{p})^T (p_i - \bar{p}), \quad (1)$$

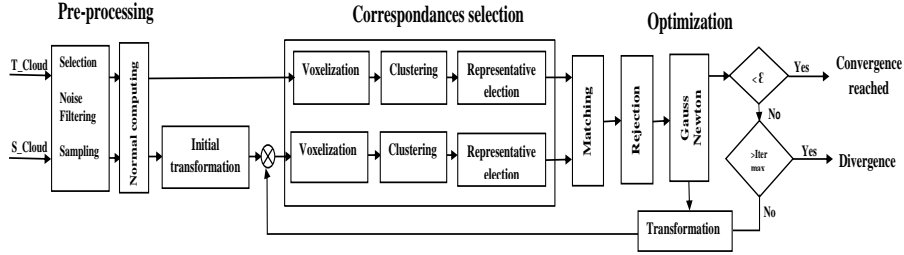
where,  $C$  is the covariance matrix of the nearest neighbors (NNs),  $k$  is the number of considered nearest neighbours  $p_i$ , and  $\bar{p}$  is its corresponding centroid along the tangent plane.

Thereafter, the target cloud is subdivided into small voxels. Points belonging to each voxel are subjected to a classification process based on their normals, giving rise to different groups of points, according to the geometric variation of each voxel. Each group of points represents a local surface since they share the same normal vector. A single point is chosen from a local surface extract to be used for the matching process. The closest point to the centroid of each local surface is elected a winner.

Similarly, the source point cloud is first transformed into the reference frame of the target cloud using a pose estimate before undergoing a similar process; voxelization, normals-based classification, designation of point's representatives. At the end of these steps, the method results into two improved sets of points from the corresponding clouds. Each set contains the most probable points to match with the points of the second set (this is more particularly in the overlapping area of the two clouds, as it reflects the same geometry seen from two different viewpoints).

### 3.2 CICIP Matching Pipeline

The main contribution of this paper is the proposal of a new selection strategy. As mentioned above, instead of matching point-to-point as the classical ICP variants, points pass through an election process, which gives rise to one representative point for each small region. These representatives appear as the most likely points to be matched between each other. These good matches ultimately result in an accurate motion between



**Figure 2.** Overview of the CICIP pipeline. Given two point clouds, CICIP starts by computing the surface normals of the two clouds. It looks for points sharing the same local properties, and then elects one representative point from each local cluster. This election process is based on 3D position of points and their normals. It consists of three sub-tasks: (1) Voxelization: a set of 3D cubic regions (voxels) is generated where all voxel points have very close spatial positions. (2) Clustering: classify all points of each voxel according to their normals. (3) Matching: once this grouping step is completed, the last task consists in selecting one point from each cluster (local surface) in each voxel. Representative points serve as candidates for correspondence process. As a result, few points are used in the matching process, but which are most likely to be associated, thereby, improving on convergence and accuracy simultaneously.

the two clouds (shown in the results section). This election process is based on 3D position of points and their normals. It consists of three sub-tasks: (1) voxelization, (2) clustering and (3) representative election. The first task performs a spatial grouping which attempts to preserve the topological information based on the 3D position of the points. A set of 3D cubic regions (voxels) is generated where all points within the voxel have very close spatial positions. The second task bundles all points of each voxel based on their normals. Once this grouping is done, we perform the last task, which selects one point for each cluster (local surface) in the voxel for the matching process. Algorithm 1 depicts the workflow of CICP.

**Voxelization** It is applied in order to maintain the topological details of the scrutinized surface. As normal computation depends on the number of neighbouring points and as the resolution of points of the two clouds is different, voxelization with the same voxel size aims to generate equivalent local regions in the two clouds. A common criterion of comparison now becomes feasible. Therefore, the voxel size parameter is of paramount importance for our technique and it should be chosen carefully in order to keep the fundamental characteristics of both point clouds; be it dense or sparse with topological details. A voxel grid with cell size  $d$  is generated, where the following set of rules are verified:

**Definition 1 (Sparse Cloud).** A sparse cloud is a cloud  $C = (V, P)$  in which:  $|P| = \mathcal{O}(|V|)$ .

**Definition 2 (Dense Cloud).** A dense cloud is a cloud  $C = (V, P)$  in which:  $|P| = \mathcal{O}(k * |V|)$ , with  $k > 2$ .

whereby,

$V$ : set of voxels,  $P$ : set of points,  $\mathcal{O}$ : proportionality operator.

Definitions 1 and 2 are proposed to frame the notions of sparsity and density of point clouds. The voxel size is set according to the number of points in the sparse cloud, so that each voxel contains at least one point. This choice ensures a significant difference in density between the two clouds. A dense cloud, in our case, contains at least twice as many points as the sparse cloud. Otherwise, they are considered as equivalent.

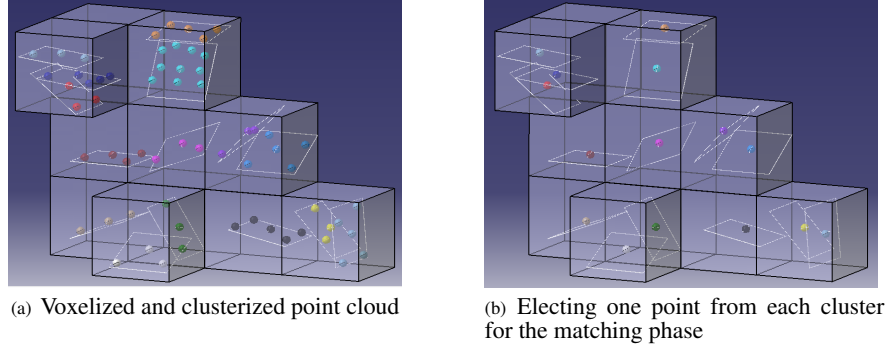
At the beginning, the procedure applies a bounding box to the entire sparse cloud by finding the minimum and maximum positions of points along the three axes  $X$ ,  $Y$  and  $Z$ . The number of voxels for this bounding box is determined by the number of points and the voxel size is deduced. The same procedure is applied to the dense cloud.

1. Voxel assignment: each voxel is identified by a unique linear index. If  $i, j, k$  represent the voxel indices in the  $X, Y, Z$  dimensions, respectively,  $numDivX, numDivY$  are the number of voxels along  $X$  and  $Y$  axes, the formula to encode the linear index is [29]:

$$idx = i + j \times numDivX + k \times numDivX \times numDivY \quad (2)$$

According to (2), we assign an index  $idx$  to each point. This relationship allows direct access to the desired voxel, thereby avoiding a linear search as in [32].

2. Voxel suppression: as the shape of the point cloud is arbitrary, the step of delimiting points by a bounding box creates many empty voxels which are later pruned out. Eventually, voxelization helps to filter noise from voxels where there is insufficient occupational evidence. An illustration of the described approach is given in Fig. 3.



**Figure 3.** Voxelized/normal-based clustering for matching process.

**Clustering** The process of electing one point from each local surface makes them good candidates for point correspondence searching, thereby rejecting wrong matches impacting alignment accuracy. At first, all the “voxelized” points are taken and a classification method is applied to identify points belonging to the same surface. In our work, k-means clustering [3] is used as the classification technique based on the normal of each point. The appropriate number of clusters (local surfaces) within each voxel is determined using the Elbow method [29,30]. An illustration of the described approach is given in Figure 3.

Grouping the point clouds using their normal aims at:

- improving the robustness of the matching step by only allowing the pairing of compatible points,
- reducing the amount of data to be processed during the matching stage.

**Matching** The clustering process generates a reduced, but different number of points in both clouds. These two sets of points are used for matching. To boost up the matching process, an off shelf PCL [25] implementation of the  $k$ -d trees is used, whereby matching is achieved using  $L_2$  norm. Outliers between the dense and sparse sets are handled using a suitable threshold.

### 3.3 Optimization Framework

In the case of a point-to-point metric, the error function to be minimized is given by:

$$E(x) = \sum_{i=1}^N \|T(\tilde{x}) p_i - q_i\|^2 \quad (3)$$

The localization problem of a sparse to a dense point cloud (or vice-versa) resolves to estimating the relative transformation  $T(\tilde{x})$  between point clouds  $\{p, q\} : \forall \{p_i, q_i\} \in \mathbb{R}^3$ . The principle of rigid body motion is applied where the transformation of a point tethered to a coordinate frame represent the whole compact body motion. For any point pair lying on the body, metric properties such as distances and orientation are preserved. This kind of body motion, discussed subsequently forms part of the special euclidean group  $\mathbb{SE}(3)$ .

Inter-frame incremental displacement is further defined as an element of the Lie groups applied on the smooth differential manifold of  $\mathbb{SE}(3)$  [5], also known as the group of direct affine isometries. Motion is parametrized as a twist (a velocity screw motion around an axis in space), denoted as  $\mathbf{x} = \{[\omega, v] | v \in \mathbb{R}^3, \hat{\omega} \in so(3)\} \in se(3)$ :  $\omega = [\omega_x \ \omega_y \ \omega_z]$ ,  $v = [v_x \ v_y \ v_z]$ , with  $so(3) = \{\hat{\omega} \in \mathbb{R}^{3 \times 3} | \hat{\omega} = -\hat{\omega}^T$ , where  $\omega$  and  $v$  are the angular and linear velocities respectively. The reconstruction of a group action  $\hat{T} \in \mathbb{SE}(3)$  from the twist consists of applying the exponential map using Rodriguez formula [16].

Equation (3) is solved iteratively in a Gauss Newton fashion, where at each iteration, a new error  $E$  and a new Jacobian matrix  $J(0)$  are computed in order to obtain the update  $x$  by:

$$x = -\left(J(0)^T J(0)\right)^{-1} J(0)^T e(x) \quad (4)$$

and the rigid transformation is updated as follows:

$$\hat{T} \leftarrow \hat{T} T(x) \quad (5)$$

Minimization is stopped when the error:  $\|e\|^2 < \alpha$  occurs, or when the calculated increment becomes too small:  $\|x\|^2 < \varepsilon$ , where  $\alpha$  and  $\varepsilon$  are predefined stop criteria.

## 4 Results

Our CICP approach is implemented in C++ without code optimization and our algorithm is thoroughly evaluated by conducting multiple experiments. The computational efficiency of the algorithm is beyond the scope of this paper. We rather focus on the methodology. The proposed method does not require any knowledge about the external orientation of the sensors at the time of acquisition, their position and orientation are estimated by the algorithm, the only requirement is that the two clouds share a tolerable overlap.

The experimental is set up as shown in Fig. 4. The centre of the two sensors; Velodyne HDL32 and that of the Leica P20 are perfectly superimposed with the help of the STANLEY Cubix cross line laser. The velodyne is then physically displaced and rotated by known translations and rotations from the graduated set up in order to perturb the 6 degrees of freedom (dof) transformation. Data acquisition is then performed under different scenarios in order to exert our CICP algorithm. Table 1 below summarizes the various experiments performed in a controlled environment. For each experiment, CICP is initialized at Identity, i.e.  $x = [0, 0, 0, 0, 0, 0]$ .

---

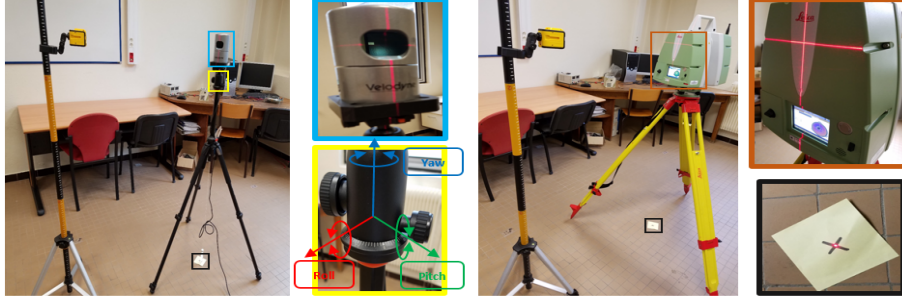
**Algorithm 1: CICP Algorithm.**

---

**Input:** targetCloud, sourceCloud ; voxelSize,  $\hat{T}$   
**Output:** Optimal  $T$

```
1 Initialize: NormalXYZ T_normals, S_normals; PointXYZ T_match, S_match
2 begin
3   T_normals = normalComputing (targetCloud)
4   S_normals = normalComputing (sourceCloud)
5   T_match = normalClustering (targetCloud, T_normals, voxelSize)
6   while ( $iteration < iter\_max \vee ||x|| > \epsilon$ ) do
7     sourceCloud = transform (sourceCloud, S_normals,  $\hat{T}$ )
8     S_match = normalClustering (sourceCloud, S_normals, voxelSize)
9     EstablishCorrespondences (T_match, S_match)
10    distanceRejection (distThreshold)
11    compute the Jacobian  $\mathbf{J}$ 
12    compute the error vector  $e(x)$  (3)
13    compute the increment  $x$  (4)
14    update the pose  $\mathbf{T}$  (5)
15    iteration  $\leftarrow$  iteration + 1
16  end
17  return  $T$ 
18 end
```

---



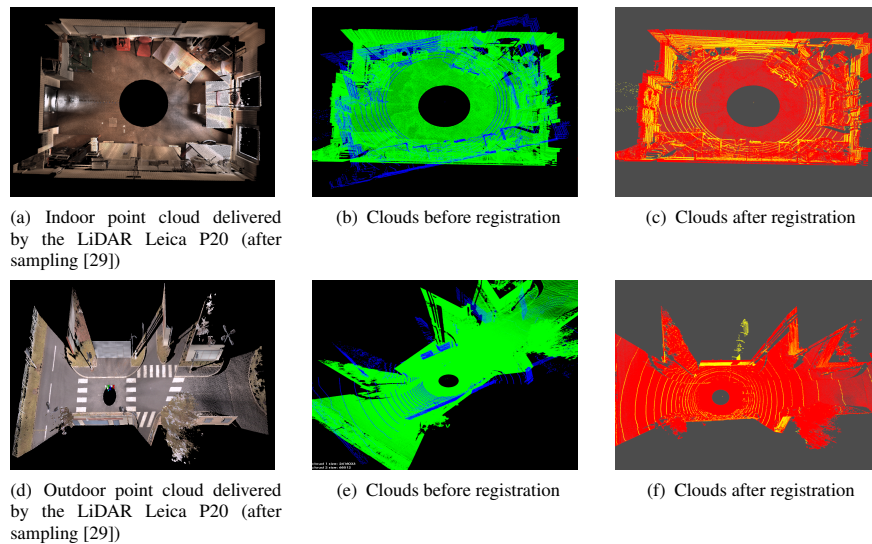
**Figure 4.** Experimental set up for data collection from Velodyne HDL32 (left) and Leica P20 (right) sensors.

#### 4.1 Dense-Sparse Registration with CICP

Two clouds are acquired with different sensors; the denser cloud produced by a 3D LiDAR Leica P20 laser scanner and the sparser cloud with an HDL-32E Velodyne LiDAR sensor. A Leica P20 generates very detailed and dense point clouds. Depending on the resolution chosen during the scanning process, these clouds can exceed 100 millions of points for a single scan. All computations are performed on a laptop with the following specifications; Intel Core i74800MQ processor, 2.7 GHz, and 32 GB of RAM. For reasons of computational resources, we perform a sampling process using [29] in order to reduce the number of points to the order of few millions without losing use-

ful information. Figures 5(a), 5(d) illustrate the output of the sampling process with 986344 and 2732783 points for the office and PAVIN<sup>5</sup> scenes, respectively. On the other hand, the HDL-32E Velodyne sensor generates sparse point clouds that do not exceed 70000 points. This represents a ratio of 14 times between the two clouds from the first environment and a ratio of 40 times, for clouds of the second environment.

Fig. 5 shows the registration process of such point clouds using the CICP method. On the left, the green cloud is from a LiDAR Leica P20 and the blue cloud is from an HDL32-E Velodyne. The corresponding results are shown on the right.



**Figure 5.** Registration results with CICP algorithm.

In order to verify the convergence of the optimization, a comparison between the two clouds at the start and the end of the registration process is recorded together with the convergence profile obtained from the evolution of the RMSE error as a function of the number of iterations to convergence (see Fig. 6). An exit loop condition is imposed on the translation ( $10^{-3}$ ) and rotation rates ( $10^{-4}$ ). We run the algorithm for several indoor and outdoor scenes, with different viewpoints, as depicted in Table 1 and shown in Fig. 6. Each experiment is performed more than 20 times. For instance, for the experiment 1 (Expt\_1), the displacement between the two clouds is  $[150, 170, 35, 5, 0, 0]$ , where the first three values correspond to the translation in millimeters and the last three to the rotation in degrees. As for the fourth experiment, the displacement is  $[65, 45, 200, 0, 7, 5]$ , which took 34 iterations for the algorithm to converge. From this table, three elements are identified that influence the registration results; inter-frame displacement, difference in density between the two clouds, as well as the nature of

<sup>5</sup> PAVIN: <http://www.institutpascal.uca.fr/index.php/en/the-institut-pascal/equipments>

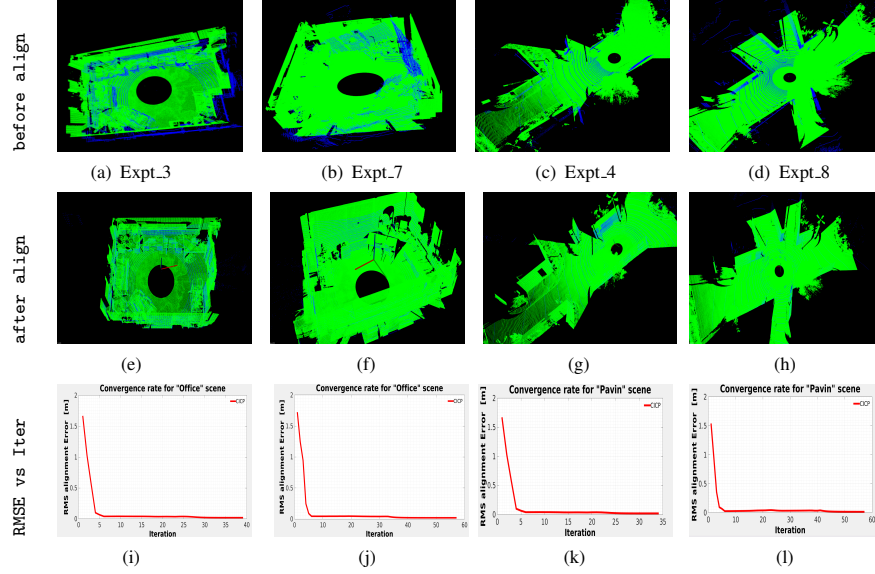


**Table 1.** CICIP registration applied to mainly two compiled data sets; OFFICE and PAVIN. The resolutions of corresponding dense and sparse cloud are given along with the initial physical measured transformation from our set up given by the first row of each experiment, whilst the second row depicts the results output by our algorithm. Convergence is evaluated from the RMSE and the number of iterations required for full registration.

Expt	Envir -onment	# dense cloud	# sparse cloud	$t_x$ (mm)	$t_y$ (mm)	$t_z$ (mm)	$\theta_x$ ( $^\circ$ )	$\theta_y$ ( $^\circ$ )	$\theta_z$ ( $^\circ$ )	RMSE (m)	# iter.
1	Office 1	411924	69952	150.0 155.0	170.0 172.9	35.0 34.8	5.0 4.7	0.0 0.4	0.0 0.1	- 0.0198	- 40
2	PAVIN 1	665260	67488	0.0 0.2	30.0 29.8	200.0 191.9	5.0 4.8	5.0 4.8	0.0 0.1	- 0.0188	- 36
3	Office 2	986344	69984	350.0 357.3	350.0 342.8	0.0 0.3	0.0 0.3	0.0 0.1	0.0 0.8	- 0.0199	- 39
4	PAVIN 2	1364245	67768	65.0 64.1	45.0 45.7	200.0 200.9	0.0 0.3	7.0 6.9	5.0 4.9	- 0.0184	- 34
5	Office 3	2550564	69728	20.0 21.7	110.0 111.1	70.0 66.9	10.0 10.1	5.0 4.2	0.0 0.1	- 0.0191	- 39
6	PAVIN 3	3218879	67936	0.0 0.5	0.0 0.1	0.0 0.3	10.0 9.4	10.0 9.8	0.0 0.2	- 0.0184	- 55
7	Office 4	4490859	69996	30.0 27.8	470.0 470.3	300.0 307.3	20.0 19.6	0.0 0.1	0.0 0.1	- 0.0190	- 57
8	PAVIN 4	5025457	67904	50.0 52.1	50.0 48.2	50.0 53.3	5.0 5.3	5.0 7.5	5.0 5.3	- 0.0152	- 56
9	Office 5	7076192	69760	50.0 55.2	50.0 50.6	50.0 53.9	5.0 5.5	5.0 4.2	5.0 4.6	- 0.0190	- 35
10	PAVIN 5	19615433	67488	0.0 0.1	50.0 49.8	200.0 200.3	10.0 9.7	0.0 0.1	5.0 5.2	- 0.0169	- 48

environment (indoor or outdoor). Overall, we would like to highlight the fact that displacement between two viewing angles are quite consequent keeping in mind that dense techniques generally require an inter frame displacement since the cost function is linearized around  $x = 0$ .

Continuing with our discussion on the influence of initial displacement, and let us take the case of experiments Expt\_3 and Expt\_6, which represent a pure translation and a pure rotation, respectively. These two experiments as a sample of several experiments that we carry out, show that generally, the pure rotation requires more energy to reach the convergence with respect to the case of pure translation. Regarding the influence of density, a quick look shows that the denser the clouds becomes, the more the RMSE decreases, leading to better registration. Finally, we observe that CICIP performance depends on the scene. In fact, the impact of scene affects the performance of registration as observed by the difference of the number of iterations required to reach



**Figure 6.** CICIP results applied to different data sets from various environments.

the convergence domain between PAVIN’s and office data sets. It is clear that the indoor environment performs better registration than the outdoor scene. This is possibly caused by the richness in planar regions of the former. It should not be overlooked that the outdoor environment contains a large amount of noise and outliers. This can be seen on Expt\_8 and Expt\_9, in which the initial displacement is the same in both experiments. However, the alignment for the office dataset requires 34 iterations to converge instead of 56 for the PAVIN data set.

For the sake of illustration, we take four experiments arbitrarily (Expt\_3, Expt\_4, Expt\_7, Expt\_8), and show their state before and after registration with their convergence profile in Fig. 6. A closer look to the RMSE curves in the third row of this figure reveals that the residues which are far away are successfully minimized. However, the convergence begins very quickly and then stabilizes for a while before it reaches its minimum. This is mainly due to the fact that there is not a perfect point-to-point equivalence in the two pairing sets. This is quite logical and it can be explained by the large difference in density between the two clouds, the noise, and the clustering defects on the two clouds.

## 4.2 Comparison with Existing Methods

In order to compare our method with the existing state-of-the-art methods, we use implemented routines of PCL [25] library for the NDT algorithm, GICP, point-to-plane ICP and simple ICP for dense methods. For the case of feature-based methods (methods based on features extraction) we also use PCL implementations of SIFT3D and FPFH to extract characteristic points from the two clouds, and use simple ICP to perform matching. The performance of each method is evaluated using three metrics: the

accuracy, the relative translational error and the relative rotational error. The former describes the evolution of the root-mean-square point-to-point distance; this can be expressed mathematically as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \|E_i\|^2} \quad (6)$$

where  $n$  is the number of points and  $E_i$  is the distance error between the source points and its correspondent in the target cloud in each iteration. This can be expressed as follows:

$$E_i = \sum_{i=0}^m p_i - q_i \quad (7)$$

where  $m$  is the total number of points in the sparse cloud.  $p_i$  and  $q_i$  which represent two points of the source and target cloud, respectively.

The second metric is the Relative Translational Error (RTE), which measures the translation gap between the ground truth ( $t_{GT}$ ) and the estimated ( $t_E$ ) translation vectors.

$$RTE = \|t_{GT} - t_E\|_2 \quad (8)$$

The Relative Rotational Error (RRE) is the sum of the absolute differences of the three Euler angles, calculated from the two rotation matrices  $R_{GT}$  (ground truth rotation matrix) and  $R_E$  (estimated rotation matrix). RRE is calculated by the equation 9 as:

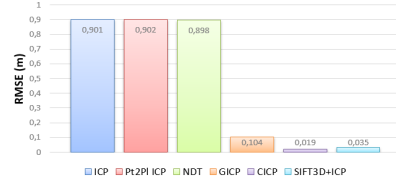
$$RRE = |Roll(R_{GT}^{-1}R_E)| + |Pitch(R_{GT}^{-1}R_E)| + |Yaw(R_{GT}^{-1}R_E)| \quad (9)$$

Table 2 presents the results gathered in processing two indoor and outdoor scenes with the state-of-the-art methods. Bold values show the best result. Quantitatively, the RMSE value of the indoor scene reaches 6 cm in the case of point-to-point, 6.2 cm point-to-plane ICP, 6.3 cm for the NDT and the GICP, more than 5 cm for SIFT3D and less than 3 cm for the proposed method. The maximum number of iterations for each test is fixed at 500 beyond which the algorithm is considered as not having converged if it reaches that ceiling, as is the case of the FPFH method.

Figure 7 shows comparison of convergences between different registration methods. Again CICP outperforms the state of the art. In addition to that, it is shown that CICP is robust against scene variation.

**Table 2.** Comparison with the state-of-the-art methods.

		Dense				Feature-based		
		ICP	pt2pl ICP	NDT	GICP	CICP	SIFT 3D+ICP	FPFH +ICP
Office [m, m, m, °, °, °] [0, 0.5, 0.5, 20, 0, 10]	RMSE (m)	0.0602	0.0620	0.0636	0.0636	<b>0.0299</b>	0.0516	failed
	RTE (m)	0.2811	0.2482	0.2019	0.2178	<b>0.0169</b>	0.0191	failed
	RRE (°)	1.8476	0.5243	1.0517	0.0978	<b>0.0144</b>	0.7660	failed
PAVIN [m, m, m, °, °, °] [0, 0.5, 0.3, 0, 0, 10]	RMSE (m)	0.0836	0.0804	0.0824	0.0860	<b>0.0347</b>	0.0678	failed
	RTE (m)	0.0365	0.0253	0.0315	0.0642	<b>0.0092</b>	0.021	failed
	RRE (°)	0.1919	0.1747	0.2048	0.3211	<b>0.1198</b>	0.1989	failed

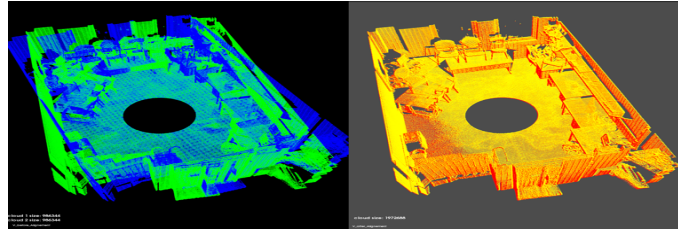


(a) Comparison of RMSE results of the different registration methods



(b) Comparison of number of iterations achieved at convergence of different registration methods

**Figure 7.** Convergence comparison between different registration methods.



**Figure 8.** Registration of two dense clouds of indoor scene captured by Leica P20 sensor.

### 4.3 Dense-to-Dense Data

Figure 8 shows the state of the two dense clouds before and after the registration. Despite the large number of points, the final result is correctly aligned. Here is another benefit of our approach, the fact of not considering the entire set of points for matching, but only a collected set of points from each local surface, which improves the convergence speed.

## 5 Conclusion

In this paper, a novel selection technique is introduced on the forefront of an ICP framework. Throughout the experimental section, the performance of the algorithm has been demonstrated where otherwise classical state-of-the-art techniques fail or perform poorly. The main highlight of this work is the way 3D surfaces are segmented in a point cloud representation using voxelization and clustering approaches. The advantages are multifold; both sparse and dense point clouds are subsampled by maintaining the geometry of the surface. Moreover, better point estimates obtained are used for later stages of matching and registration. In so doing, the convergence domain of the cost function is greatly improved leading to faster convergence of the algorithm where again, classical techniques fail. Additionally, our method of normal based segmentation not only improves on the weakness of the heterogeneous problem of sparse to dense registration but also deals with sensor noise leading to noisy normal extraction. Finally, our CICP approach improves on the precision of registration and outperforms the state of the art. This work can also be viewed as a direct solution to localization problems of a mobile

robot in an *a priori* map constructed with different depth sensors. Further work includes an extension of our approach to vast scale outdoor mapping and localization as well as optimizing the computational time on the CPU for real time applications.

## References

1. Agamennoni, G., Fontana, S., Siegwart, R.Y., Sorrenti, D.G.: Point Clouds Registration with Probabilistic Data Association. In: IEEE/RSJ Int. Conf. on Intel. Robots and Systems (IROS). pp. 4092–4098 (2016)
2. Aldoma, A., Vincze, M., Blodow, N., Gossow, D., Gedikli, S., Rusu, R.B., Bradski, G.: CAD-model recognition and 6DOF pose estimation using 3D cues. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE Int. Conf. on. pp. 585–592. IEEE (2011)
3. Arthur, D., Vassilvitskii, S.: K-means++: The Advantages of Careful Seeding. In: Proc. of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1027–1035. SODA '07, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2007)
4. Besl, P.J., McKay, N.D.: A Method for Registration of 3-D Shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14(2), 239–256 (1992)
5. Blanco, J.L.: A tutorial on se(3) transformation parameterizations and on-manifold optimization. Tech. rep., University of Malaga (2010)
6. 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 Transactions on Robotics 32(6), 13091332 (2016)
7. Caselitz, T., Steder, B., Ruhnke, M., Burgard, W.: Monocular Camera Localization in 3D LiDAR Maps. In: IEEE/RSJ Int. Conf. on Intel. Robots and Systems (IROS). pp. 1–6 (2016)
8. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. In: IEEE Int. Conf. on Robotics and Automation. pp. 2724–2729 (1991)
9. Costa, C.M., Sobreira, H.M., Sousa, A.J., Veiga, G.M.: Robust 3/6 DoF self-localization system with selective map update for mobile robot platforms. Robotics and Autonomous Systems 76(C), 113–140 (2016)
10. Das, A., Diu, M., Mathew, N., Scharfenberger, C., Servos, J., Wong, A., Zelek, J.S., Clausi, D.A., Waslander, S.L.: Mapping, planning, and sample detection strategies for autonomous exploration. Journal of Field Robotics 31(1), 75–106 (2014)
11. Feng, Y., Schlichting, A., Brenner, C.: 3D feature point extraction from LiDAR data using a neural network. Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives 41 (2016) 41, 563–569 (2016)
12. Hänsch, R., Weber, T., Hellwich, O.: Comparison of 3D interest point detectors and descriptors for point cloud fusion. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences II-3(September), 57–64 (2014)
13. Holz, D., Ichim, A.E., Tombari, F., Rusu, R.B., Behnke, S.: Registration with the Point Cloud Library PCL. IEEE Robotics & Automation Magazine 22(4), 1–13 (2015)
14. Jolliffe, I.T.: Principal Component Analysis for Special Types of Data, pp. 199–222. Springer New York, New York, NY (1986)
15. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of the Seventh IEEE Int. Conf. on Computer Vision. pp. 1150–1157 (1999)
16. Ma, Y., Soatto, S., Koščeká, J., Sastry, S.S.: An Invitation to 3-D Vision. Springer (2004)
17. Maddern, W., Newman, P.: Real-time probabilistic fusion of sparse 3d lidar and dense stereo. In: Intel. Robots and Systems (IROS), 2016 IEEE/RSJ Int. Conf. on. pp. 2181–2188. IEEE (2016)
18. Magnusson, M., Lilienthal, A., Duckett, T.: Scan registration for autonomous mining vehicles using 3D-NDT. Journal of Field Robotics 24(10), 803–827 (2007)

19. Marani, R., Reno, V., Nitti, M., D'Orazio, T., Stella, E.: A modified iterative closest point algorithm for 3D point cloud registration. *Computer-Aided Civil and Infrastructure Engineering* 31(7), 515–534 (2016)
20. Nieto, J., Bailey, T., Nebot, E.: Scan-SLAM: Combining EKF-SLAM and scan correlation. *Springer Tracts in Advanced Robotics* 25, 167–178 (2006)
21. Pomerleau, F., Colas, F., Siegwart, R.: A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics* 4(1-104), 1–104 (2015)
22. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: *Proc. of Int. Conf. on 3-D Digital Imaging and Modeling, 3DIM*. pp. 145–152 (2001)
23. Rusu, R.B., Blodow, N., Beetz, M.: Fast Point Feature Histograms (FPFH) for 3D registration. In: *IEEE Int. Conf. on Robotics and Automation*. pp. 3212–3217 (2009)
24. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3d recognition and pose using the view-point feature histogram. In: *Intel. Robots and Systems (IROS), 2010 IEEE/RSJ Int. Conf. on*. pp. 2155–2162. IEEE (2010)
25. Rusu, R.B., Cousins, S.: 3D is here: point cloud library. In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. pp. 1 – 4 (2011), <http://pointclouds.org/>
26. Segal, A., Haehnel, D., Thrun, S.: Generalized-ICP. In: *Robotics: Science and Systems* (2009)
27. Serafin, J., Grisetti, G.: NICP: Dense normal based point cloud registration. In: *IEEE Int. Conf. on Intel. Robots and Systems*. vol. 2015-Decem, pp. 742–749 (2015)
28. Serafin, J., Olson, E., Grisetti, G.: Fast and Robust 3D Feature Extraction from Sparse Point Clouds. In: *IEEE/RSJ Int. Conf. on Intel. Robots and Systems (IROS)*. pp. 4105–4112 (2016)
29. Tazir, M.L., Checchin, P., Trassoudaine, L.: Color-based 3D Point Cloud Reduction. In: *the 14th Int. Conf. on Control, Automation, Robotics and Vision, ICARCV*. pp. 1–7 (2016)
30. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63(2), 411–423 (2001)
31. Velas, M., Spanel, M., Herout, A.: Collar Line Segments for fast odometry estimation from Velodyne point clouds. In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. pp. 4486–4495 (2016)
32. Wiemann, T., Mrozinski, M., Feldschnieders, D., Lingemann, K., Hertzberg, J.: Data Handling in Large-Scale Surface Reconstruction. In: *13th Int. Conf. on Intel. Autonomous Systems*. pp. 1–12 (2014)
33. Wolcott, R.W., Eustice, R.M.: Visual localization within LIDAR maps for automated urban driving. In: *Intel. Robots and Systems (IROS 2014), 2014 IEEE/RSJ Int. Conf. on*. pp. 176–183. IEEE (2014)
34. Yang, B., Dong, Z., Liang, F., Liu, Y.: Automatic registration of large-scale urban scene point clouds based on semantic feature points. *ISPRS Journal of Photogrammetry and Remote Sensing* 113, 43–58 (2016)
35. Yang, J., Li, H., Jia, Y.: Go-icp: Solving 3d registration efficiently and globally optimally. In: *Proc. of the IEEE Int. Conf. on Computer Vision*. pp. 1457–1464 (2013)