Robotica (2016) volume 34, pp. 1630–1658. © Cambridge University Press 2014
doi:10.1017/S0263574714002483

Robust and fast 3-D scan registration using normal distributions transform with supervoxel segmentation

Ji W. Kim* and Beom H. Lee

Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea

(Accepted September 23, 2014. First published online: October 29, 2014)

SUMMARY

This paper presents what is termed as the supervoxel normal distributions transform (SV-NDT), a novel three-dimensional (3-D) registration algorithm which improves the performance of the three-dimensional normal distributions transform (3-D NDT) significantly. The 3-D NDT partitions a model scan using a 3-D regular grid. Generating normal distributions using the 3-D regular grid causes considerable information loss because the 3-D regular grid does not use any information pertaining to the local surface structures of the model scan. The best type of surface (the constituent unit of each scan) for modeling with one normal distribution is known to be the plane. The SV-NDT reduces the loss of information using a supervoxel-generating algorithm at the partitioning stage. In addition, it uses the information of the local surface structures from the data scan by replacing the Euclidean distance with a function that uses local geometries as well as the Euclidean distance when each point in the data scan is matched to the corresponding normal distribution. Experiments demonstrate that the use of the supervoxel-generating algorithm increases the modeling accuracy of the normal distributions and that the proposed 3-D registration algorithm outperforms the 3-D NDT and other widely used 3-D registration algorithms in terms of robustness and speed on both synthetic and real-world datasets. Additionally, the effect of changing the function to create correspondences is also verified.

KEYWORDS: 3-D Scan registration; Normal distributions transform; SLAM; Mobile robotics.

1. Introduction

In mobile robotics, scan registration algorithms are essential to guarantee the performance of various applications, such as simultaneous localization and mapping (SLAM), navigation, exploration, rescue, and surveillance. Many SLAM techniques, in particular, use scan registration methods to estimate the relative transformation between two scans. Scan registration algorithms allow robots to collect more information about the surrounding environment by integrating two scans taken at different times or places.

Scan registration algorithms can be broadly categorized into local methods and global methods. Local methods conduct a scan registration by iteratively optimizing a cost function which represents the registration error between two scans. Given that the cost function has local minima, in most cases, the results from local methods depend on the initial transformation. The initial transformation can be obtained by odometry, an inertial measurement unit (IMU), or a GPS if this type of system is available. If the initial transformation is sufficiently close to the ground truth, local methods can estimate the relative transformation finely compared to global methods. There are various algorithms used with local methods, such as the iterative closest point (ICP),[1] the normal distributions transform (NDT),[2] and the polar scan matching (PSM).[3] Global methods undertake scan registration with the distinct local geometrical features of each scan. They vary depending on their means of extracting

* Corresponding author. E-mail: kjw116@snu.ac.kr

features such as the Hough transform,[4] the fast point feature histogram (FPFH),[5] the phase only matched filtering (POMF),[6] and other methods.

One of the most popular scan registration algorithms is the ICP algorithm. The ICP algorithm is a point-to-point algorithm that estimates the optimal transformation to overlap two scans, a model and a data scan, by iteratively minimizing the sum of the squared Euclidean distances between the corresponding points. The ICP algorithm regards the closest points in different scans as corresponding points. Because a closed-form solution exists for optimizing the sum of the squared Euclidean distances between associated pairs, the ICP is easily implemented. Although the nearest neighbor search is a bottleneck when using the ICP due to the high computational cost, using the k-d tree[7] or approximate k-d tree[8] can mitigate this problem. However, the assumption that the closest points in different scans are the corresponding points is satisfied well only when the relative rotation difference between the two scans is small. Given a large relative rotation difference, points that are far from the sensor move far away, with many of these points not associated correctly as a result. For this reason, iterative dual correspondence (IDC)[9] generates corresponding points for rotation and translation separately and alternatively minimizes the sum of the squared Euclidean distances. Metric-based ICP (MbICP)[10] establishes correspondences between two scans with a new metric which takes into account rotation as well as translation. Every scan is composed of the closest surfaces in the surroundings to the sensor. Generalized-ICP (G-ICP)[11] uses local surface structures while retaining the simplicity of the ICP. The G-ICP models the vicinity of each point as a locally planar surface by means of a covariance matrix and applies this information to the cost function to decrease the effect of incorrectly associated points.

Another algorithm for scan registration is the NDT algorithm. The NDT algorithm was initially proposed as a two-dimensional scan registration algorithm[2]and was later extended to three dimensions.[12] The three-dimensional (3-D) NDT algorithm is not a point-to-point method which performs registration between two scans directly but is instead a point-to-distribution method that carries out registration between the data scan and a set of distributions generated from the model scan. Because the NDT does not need to search for the closest points or store the raw data from the model scan, it has low computational complexity and can greatly reduce the amount of memory required. In addition, the gradient vector and the Hessian matrix of its score function have analytic forms, allowing the simple use of standard non-linear optimization methods to estimate the optimal transformation.

However, the use of a regular grid by the NDT causes several fundamental problems. The first is the discontinuity of the score function. When a data scan point passes one of its cell boundaries, the value of the score function jumps. Because this can cause a problem, a method using tri-linear interpolation between distributions within neighboring voxels, which relieves the effects of discontinuities,[13] was proposed. Furthermore, an alternative method was suggested that modifies the score function so that it becomes continuous.[14] Because this method employs greedy clustering to partition the scan, there are few distributions. Thus, the modified score function includes the scores of all of the normal distributions for each point in the data scan. In addition, a distribution-to-distribution registration approach was proposed which transforms the data scan as well as the model scan into normal distributions.[15] The second fundamental problem is that the registration performance relies on the cell size of the regular grid. In one study,[16] the cell size of each cell was made to vary with the distance from the sensor in an effort to solve this problem. In addition, the use of multi-layered NDT (ML-NDT)[17] changes the cell size from large to small during the iterative optimization process. A more serious problem, though, is that a normal distribution in each voxel does not represent the local structure of the point subset within the voxel accurately because the regular grid does not take into account the structures of the model scan. When part of the scan data, i.e., that composed of surfaces, is modeled by a normal distribution, the shape of the point subset which minimizes information loss is a plane. Thus, in order to utilize the merits of the NDT and minimize the loss of information, the model scan should be divided into locally planar surfaces by means of segmentation techniques.

Segmentation techniques for a 3-D point cloud can be divided into object-based segmentation techniques, which separate a point set based on objects, and over-segmentation techniques, which partition more finely into point subsets by gathering points that have locally similar geometries. Object-based segmentation methods mostly consist of two stages. The first stage splits the point cloud into ground points and non-ground points, and the second stage generates objects by grouping the non-ground points. In another study,[18] Gaussian process incremental sample consensus

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

(GP-INSAC) is used at the first stage, and the non-ground points are clustered according to the principle of local voxel adjacency. To increase the speed of the ground segmentation process, the point cloud is divided into several sectors and one-dimensional GP-INSAC is then carried out for each sector.[19] Another approach with which to extract ground points is a graph-based method.[20] In that study,[20] segmentation is performed by means of local convexity. The segment whose normal vector has the largest z-component is regarded as the ground, and the other segments are considered as objects. For urban environments, random sample consensus (RANSAC) and a Kalman filter are used to extract the road and street furniture.[21]

Over-segmentation techniques are used mainly as a preprocessing stage for image segmentation in computer vision. Well-known image segmentation approaches such as the Markov random field (MRF) and the conditional random field (CRF) usually estimate the class of each pixel. However, because most present-day images have a great number of pixels, a considerable amount of computing time is required. To reduce the number of regions to be estimated, over-segmentation techniques such as the use of the superpixels, which groups similar pixels into one region, are used. Recently, this idea was extended to three dimensions, and some algorithms were proposed in this context. In one such study,[22] non-ground points are partitioned into super-segments according to connectivity and similarity in the normal direction. Another approach constructs evenly spaced supervoxels by clustering a voxel-cloud based on the similarity of local geometries.[23] Furthermore, over-segmentation is performed by spheres of different sizes that are determined by the local curvatures and densities.[24]

In this paper, supervoxel-NDT (SV-NDT) is proposed to solve the problem in which the normal distributions generated by a regular grid do not represent the local surface structures of the model scan accurately. SV-NDT exploits the 3-D supervoxel segmentation technique to use local surface structures to partition the model scan. In addition, the criterion of matching each point in the data scan to the corresponding distribution is modified using the local geometries of the data scan. The SV-NDT is evaluated and compared to the 3-D NDT and to several 3-D scan registration algorithms. In addition, the supervoxel-generating algorithm and the criterion for devising correspondences are evaluated.

This paper is organized as follows. The problem formulation for 3-D scan registration on the basis of the 3-D NDT is described in the next section. In Sections 3 and 4, the method of supervoxel generation and the proposed method are presented in sequence. Section 5 evaluates the performance of each element of the SV-NDT and the SV-NDT compared to other widely used 3-D registration algorithms. Finally, this paper is concluded in Section 6.

2. Problem Formulation

The goal of 3-D scan registration is to estimate the optimal transformation between two scans. The model and the data scan are denoted as $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_{N_Y}\}$ and $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_X}\}$ respectively, and the set of 3-D rigid transformations is $\mathbb{SE}(3)$. A 3-D rigid transformation T is composed of a translation vector $\mathbf{t} \in \mathbb{R}^3$ and a rotation matrix $\mathbf{R} \in \mathbb{SO}(3)$. A 3-D rotation matrix can be represented in various ways, such as a unit quaternion and the Euler angles. In our approach, rotation matrices are represented by the roll, pitch, and yaw convention, which is a sequence of three basic rotations about the fixed axes x, y, and z. Therefore, the transformation parameter $\mathbf{p}$ can be represented by $[t_x t_y t_z \theta_x \theta_y \theta_z]^T$. If a data scan point $\mathbf{x}_n$ is transformed by $T(\mathbf{p})$, it can be denoted as $T(\mathbf{p}) \cdot \mathbf{x}_n$, and the transformed point $\mathbf{x}'_n$ can be written as

$$\mathbf{x}'_n = T(\mathbf{p}) \cdot \mathbf{x}_n = \mathbf{R}_z(\theta_z)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)\mathbf{x}_n + \mathbf{t}, \tag{1}$$

where $\mathbf{R}_x(\theta_x)$, $\mathbf{R}_y(\theta_y)$, and $\mathbf{R}_z(\theta_z)$ are the basic 3-D rotation matrices for rotations about each of the axes by angles θ_x, θ_y, and θ_z, respectively. In the same manner, the transformed data scan X' with the transformation parameter $\mathbf{p}$ is denoted by

$$X' = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_{N_X}\} = \{T(\mathbf{p}) \cdot \mathbf{x}_1, \ldots, T(\mathbf{p}) \cdot \mathbf{x}_{N_X}\}. \tag{2}$$

The 3-D NDT performs the registration between the data scan and the normal distributions generated from the model scan using a 3-D regular grid. To be more concrete from a statistical point of view, the 3-D NDT adopts the model scan as Gaussian mixture model (GMM) which can

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

model an arbitrarily smooth probability density function and estimates the optimal transformation by maximizing the likelihood function of the transformed data scan.[25] The likelihood function of a point $\mathbf{x}$, is defined as

$$p(\mathbf{x}|GMM_{\text{model}}) = \sum_{i=1}^{N} w_i \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{\det \boldsymbol{\Sigma}_i}} \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}{2}\right), \tag{3}$$

where GMM_{model} is a set of parameters of the GMM for the model scan, N is the number of components of the GMM, and w_i, $\boldsymbol{\mu}_i$, and $\boldsymbol{\Sigma}_i$ are the weight, the mean vector, and the covariance matrix of the ith component of the GMM, respectively. The likelihood function of the transformed data scan X' can then be represented by a function of the transformation parameter $\mathbf{p}$:

$$\Psi(\mathbf{p}) = p\left(X'|GMM_{\text{model}}\right) = \prod_{n=1}^{N_X} p\left(\mathbf{x}'_n|GMM_{\text{model}}\right), \tag{4}$$

where $\mathbf{x}'_n$ is the nth point in the data scan and N_X is the number of points in the data scan. The optimal transformation parameter can be estimated by means of the maximum likelihood estimation (MLE) for Eq. (4). This is equivalent to minimizing the negative log-likelihood which is given by

$$\bar{\Psi}(\mathbf{p}) = -\log \Psi(\mathbf{p}) = -\sum_{n=1}^{N_X} \log\ p(\mathbf{x}'_n|GMM_{\text{model}}). \tag{5}$$

In order to perform the MLE, GMM_{model} needs to be obtained from the model scan. Usually, when the observed data and the number of components of a GMM are given, the set of parameters GMM_{model} is computed by means of an expectation-maximization (EM) algorithm. However, GMM_{model} is obtained without complicated calculations by several simplifications in the 3-D NDT. First, the model scan Y is divided into the point subsets $Y_n = \{\mathbf{y} \in Y : \mathbf{y} \in v_n\}$ contained within the voxel v_n using a 3-D regular grid. The mean vector $\boldsymbol{\mu}_n$ and the covariance matrix $\boldsymbol{\Sigma}_n$ of each voxel v_n are then calculated by

$$\boldsymbol{\mu}_n = \frac{1}{|Y_n|} \sum_{k:\mathbf{y}_k \in Y_n} \mathbf{y}_k, \tag{6}$$

$$\boldsymbol{\Sigma}_n = \frac{1}{|Y_n| - 1} \sum_{k:\mathbf{y}_k \in Y_n} (\mathbf{y}_k - \boldsymbol{\mu}_n)(\mathbf{y}_k - \boldsymbol{\mu}_n)^T. \tag{7}$$

Because a covariance matrix of fewer than four points is always singular, voxels which have fewer than four points are regarded as unoccupied. The normal distribution in each occupied voxel models the local geometry within each voxel and is considered as a component of the GMM. The weight w_i of each component is set to $1/N_X$. However, for the sake of simplicity, the likelihood of each $\mathbf{x}'_n$ takes into account only the closest component of the GMM from $\mathbf{x}'_n$. The simplified negative log-likelihood function can be written as

$$\tilde{\Psi}(\mathbf{p}) = -\sum_{n=1}^{N_X} \log\ p(\mathbf{x}'_n|\boldsymbol{\mu}_{c_n}, \boldsymbol{\Sigma}_{c_n}) = \sum_{n=1}^{N_X} \frac{(\mathbf{x}'_n - \boldsymbol{\mu}_{c_n})^T \boldsymbol{\Sigma}_{c_n}^{-1}(\mathbf{x}'_n - \boldsymbol{\mu}_{c_n})}{2} + const, \tag{8}$$

where c_n is the index of the closest component of the GMM from $\mathbf{x}'_n$. Because Eq. (8) is proportional to the sum of the squared distances between each transformed point $\mathbf{x}'_n$ in the data scan and the mean vector $\boldsymbol{\mu}_{c_n}$ of the corresponding distribution of $\mathbf{x}'_n$, it is sensitive to outliers. This problem is alleviated by substituting the likelihood function of each point with a normal-uniform mixture distribution:[26]

$$\hat{p}(\mathbf{x}'_n|\boldsymbol{\mu}_{c_n}, \boldsymbol{\Sigma}_{c_n}) = \xi_1 \exp\left(-\frac{(\mathbf{x}'_n - \boldsymbol{\mu}_{c_n})^T \boldsymbol{\Sigma}_{c_n}^{-1}(\mathbf{x}'_n - \boldsymbol{\mu}_{c_n})}{2}\right) + \xi_2 p_o, \tag{9}$$

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

where ξ_1 and ξ_2 are constants such that Eq. (9) integrates over the c_nth voxel to one, and p_o is the expected ratio of the outliers. The negative log-likelihood of Eq. (9) is somewhat complicated and can therefore also be approximated by

$$\begin{aligned} -\log\ \hat{p}\left(\mathbf{x'}_n|\boldsymbol{\mu}_{c_n}, \boldsymbol{\Sigma}_{c_n}\right) &= -\log\left(\xi_1 \exp\left(-\frac{(\mathbf{x'}_n - \boldsymbol{\mu}_{c_n})^T \boldsymbol{\Sigma}_{c_n}^{-1}(\mathbf{x'}_n - \boldsymbol{\mu}_{c_n})}{2}\right) + \xi_2 p_o\right) \\ &\approx -d_1 \exp\left(-d_2 \frac{(\mathbf{x'}_n - \boldsymbol{\mu}_{c_n})^T \boldsymbol{\Sigma}_{c_n}^{-1}(\mathbf{x'}_n - \boldsymbol{\mu}_{c_n})}{2}\right) + d_3, \end{aligned} \tag{10}$$

where d_1, d_2, and d_3 are constants which are determined by ξ_1, ξ_2, and p_o. After dropping the constant d_3, which plays no role in the optimization process, the score of nth data scan point can be defined as

$$s(\mathbf{p}, \mathbf{x}_n, c_n) = d_1 \exp\left(-d_2 \frac{(\mathbf{x'}_n - \boldsymbol{\mu}_{c_n})^T \boldsymbol{\Sigma}_{c_n}^{-1}(\mathbf{x'}_n - \boldsymbol{\mu}_{c_n})}{2}\right). \tag{11}$$

Finally, the cost function of the 3-D NDT, which is the negative of the score function, can be defined as

$$\text{cost}(\mathbf{p}) = -\sum_{n=1}^{N_X} s(\mathbf{p}, \mathbf{x}_n, c_n). \tag{12}$$

The optimal $\mathbf{p}$ can be estimated by minimizing Eq. (12). Given that the first and second derivatives of this cost function have analytic forms, a standard non-linear optimization algorithm can easily be used, such as Newton's method or the Levenberg–Marquardt method. Newton's method is used in our approach.

Newton's method minimizes the cost function by updating arguments iteratively. Each update is performed to optimize the second-order approximation of the cost function; therefore, Newton's step $\Delta\mathbf{p}$ can be calculated by solving the following equation:

$$\mathbf{H}\Delta\mathbf{p} = -\mathbf{g}, \tag{13}$$

where $\mathbf{g}$ and $\mathbf{H}$ are the gradient vector and the Hessian matrix of the cost function at $\mathbf{p}$, respectively. The step size γ is obtained by a line search, after which $\mathbf{p}$ is updated by

$$\mathbf{p} \leftarrow \mathbf{p} + \gamma\Delta\mathbf{p}. \tag{14}$$

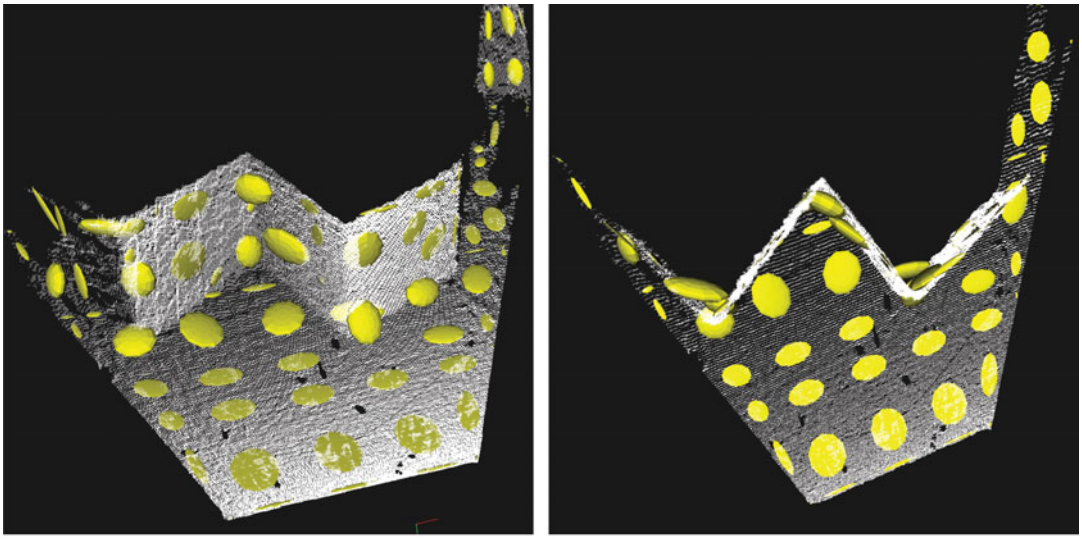
These two steps, computing $\Delta\mathbf{p}$ and γ at $\mathbf{p}$ and updating $\mathbf{p}$, are iterated until $\mathbf{p}$ converges.

3. Supervoxel Generation

3.1. Motivation

The 3-D NDT algorithm does not use the model scan directly but instead employs normal distributions that are generated from the model scan. A mean vector and a covariance matrix can represent all of the characteristics of a normal distribution; in other words, an ellipsoid is able to contain the entire set of information pertaining to a normal distribution. When a surface is modeled as a normal distribution, the lower its curvature is, the less information it loses. A scan is composed of the closest surfaces in each direction from the sensor; therefore, the scan should be partitioned into planar surfaces to reduce the loss of information. If a scan is transformed into normal distributions in the manner of the 3-D NDT, which uses a 3-D regular grid without considering the surface structures of the model scan, there will be some distributions whose mean vectors are not on any surfaces modeled by those distributions, or whose normal vectors, i.e. the eigenvectors corresponding to the minimum eigenvalue of each covariance matrix, are not parallel to any normal directions of surfaces modeled by those

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press



(a) (b)

Fig. 1. (a) Front view. (b) Top view. The normal distributions which employ a 3-D regular grid to transform the model scan in a structured environment from the dataset.[27] Each ellipsoid represents a normal distribution, and its surface is a set of the points whose Mahalanobis distance from the mean is equal to 1. Most of the distributions around the corners cannot model the local surfaces.

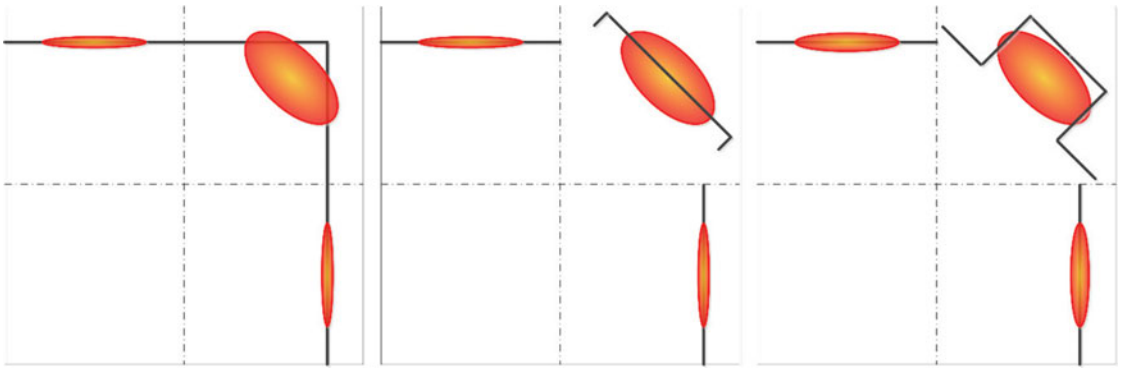


Fig. 2. Several examples of 2-D scans which are transformed into identical normal distributions but with different local geometries.

distributions (Fig. 1). This means that the voxels which have those distributions contain surfaces having different local geometries concurrently, and those normal distributions represent the average local geometry. As a result, the surfaces which belong to such voxels cannot be specified precisely due to the loss of information. Figure 2 shows several examples of two-dimensional scans which are transformed into identical normal distributions but with different local geometries. Decreasing the size of the voxels in order to alleviate this problem increases the computational load owing to the increase in the number of distributions. Moreover, the performance of the 3-D NDT can be diminished due to the increased number of unoccupied voxels. Thus, a segmentation technique which partitions the model scan into planar surfaces using the information of the local surface structures is required.

Researchers have recently proposed several scan registration algorithms which use object-based segmentation methods.[28,29] If a model scan is partitioned by means of object-based segmentation instead of the 3-D regular grid in the 3-D NDT, each object is modeled by one normal distribution. However, when an object has complex structures, it cannot be represented accurately by one normal distribution. As mentioned earlier, in order for the model scan to be modeled by normal distributions accurately, it should be partitioned into locally planar surfaces. To achieve this goal, over-segmentation techniques are required. The segmentation algorithm in one study[24] creates spherical segments with radii determined by local convexities and densities. However, the spherical shape of each segment

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

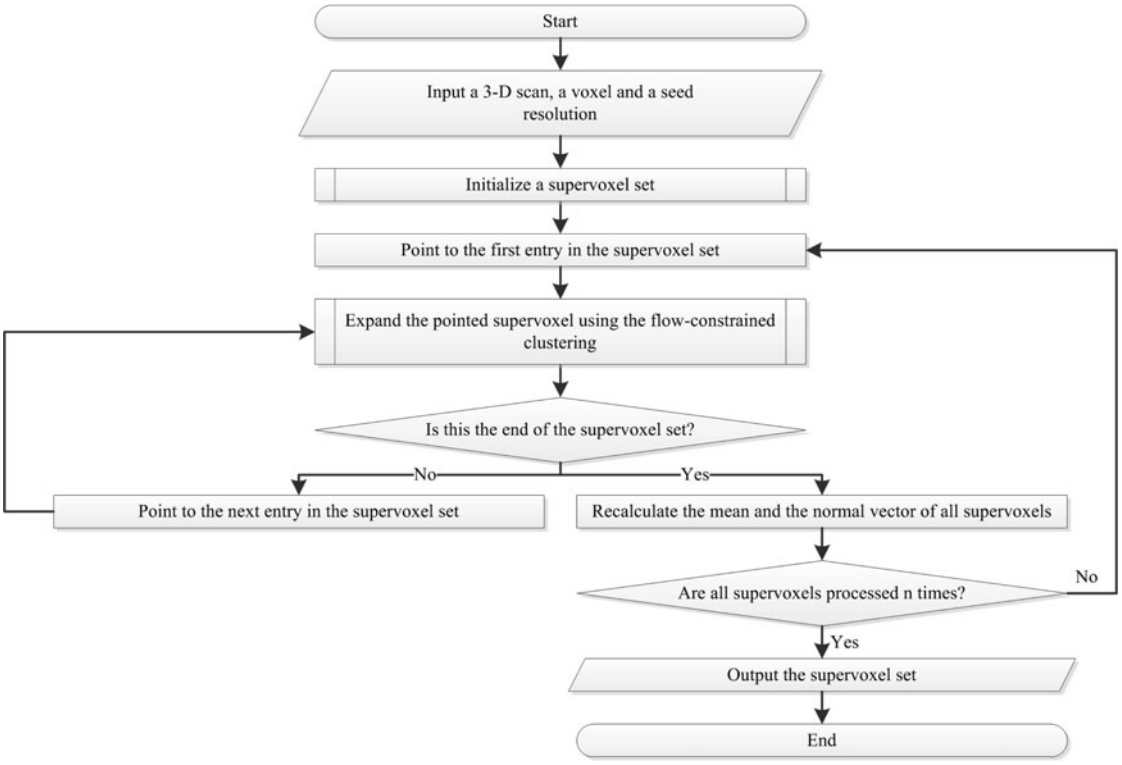


Fig. 3. Flowchart of the supervoxel-generating algorithm.

is not appropriate for representing a local surface. In another study,[22] tiny clusters are constructed using only the Euclidean distances between points, with the clusters then grouped into segments after a comparison of features of the local geometry, such as the connectivity and normal directions. Although this segmentation method can partition the model scan into planar surfaces, it is possible that a segment would be very large, such as the case of a wall. When this segment is modeled by one normal distribution, it is impossible to detect holes, such as windows and open doors. In other work,[23] a voxel-cloud is formed using a 3-D regular grid with small voxel size at the beginning. Then, evenly spaced seed voxels are selected, and each seed voxel is considered as a supervoxel. Finally, flow-constrained clustering which expands each supervoxel by including voxels that have locally similar geometries is performed. The sizes of the supervoxels generated when using this method are limited to a certain upper bound; thus, the supervoxels consist of only local points. In this regard, this 3-D supervoxel-generating algorithm is consistent with the goal of dividing the model scan into locally planar surfaces. Therefore, the model scan is partitioned by the 3-D supervoxel-generating algorithm in our approach.

3.2. Supervoxel-generating algorithm

The 3-D supervoxel-generating algorithm in an aforementioned study[23] which uses voxel cloud connectivity segmentation (VCCS) was proposed to carry out the segmentation of a 3-D colored scan from a RGB-D camera. However, 3-D scans dealt with by our approach are only non-colored; thus, the supervoxel-generating algorithm is modified so that it does not use color. In addition, although the original VCCS uses the FPFH as the local geometric information within each voxel, this information is simplified, taking the form of a normal vector. Along with these simplifications, a flowchart of the method used to generate the supervoxels is shown in Fig. 3. A 3-D scan is transformed into a voxel-cloud with a voxel resolution of R_v, after which a supervoxel set is initialized using seed voxels which are spaced about a seed resolution that is R_s apart. Next, all supervoxels are expanded by flow-constrained clustering with an adjacency graph. After that, the mean and the normal vector of all supervoxels are recalculated. These two steps, an expansion and a recalculation step, are repeated n times, and the supervoxel set is finally obtained. The number of repetitions n is defined as

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

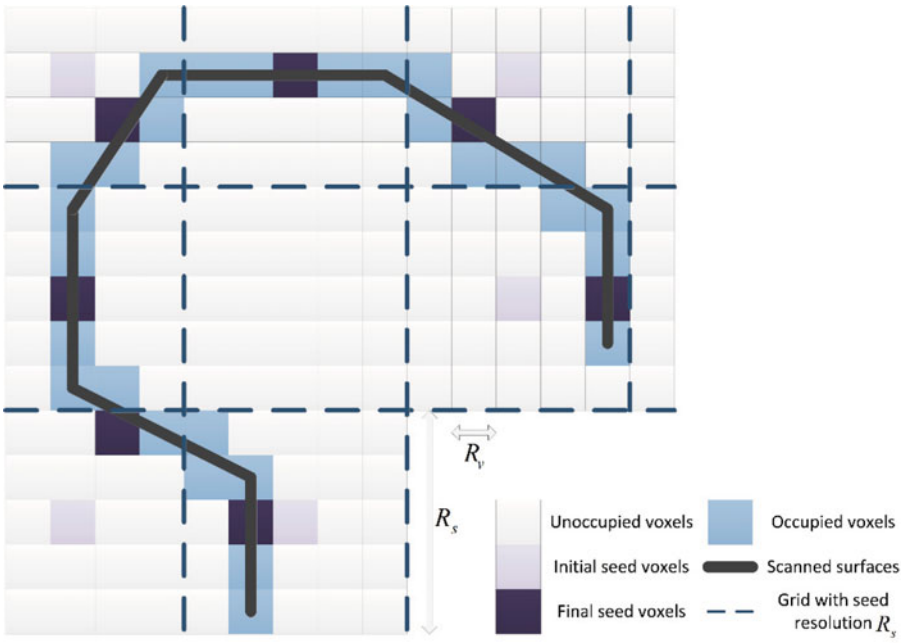


Fig. 4. A 2-D example of generation of a set of seed voxels.

$\lfloor \sqrt{3} R_s / R_v \rfloor$ so that every occupied voxel is contained in one of the supervoxels. In this algorithm, the seed resolution R_s denotes the voxel size of the 3-D regular grid in the 3-D NDT. The voxel resolution R_v should be sufficiently smaller than R_s. R_v is set to $R_s/10$ in our approach. Detailed descriptions of two sub-processes, the initialization of a set of supervoxels and the expansion of each supervoxel by means of flow-constrained clustering, are presented in Subsections 3.2.1 and 3.2.2, respectively.

3.2.1. Initialization of a set of supervoxels. First, a voxel-cloud $V = \{v_1, \ldots, v_{N_V}\}$ is generated from the 3-D scan X and the 3-D regular grid G^v_{grid} with a voxel resolution of R_v. Each v_i is an occupied voxel which contains more than three points and has indices of points contained within it. After that, the data of all voxels v_i in V are initialized. The initialization stage of the voxel data calculates the mean vector $\boldsymbol{\mu}_i$ and normal vector $\mathbf{n}_i$ and sets d_i to zero. d_i is the distance between v_i and the supervoxel containing v_i. Next, the adjacency graph $AG = (U, E)$ of the voxel-cloud V is generated. A set of nodes is denoted as $U = \{1, \ldots, N_V\}$, and each node i corresponds to the ith voxel v_i. The undirected edge set E of AG is defined as follows:

$$E = \{(i, j) | v_j \in V \cap \mathrm{N}_{G^v_{\text{grid}}}(v_i); v_i \in V\}, \tag{15}$$

where $\mathrm{N}_{G^v_{\text{grid}}}(v_i)$ is a set of 26 neighbors of v_i in G^v_{grid}. Then, a set of seed voxels spaced at intervals of approximately the seed resolution R_s, denoted as $V_{\text{seed}} = \{v_{s_1}, \ldots v_{s_{N_S}}\}$, is created. Initially, a new 3-D regular grid G^s_{grid} is generated with the seed resolution of R_s. Next, the initial seed voxels which contain the center of the occupied voxels in G^s_{grid} are selected in G^v_{grid}. The closest occupied voxel in G^v_{grid} from each initial seed voxel is then considered as the final seed voxel. A 2-D example is shown in Fig. 4. After that, N_S supervoxels are created and initialized. The data of the ith supervoxel to initialize are sv_i, M_i, $\mathbf{N}_i$, and Q_i. sv_i is a set of voxels belonging to the ith supervoxel, and M_i and $\mathbf{N}_i$ are respectively a mean vector and a normal vector which are calculated by all points contained within the voxels in sv_i. Q_i is the set of voxels to visit at the current level in the breadth-first search (BFS) of the ith supervoxel. In order to perform flow-constrained clustering, the BFS traverses AG from the node corresponding to the ith seed voxel v_{s_i}. To initialize the clustering process, all instances of sv_i and Q_i are set to $\{v_{s_i}\}$ and the voxels adjacent to v_{s_i} in the AG respectively, and M_i and $\mathbf{N}_i$ are calculated.

3.2.2. Expansion of supervoxels by means of flow-constrained clustering. The flow-constrained clustering stage is based on the BFS of each supervoxel. This clustering stage proceeds in a

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

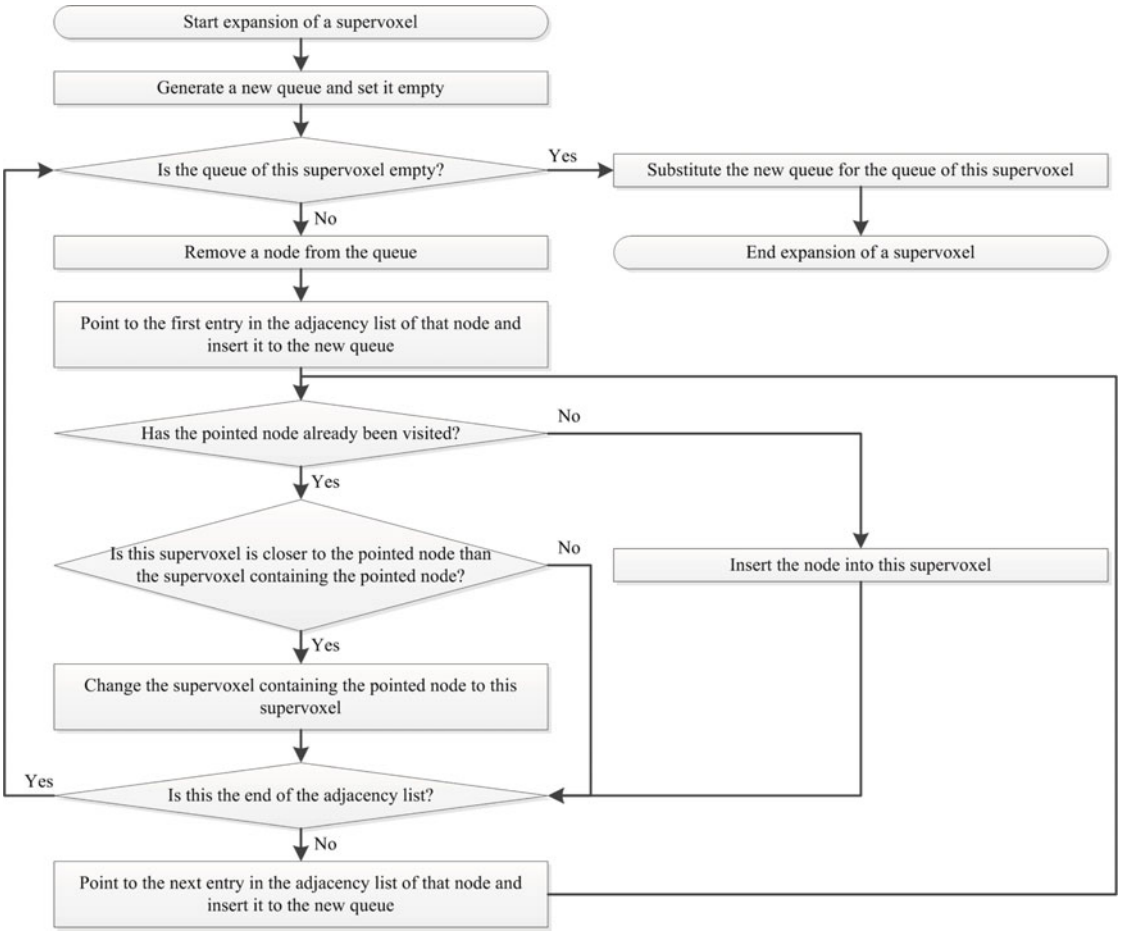


Fig. 5. Flowchart of the expansion of supervoxels using the flow-constrained clustering.

breadth-first fashion, which means that the level to traverse increases after every BFS traverses the same level in turn. A flowchart of this sub-process is shown in Fig. 5. A new queue, $\tilde{Q}_i$, is the set of voxels to visit at the next level in the BFS for the ith supervoxel; it is emptied before each level is traversed. A detailed example of the traversing strategy is shown in Fig. 6. The clustering algorithm visits all of the adjacent voxels v_j of every voxel v in Q_i, and each v_j is dealt with by dividing it into three cases. The first case is one in which v_j does not belong to any supervoxel. In this case, sv_i and $\tilde{Q}_i$ include v_j, and d_j is calculated. The distance function between the ith supervoxel and v_j is defined as

$$D(i,j) = \frac{\left\| \mathbf{M}_i - \boldsymbol{\mu}_j \right\|}{R_s} + \left(1 - \mathbf{N}_i \cdot \mathbf{n}_j\right). \tag{16}$$

The second case is that v_j already belongs to the i'th supervoxel, but the ith supervoxel is closer to v_j than the i'th supervoxel. In this case, v_j moves from the i'th supervoxel to the ith supervoxel with the update of sv_i, $\tilde{Q}_i$, $sv_{i'}$, $\tilde{Q}_{i'}$, and d_j. The last case is the final case, during which all of the clustering data remain unchanged. After visiting all voxels in from Q_1 to Q_{N_S}, each Q_i is replaced with $\tilde{Q}_i$.

4. Supervoxel-NDT Registration Algorithm

The SV-NDT partitions the model scan effectively with the supervoxel-generating algorithm presented in the previous section instead of a 3-D regular grid, which does not take into account the surface structures of the model scan. This reduces the loss of information when approximating the local geometries of the model scan to ellipsoids.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

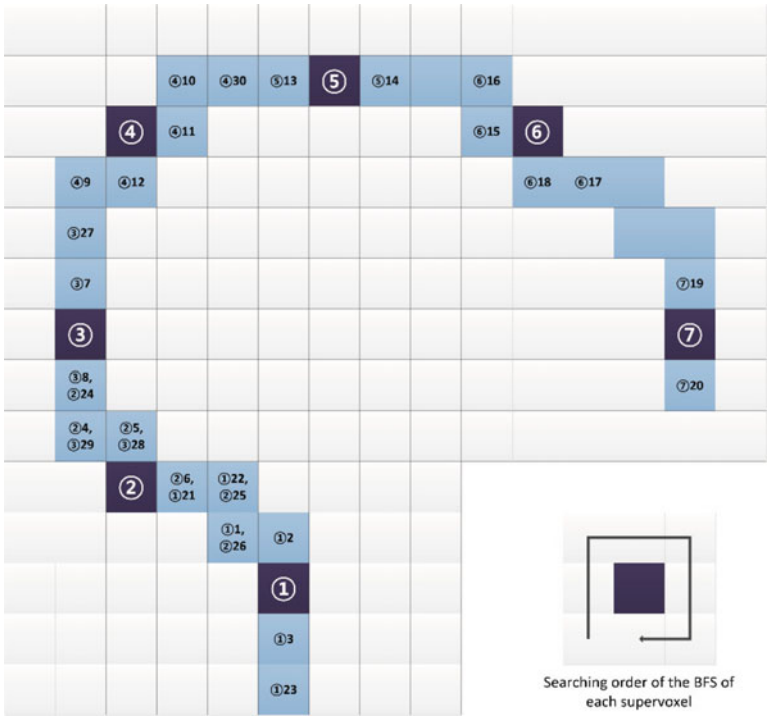


Fig. 6. An example of the traversing strategy of the flow-constrained clustering in a 2-D case. The circled number within each seed voxel represents the index of each supervoxel. The pairs of a circled number and a number in each occupied voxel, ij, represent that this voxel is visited by the *i*th supervoxel at the *j*th traversing step. The traversing steps are written up to 30 steps in this figure. The voxels in the same level of each BFS are visited clockwise direction from left-bottom corner in this figure.

In addition, the criterion to create correspondences between each point in the data scan and each normal distribution is modified. In the 3-D NDT, each point in the data scan corresponds to the normal distribution of the voxel containing it. If a point does not belong to an occupied voxel, it corresponds to the normal distribution of the closest occupied voxel. Similarly, each point in the data scan corresponds to its closest normal distribution in the SV-NDT as well. However, the distance function is not the Euclidean distance but the new distance function Δ. Δ, taking into account not only the Euclidean distance but the similarity of the normal vectors between the *i*th point in the data scan and the normal distribution of the *j*th supervoxel, is defined as

$$\Delta(i, j, \mathbf{p}) = \left(1 - \log_2\left(1 - \frac{\arccos\left|\mathbf{n}_{\mathbf{x}'_i} \cdot \mathbf{n}_{Y_j}\right|}{\frac{\pi}{2}}\right)\right) \cdot \left\|\mathbf{x}'_i - \boldsymbol{\mu}_j\right\|_2, \tag{17}$$

where $\mathbf{x}'_i$ is the *i*th point in the transformed data scan X', $\boldsymbol{\mu}_j$ is the mean vector of the point subset Y_j within the *j*th supervoxel, $\mathbf{n}_{\mathbf{x}'_i}$ is the normal vector of X' at $\mathbf{x}'_i$, $\mathbf{n}_{Y_j}$ is the normal vector of the surface within the *j*th supervoxel, and $\mathbf{p}$ is the current transformation parameter. $\mathbf{n}_{\mathbf{x}'_i}$ and $\mathbf{n}_{Y_j}$ are the normalized eigenvectors associated with the smallest eigenvalue of the covariance matrix of Y_j and the points in X' around $\mathbf{x}'_i$, respectively. The ratio of Δ to the Euclidean distance at each included angle between $\mathbf{n}_{\mathbf{x}'_i}$ and $\mathbf{n}_{Y_j}$ is shown in Fig. 7(a). When the included angle equals $\pi/4$, Δ is equal to double the Euclidean distance, and when the included angle equals $\pi/2$, Δ goes to infinity. Therefore, the distribution of the surface which is perpendicular to the surface around $\mathbf{x}'_i$ cannot be the corresponding distribution. A simple example of this effect is shown in Fig. 7(b).

A flowchart of the SV-NDT registration algorithm is shown in Fig. 8. First, a supervoxel set SV is obtained by applying the supervoxel-generating algorithm to the model scan Y. Next, the mean vector and the covariance matrix of the point subset within each supervoxel are calculated. This process is carried out using Eqs. (6) and (7) with $Y_n = \{\mathbf{y} \in Y | \mathbf{y} \in \bigcup_{k:v_k \in sv_n} v_k\}$ for the *n*th supervoxel.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

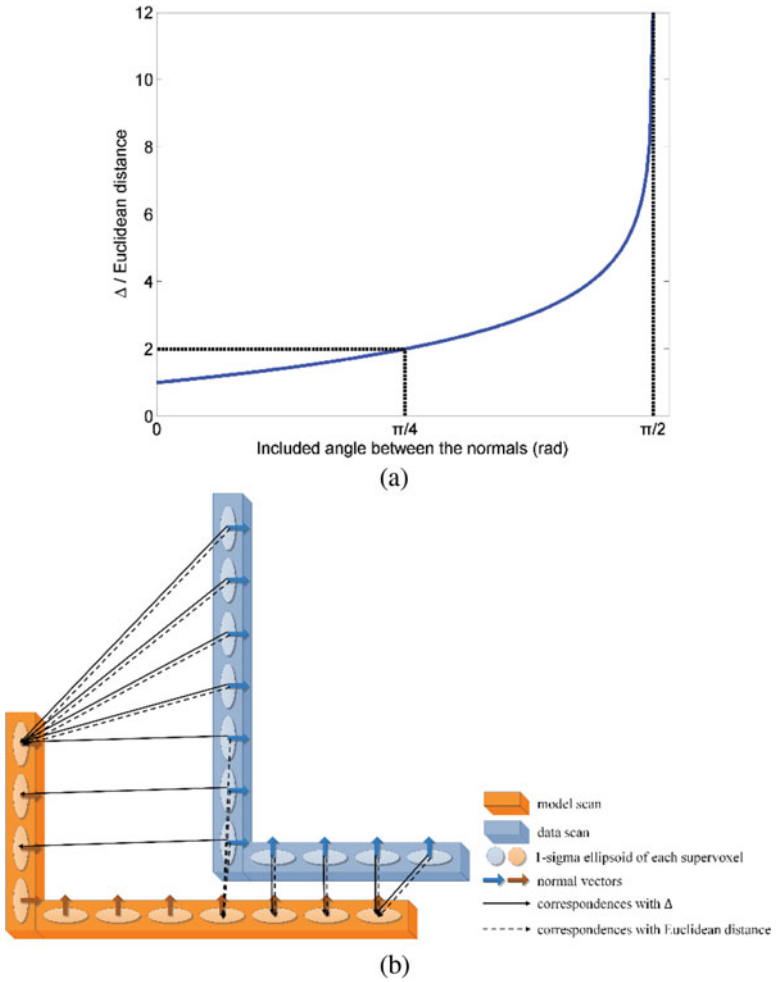


Fig. 7. (a) Relationship between the ratio of Δ to the Euclidean distance and the included angle between the normal vectors. (b) Comparison between correspondences generated by using Δ and the Euclidean distance.

After that, each normal vector $\mathbf{n}_{\mathbf{x}_i}$ of X at $\mathbf{x}_i$ and each normal vector $\mathbf{n}_{Y_j}$ of Y_j is calculated. Then, the data scan is transformed by the initial transformation parameter. After the initialization phase, the iterative optimization phase begins. The first step is to calculate the indices of the corresponding normal distributions of the transformed data scan X'. The second step is to calculate the cost function f, the gradient vector $\mathbf{g}$, and the Hessian matrix $\mathbf{H}$. Next, Newton's step $\Delta\mathbf{p}$ and the step size γ are calculated, after which $\mathbf{p}$ is updated by Eqs. (13) and (14). Subsequently, each transformed point $\mathbf{x}'_i$ with its normal vector $\mathbf{n}_{\mathbf{x}'_i}$ is calculated with the current transformation parameter $\mathbf{p}$. The normal distributions of the scan used in Fig. 1, which is generated by the supervoxel-generating algorithm are shown in Fig. 9. The surface within each supervoxel is mostly flat; thus, many normal distributions are near-degenerate. We denote the eigenvalues of the covariance matrix of each normal distribution as λ_1, λ_2, and λ_3, satisfying $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Then, if λ_2 is less than $\lambda_1/10$, λ_2 and λ_3 are replaced with $\lambda_1/10$. Also, if λ_2 is greater than or equal to $\lambda_1/10$ and λ_3 is less than $\lambda_1/10$, only λ_3 is replaced with $\lambda_1/10$.

5. Evaluation

The performance evaluation of the SV-NDT is separated into an evaluation of the supervoxel-generating algorithm and the registration process. The experiments for the performance evaluation of the supervoxel-generating algorithm under various scan conditions are presented in Section 5.1. The

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

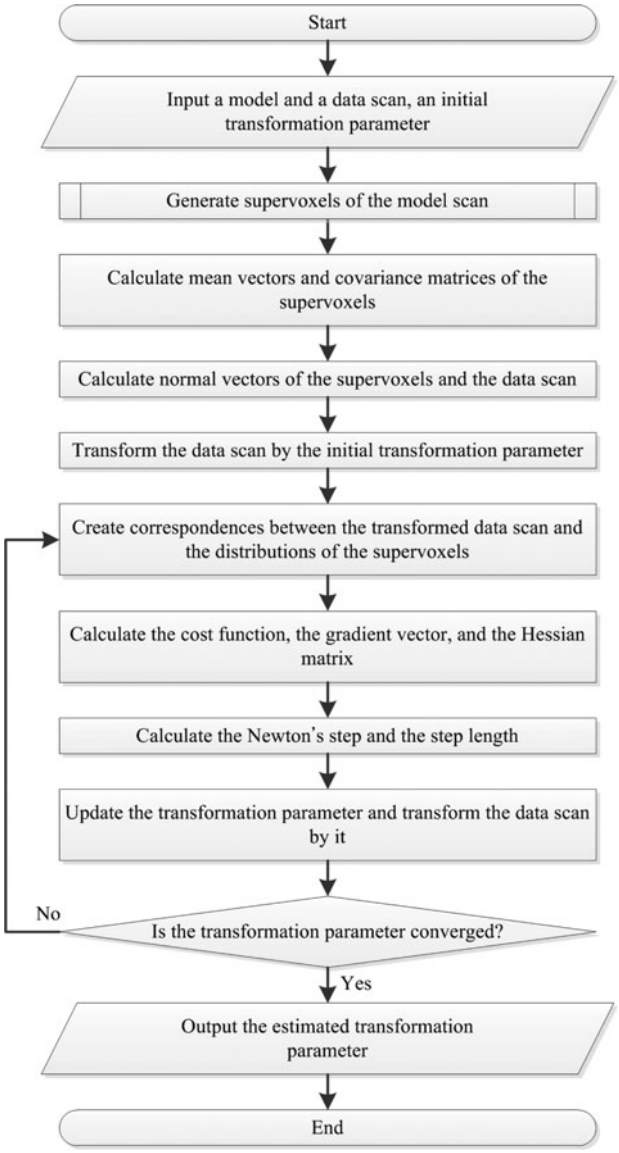


Fig. 8. Flowchart of the SV-NDT registration algorithm.

performance of the registration process of the SV-NDT is assessed through a comparison with other widely used scan registration algorithms in Section 5.2.

5.1. Supervoxel-generating algorithm

The experiments to assess the runtime of the supervoxel-generating algorithm and the modeling accuracy of the normal distributions as generated using the supervoxel-generating algorithm were performed with several types of model scans. In these experiments, synthetic scan data were used to control the attributes of the model scans and to simplify the analysis of the results. The synthetic scan data were generated by setting up a situation in which a robot in an urban environment collected scans using a Velodyne HDL-64E LiDAR sensor. Therefore, the specifications of the Velodyne HDL-64E were used to determine the angular resolutions and the field of view. Three scenes were constructed for the experiments, as shown in Fig. 10. A total of 16 scans were created by varying the three conditions. The first condition involves the existence of relatively small objects, such as trees, cars, and posts. The scans without objects are composed of large surfaces such as the ground and buildings. The second condition is the curvature of the surfaces. The scans in Fig. 10 have only planes. The rounded surfaces were generated by bending the surfaces around the edges. Given that there could

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

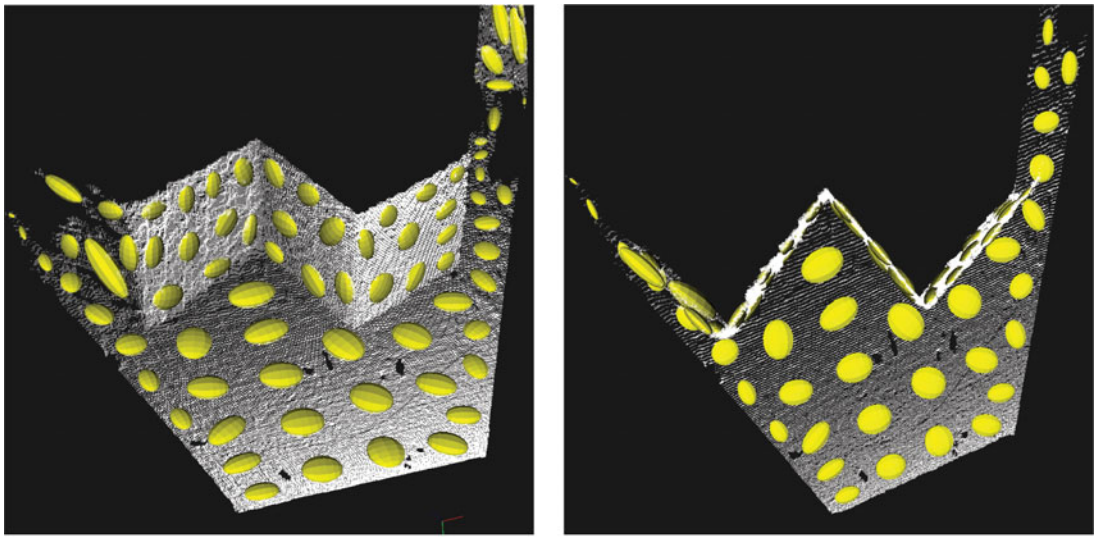


Fig. 9. (a) Front view. (b) Top view. The normal distributions which employ the supervoxel-generating algorithm to transform the model scan which is used in Fig. 1. The normal distributions model the model scan more accurately using fewer distributions. The number of normal distributions is 87 in Fig. 1 but 78 in Fig. 9.

be four types of surfaces, four scans could be generated per scene by changing these two conditions. The four scans generated from the second scene are illustrated in Fig. 11. The final condition is the noise level. The noise is modeled as additive zero-mean Gaussian noise, and its standard deviation is used as the noise level. Four values, i.e., 0, 1, 5, and 10 cm, were used as the noise levels. The synthetic scan data contain 251,688 points on average.

First, an evaluation of the runtime of the supervoxel-generating algorithm was performed. The average runtimes of the supervoxel-generating algorithm using each type of scan are given in Table I when the seed resolution was set to 5 m. The runtime does not depend on whether the objects exist, but it increased slightly when the surfaces of the scans were rounded at the edges. However, this result depends heavily on the noise levels. The runtimes at a noise level of 10 cm are more than doubled relative to those in noise-free cases. Nevertheless, the overall runtime is quite small, at less than 0.4 s, considering the number of points in the scans.

Next, the evaluation of the modeling accuracy of the normal distributions generated using the supervoxel-generating algorithm was conducted by comparing the normal distributions generated using the 3-D regular grid with the resolution equal to the seed resolution of the supervoxel-generating algorithm. The second scene was selected for the evaluation because it has the most objects, and 0 and 10 cm were used as the noise levels. Initially, noise-free scan data were used; Fig. 12 shows the normal distributions created by each method according to the type of surface. The columns present the results of the 3-D regular grid and the supervoxel-generating algorithm, and the rows show the results for each type of surface, which are in this case large planar surfaces, large rounded surfaces, large planar surfaces with objects, and large rounded surfaces with objects. For the large planar surfaces, the simplest type of surfaces, many of the normal distributions shown in Fig. 12(a) do not model local surface structures accurately because the boundaries of the cell in the 3-D regular grid do not take into account the boundaries of the local surfaces, which have different local geometries. However, the normal distributions which are generated with the supervoxel-generating algorithm model the model scan accurately (Fig. 12(b)). In Fig. 12(b), some supervoxels are composed of points on a curve and not a surface. It appears that the distributions from those supervoxels do not model the local surfaces accurately, but this stems from the limited angular resolution of the sensor. When the ray from the sensor to a point is nearly parallel to the surface containing that point, the points obtained around that point are locally formed, not as a surface but as a curve due to the limit of the angular resolution of the sensor, and the curves are far apart from each other. There is no

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

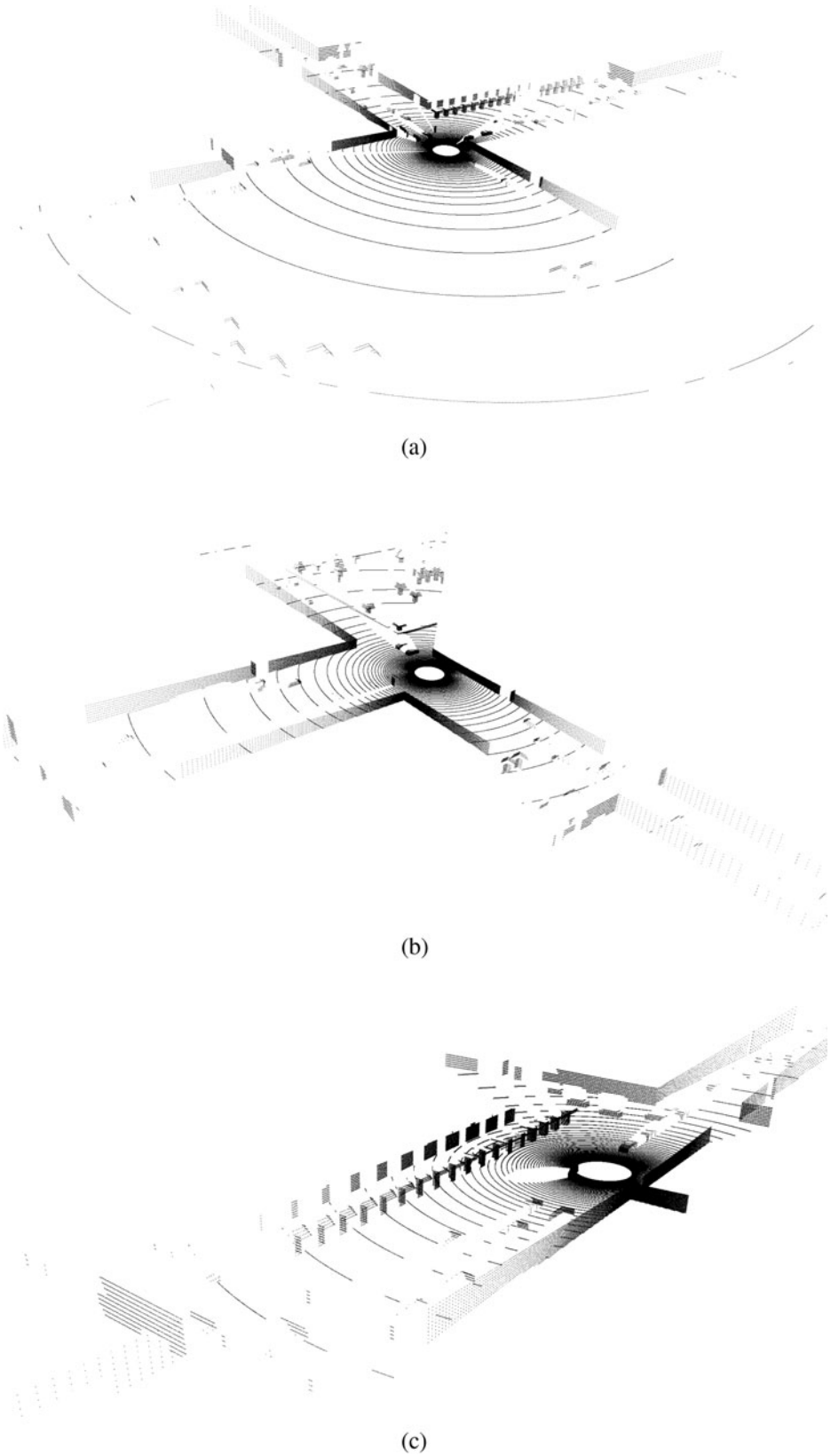


Fig. 10. Scenes of the synthetic scan data.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

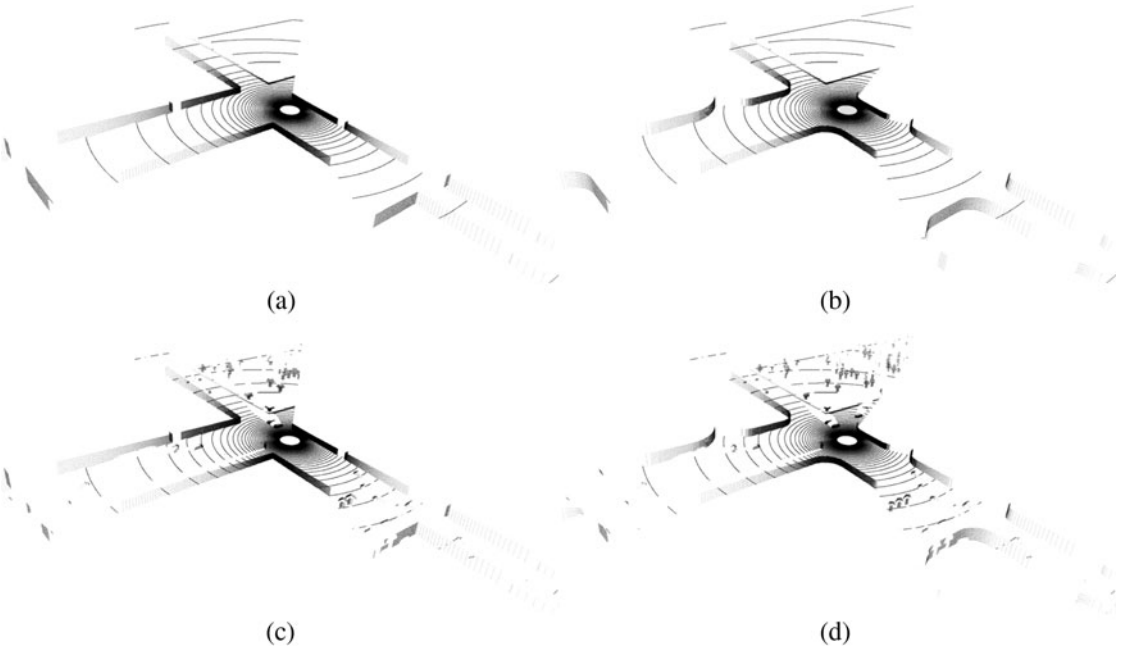


Fig. 11. Scans created by varying the type of surfaces from the second scene. (a) large planar surface (b) large rounded surfaces (c) large planar surfaces with objects (d) large rounded surfaces with objects.

Table I. Average runtime for the supervoxel generation using each kind of scans (ms).

Type of surfaces \ Noise level (cm)	0	1	5	10
Large planar surfaces	142	264	268	352
Large rounded surfaces	160	269	328	387
Large planar surfaces with objects	148	271	297	333
Large rounded surfaces with objects	148	276	322	353

reason for these curves to be considered as constituents of one local surface when no information about the angular resolution of the sensor is given. Thus, although supervoxels with ground points that are far from the sensor may contain a curve, the modeling accuracy is not affected. To evaluate the influence of the first condition, i.e., the existence of the objects, the first and the second row in Fig. 12 are compared with the third and last row in Fig. 12, respectively. When the 3-D regular grid is used, normal distributions that could not model local surfaces were created. This mainly results from the fact that both the object and some part of the ground were modeled by one normal distribution. Because a few ground points are far from the sensor, the trees near them are modeled quite well. With the supervoxel-generating algorithm, although a few distributions deteriorated the modeling quality, the added objects were modeled accurately by adding small normal distributions. To assess the influence of the second condition, i.e., whether the surfaces are planar or round, the first and the third row in Fig. 12 are compared with the second and last row in Fig. 12, respectively. However, the results of both methods are not affected by the second condition. To evaluate the effect of the final condition, i.e., the noise level, the same process was repeated with a noise level of 10 cm. These results are shown in Fig. 13. Because some of the points near the cell boundaries pass these boundaries, the modeling accuracy of the 3-D regular grid decreased sharply (the left column in Fig. 13). However, the noise had little impact on the results of the supervoxel-generating algorithm (the right column in Fig. 13). The supervoxel-generating algorithm created the boundaries of the supervoxels based on the local surface structures; thus, it is robust to a high noise level.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

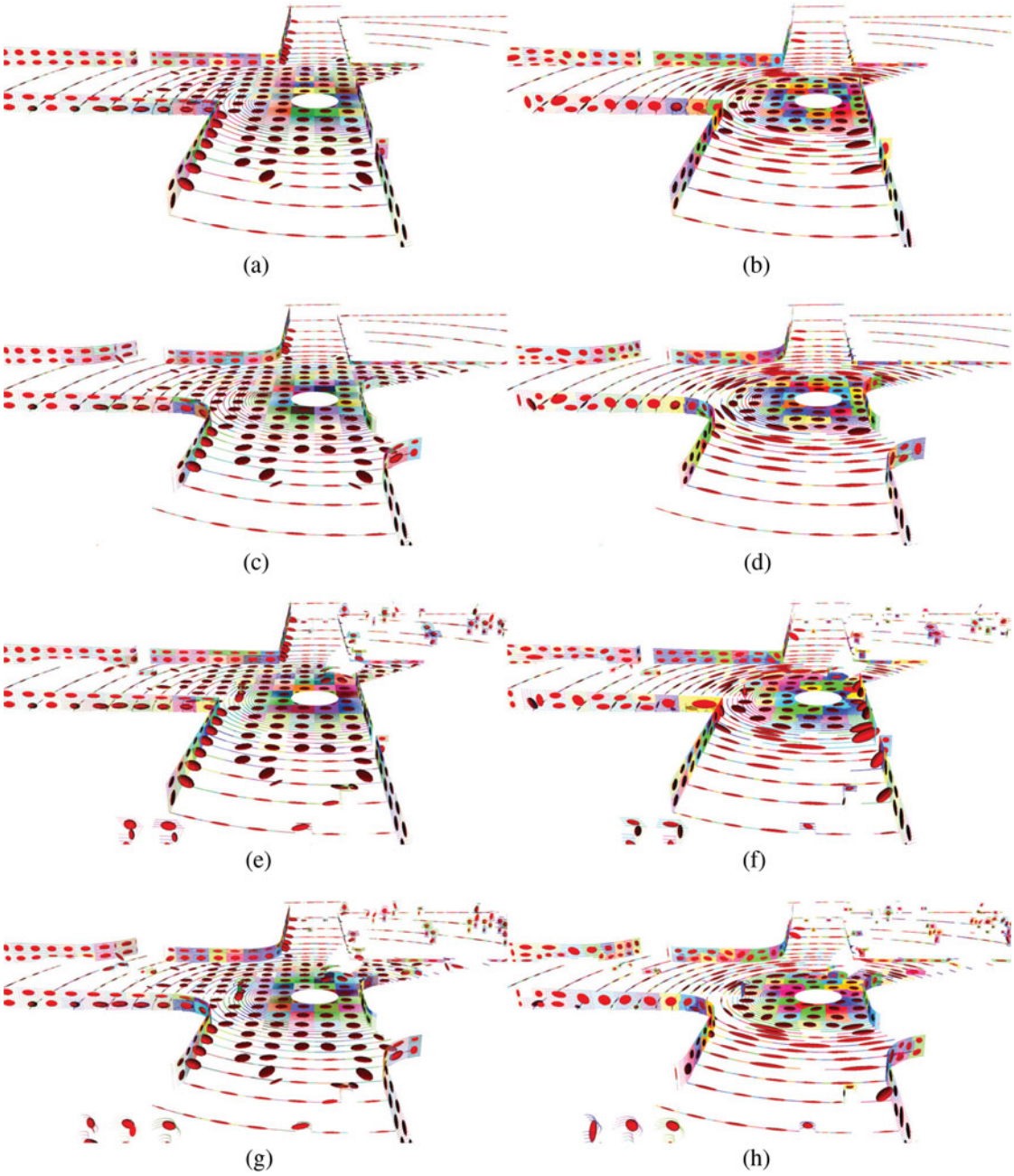


Fig. 12. Normal distributions created by each method according to the type of surfaces with zero noise level. Each column presents the results of the 3-D regular grid and the supervoxel-generating algorithm, and each row shows the results of each type of surfaces, which are large planar surfaces, large rounded surfaces, large planar surfaces with objects, and large rounded surfaces with objects.

5.2. Registration

The experiments to assess the performance of the registration process by the SV-NDT used a point cloud dataset in both synthetic and real-world datasets. Not only the 3-D NDT but also the ICP and the G-ICP, which are both widely used 3-D scan registration algorithms, were selected for the comparison of the performance. The comparison factors were the robustness, accuracy, and runtime. These three algorithms were implemented with a point cloud library (PCL). In addition to those algorithms, the SV-NDT which uses the Euclidean distance when creating correspondences between the points in the data scan and the normal distributions was also evaluated on the synthetic scan data in order to assess the effect of the proposed distance function.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

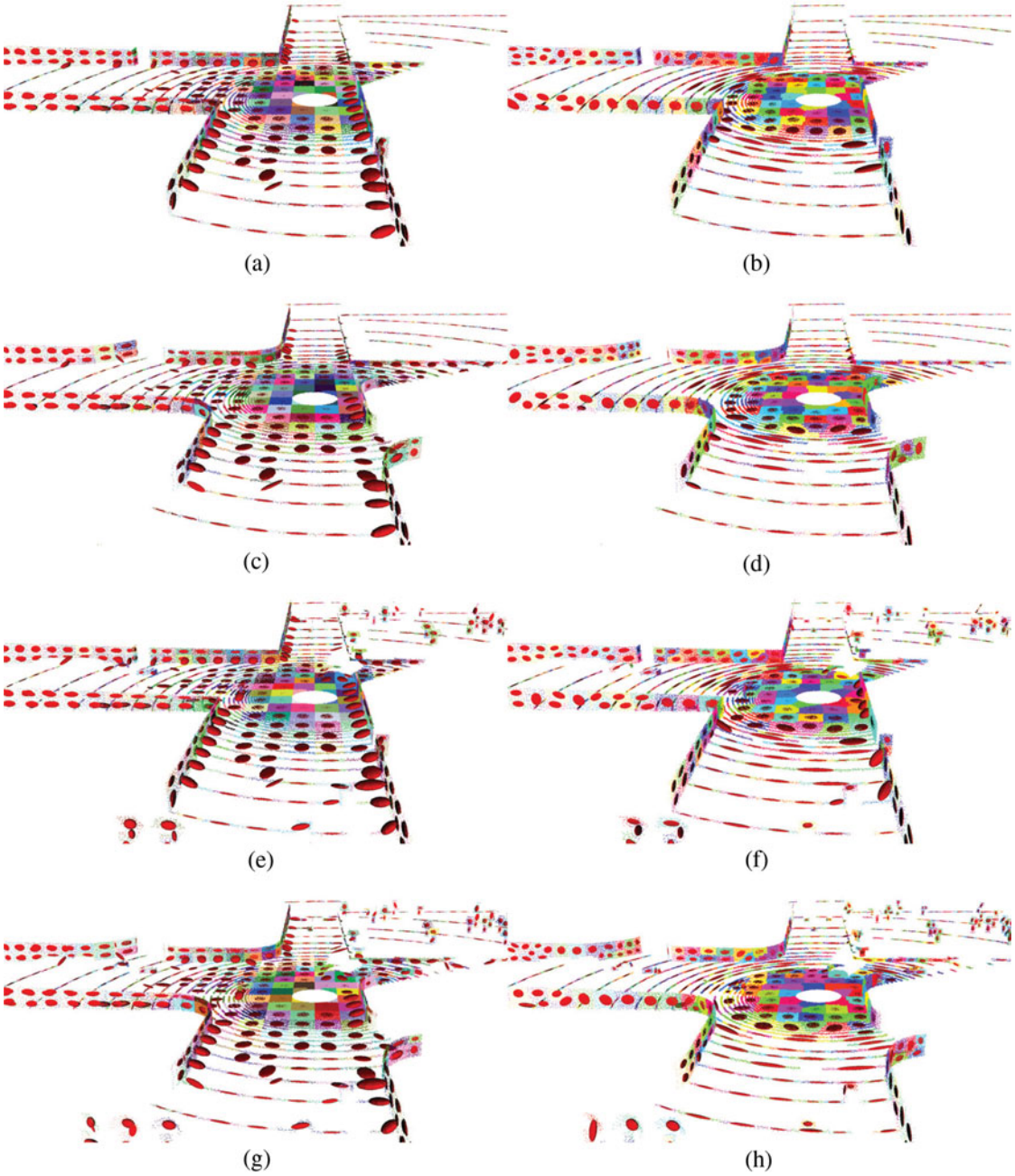


Fig. 13. Normal distributions created by each method according to the type of surfaces with 10 cm noise level. Each column presents the results of the 3-D regular grid and the supervoxel-generating algorithm, and each row shows the results of each type of surfaces, which are large planar surfaces, large rounded surfaces, large planar surfaces with objects, and large rounded surfaces with objects.

First, the synthetic scan data are used for the experiments. Each point in the synthetic scan data used in the previous section was paired by creating new scans whose origins are 5 m away from the origins of the existing scans. Therefore, a total of 48 scan pairs were used, and the raw data were used for the experiments. The seed resolution in the SV-NDT and the voxel resolution in the 3-D NDT were set to 5 m. Likewise, the maximum distance threshold between two corresponding points is a parameter in the ICP and the G-ICP, and it was set to 10 m for each algorithm. These parameters were determined from among 2.5, 5, and 10 m by experiments to determine which one provides the best performance with those scan data. The registration experiments were performed using a variety of

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

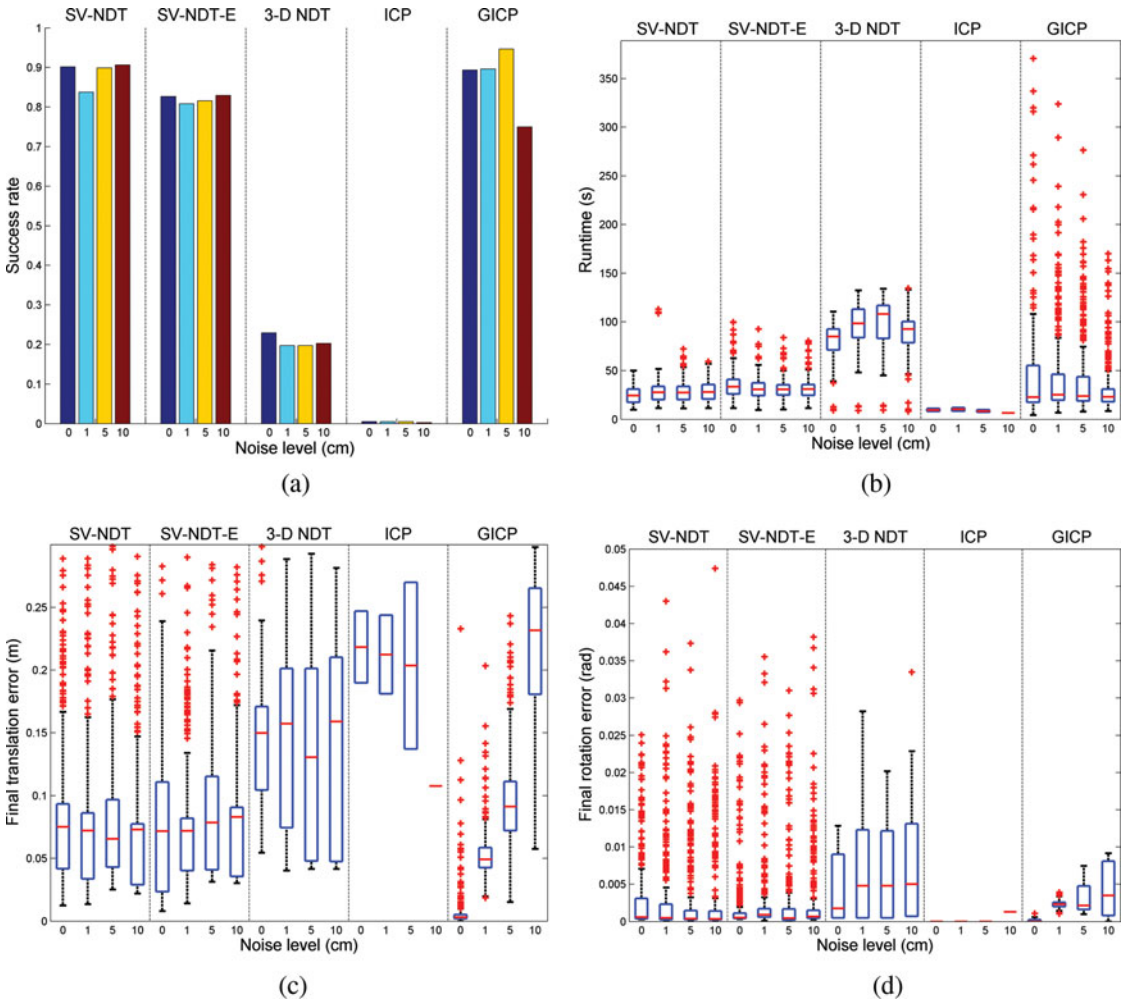


Fig. 14. Registration results of each registration algorithm according to the noise levels using the type of large planar surfaces. (a) success rate (b) runtime (c) final translation error (d) final rotation error.

initial transformation errors. The tested initial transformation errors were composed of the translation errors from −3 to 3 m at intervals of 1.5 m along the x and y axes and the rotation errors from −30 to 30° at intervals of 15° about the z axis. Thus, a total of 125 initial transformation errors were tested for each scan pair. To measure the robustness and accuracy, the success criteria were set to a translation error of 0.3 m and a rotation error of 0.05 radians. The robustness of each registration algorithm was measured by the success rate, and the success rate of each algorithm according to the conditions, the type of surfaces and the noise level, was calculated by the success rate out of a total 375 registration results generated by applying 125 initial transformation errors to three scenes with the same conditions. The runtime and the accuracy of each algorithm were measured using the runtime, the final translation error, and the final rotation error of the successful registration cases. For each type of surface, the results for each algorithm according to the noise level are illustrated in Figs. 14–17. The results in Figs. 14–17 were obtained using large planar surfaces, large rounded surfaces, large planar surfaces with objects, and large rounded surfaces with objects. To compare each algorithm easily, every figure is divided into five sections (from left: SV-NDT, SV-NDT with the Euclidean distance (SV-NDT-E), the 3-D NDT, ICP, and G-ICP). Each section shows the results of each algorithm according to the noise level (from left: 0, 1, 5, and 10 cm).

The first plot of each figure shows the success rate of each algorithm using bar graphs. The second to fourth plots of each figure respectively present the runtime, final translation and rotation error of each algorithm using boxplots. In each boxplot, the red line shows the median, and the blue box represents the range from the 25th percentile to the 75th percentile. The length of the whiskers is

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

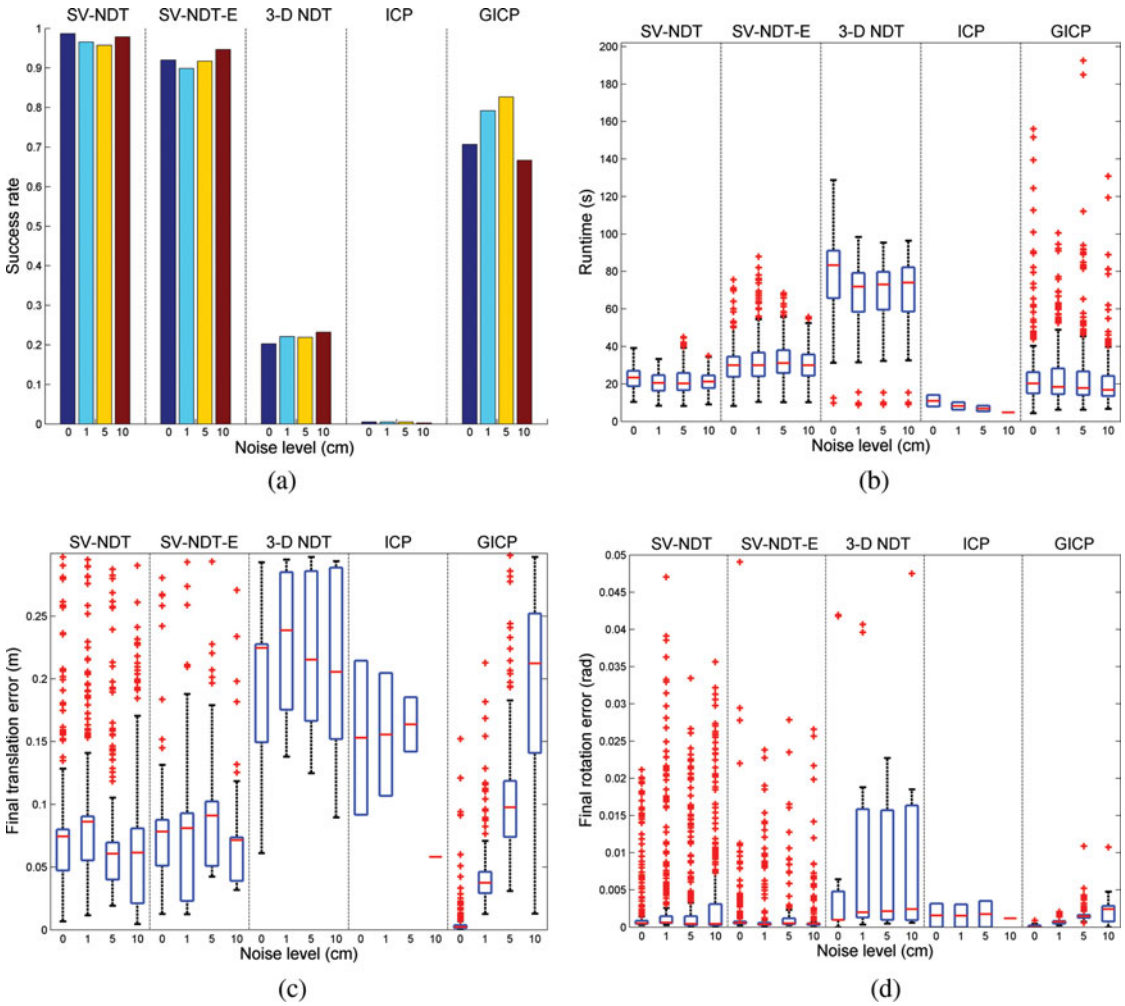


Fig. 15. Registration results of each registration algorithm according to the noise levels using the type of large rounded surfaces. (a) success rate (b) runtime (c) final translation error (d) final rotation error.

1.5 times the length between the 25th percentile and the 75th percentile, and the results outside the whiskers are considered as outliers.

First, the robustness of each algorithm was analyzed. As shown in the figures, the most robust algorithm is the SV-NDT, and its success rates are greater than 0.9 in all but two cases of conditions. Moreover, its lowest success rate is 0.8373. The G-ICP is more robust than the SV-NDT-E except for the case of the large rounded surfaces, and it is followed by the 3-D NDT and the ICP. The 3-D NDT is more robust than the ICP, but it cannot overcome the large initial transformation errors. For all except for the SV-NDT-E, the success rates increased when the scan data with the objects were used, as the objects provide additional information with which to estimate the relative transformation between the two scans. The success rates of the SV-NDT and the SV-NDT-E when rounded surfaces are used are larger than those when planar surfaces are used, but the G-ICP has the opposite effect. The overall robustness levels of the SV-NDT, the SV-NDT-E, and the 3-D NDT depend little on the noise level, whereas the robustness of the G-ICP relies heavily on the noise level. When the noise level is 10 cm, the success rate of the G-ICP falls rapidly. This outcome results from the number of points used to estimate the covariance matrices. The SV-NDT and the 3-D NDT use points in each part of the partitioned model scan; thus, many covariance matrices are reliably estimated due to the use of a sufficient number of points. However, the G-ICP uses only a predefined number of points around each point to estimate the covariance.

Next, the runtime of each algorithm is evaluated. The ICP is the fastest, followed by the SV-NDT, the SV-NDT-E, the G-ICP, and the 3-D NDT. The ICP achieves success only when the initial

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

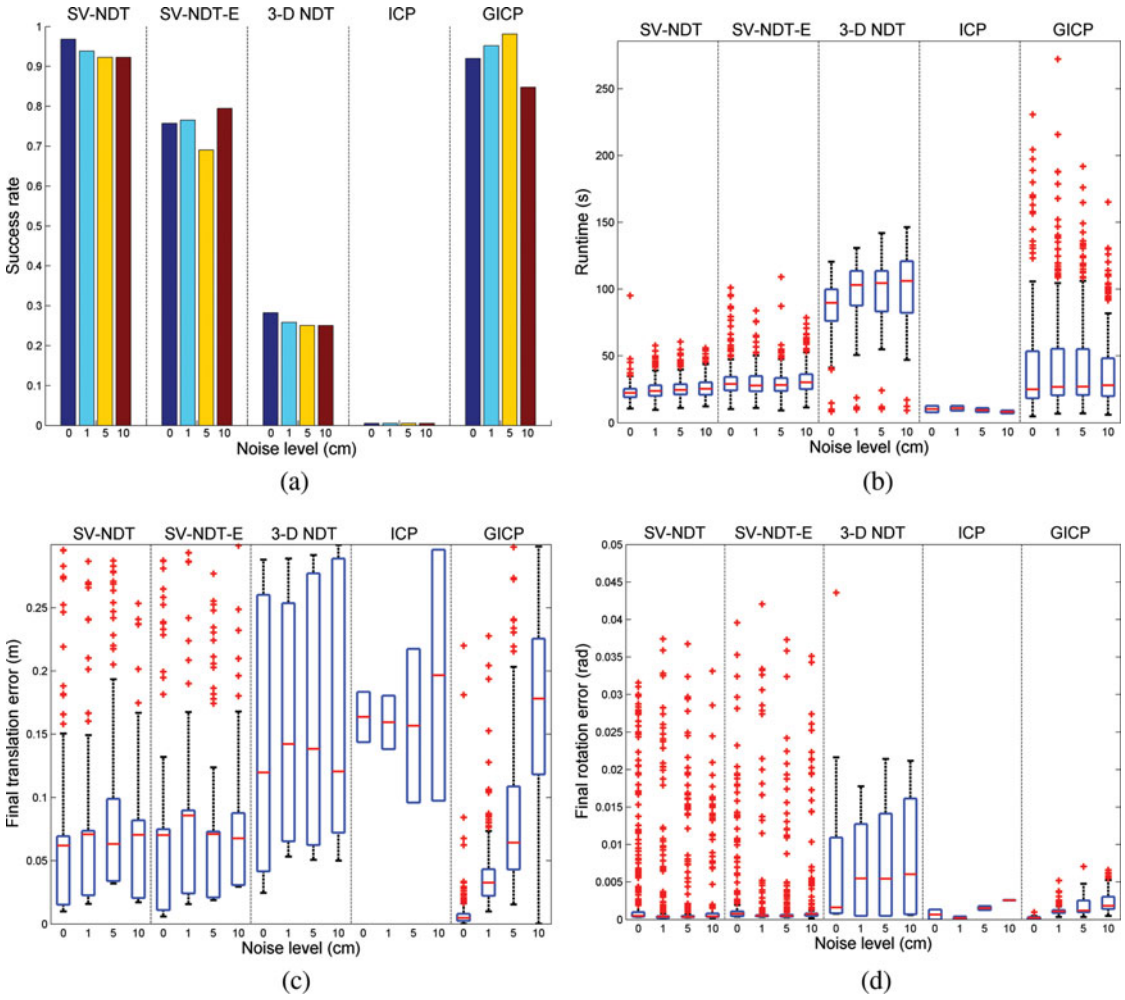


Fig. 16. Registration results of each registration algorithm according to the noise levels using the type of large planar surfaces with objects. (a) success rate (b) runtime (c) final translation error (d) final rotation error.

transformation errors are quite small. Thus, the runtime is very small, about 8.86 s, compared to the other algorithms. The median runtimes of the SV-NDT and the G-ICP, about 24.26 and 23.14 s respectively, are similar to each other, but their corresponding 75th percentiles, about 29.43 and 40.58 s, differ considerably. The median runtime of the SV-NDT-E is about 30.07 s, which is larger than that of the G-ICP, but the 75th percentile of the SV-NDT-E, about 36.03 s, is smaller than that of the G-ICP.

Finally, the accuracy levels were compared. The results of the SV-NDT, the SV-NDT-E, and the 3-D NDT do not show any tendency with regard to different noise levels. In addition, they show similar accuracy levels in many cases. However, the G-ICP shows a clear tendency. In the noise-free case, the G-ICP is the most accurate, but the accuracy decreases sharply with an increase in the noise level. The reason is identical to that of the success rate: the number of points used to estimate each covariance matrix. The overall accuracy of the 3-D NDT is much lower than that of the SV-NDT because the normal distributions generated when using the 3-D regular grid cannot model the model scan accurately, as shown in Section 5.1.

The SV-NDT-E is significantly superior to the 3-D NDT in all aspects but lacks the performance in terms of the robustness and runtime compared to the SV-NDT. This shows that the registration performance is heavily affected by the modeling accuracy of the normal distributions as transformed from the model scan. Moreover, the proposed distance function has a positive effect on the size of the basin of attraction and the speed of convergence.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

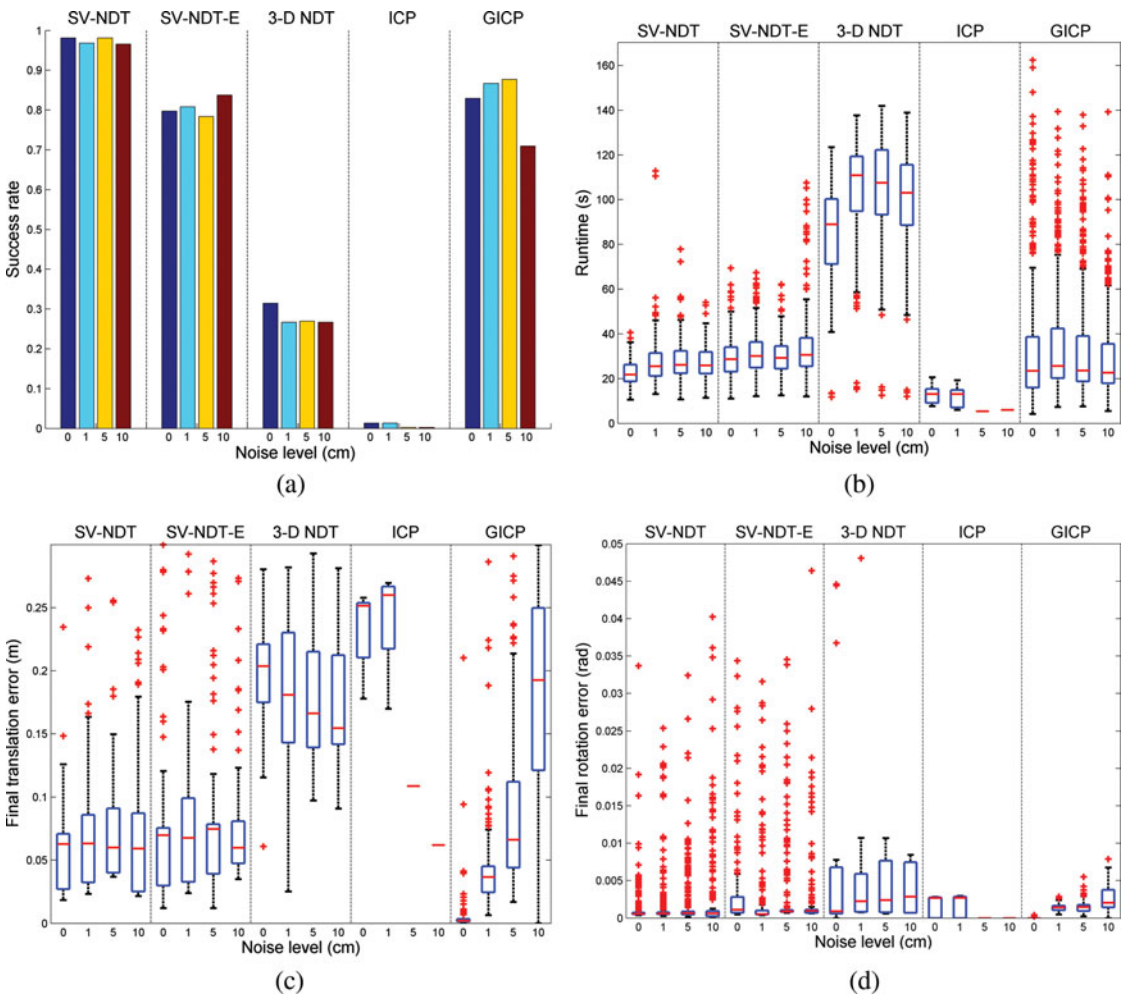


Fig. 17. Registration results of each registration algorithm according to the noise levels using the type of large rounded surfaces with objects. (a) success rate (b) runtime (c) final translation error (d) final rotation error.

The real-world dataset is the Ford Campus Vision and Lidar Data Set[30] gathered using a Velodyne HDL-64E LiDAR sensor. Five distinguishing scan pairs with time step differences of 10 were selected for the test. Figure 18 and Table II show the positions and the detailed surroundings of the test scan pairs, respectively. The raw data, which contain 78,341 points on average, were used in an intact form. The seed resolution in the SV-NDT and the voxel resolution in the 3-D NDT were set to 10 m and 5 m, respectively, and the maximum correspondence distance was set to 10 m for each algorithm. These parameters were determined from among 3, 5, 10, and 20 m by experiments to determine which one provides the best performance with the dataset.

Initially, cases in which the initial transformation errors consist of either translation or rotation errors were evaluated so as to evaluate the robustness of each algorithm to the initial translation and initial rotation errors separately. First, experiments to test the robustness to initial translation errors were performed for each algorithm. The tested initial translation errors ranged from -5 to 5 m at intervals of 1 m along the x and y axes. Figure 19 shows boxplots of the results of these experiments. The horizontal axis represents the magnitude of each initial translation error vector. Because the initial translation errors are two-dimensional vectors, the magnitude difference is not uniform. The dashed line represents the success criterion, which was set to 0.3 m. The results show that the SV-NDT and the G-ICP are most robust to the initial translation error. In the ICP case, the final translation errors increase with greater initial translation errors. The 3-D NDT fails in registration when the initial translation errors are larger than 4 m. However, all of the medians of the final translation errors of the SV-NDT and the G-ICP are under the dashed line. Nevertheless, the SV-NDT has many outliers

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

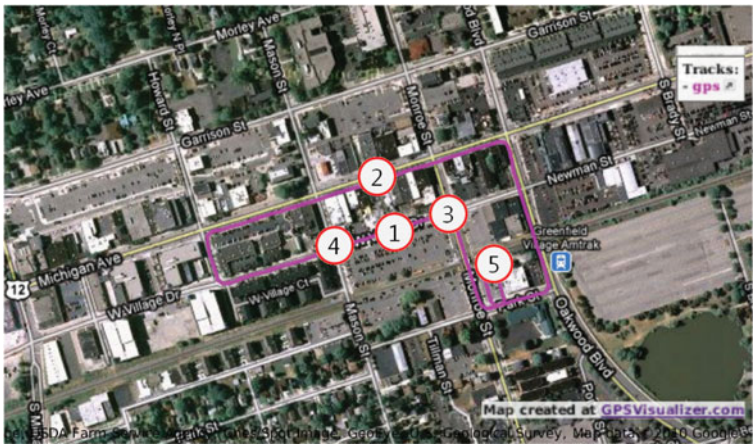


Fig. 18. Positions of the test scan pairs with full path of the dataset.

Table II. Detailed surroundings of the test scan pairs.

Scan pair number	Detailed surroundings
1	A straight road with low car density
2	A straight road with high car density
3	Turning left at an intersection
4	Going straight at an intersection
5	A parking lot

when the initial translation errors are greater than or equal to the square root of 18. About half of these outliers occur from the second scan pair. The major characteristic of the environment of the second scan pair is that there are numerous moving cars. Considering that most of the scanned points from cars are from the sides of the cars, the estimated local surfaces from the cars are perpendicular to the ground. The G-ICP algorithm makes correspondences using the Euclidean distance and takes the normal directions of the local surfaces into consideration by adjusting for the influence of the correspondences via a cost function according to the similarity of the normal directions. The points from moving cars in the data scan correspond to the points from the ground; thus, their influences on the cost function are reduced sharply. Consequently, the G-ICP has few outliers in spite of the many moving cars which existed in this case. However, the SV-NDT algorithm takes the normal directions of the local surfaces into account when making correspondences. Thus, the points from moving cars in the data scan correspond to the distributions from the same cars which are, however, not in the same position or from other surfaces which are not parallel to the ground. As a result, a few final translation errors are quite large and are therefore considered as outliers, but most of the final translation errors, at least three quarters of them except for the initial translation errors, are the square roots of 18 and 50, i.e., small enough to be regarded as successful cases. Next, experiments to test the robustness to the initial rotation error were performed for each algorithm. The tested initial rotation errors ranged from -50 to 50° at intervals of 10° about the z axis. Boxplots of the results of these experiments are shown in Fig. 20. The dashed line, the success criterion, was set to 0.05 radians. The SV-NDT, the 3-D NDT, and the G-ICP carried out the scan registration successfully with an initial rotation error of up to 20°, but the ICP failed with small initial rotation errors. Because the 25th percentiles of the final rotation errors of the SV-NDT and the 3-D NDT are under the dashed line in the case of 30°, the SV-NDT and the 3-D NDT are the most robust algorithms with regard to the initial rotation errors.

Given that the initial transformation error consists of both translation and rotation error concurrently in general cases of scan registration, an assessment of those cases is also required. Thus, experiments whose initial transformation errors were composed of initial translation and initial rotation errors were performed for each algorithm. The tested initial formation errors ranged from -5 to 5 m of the

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

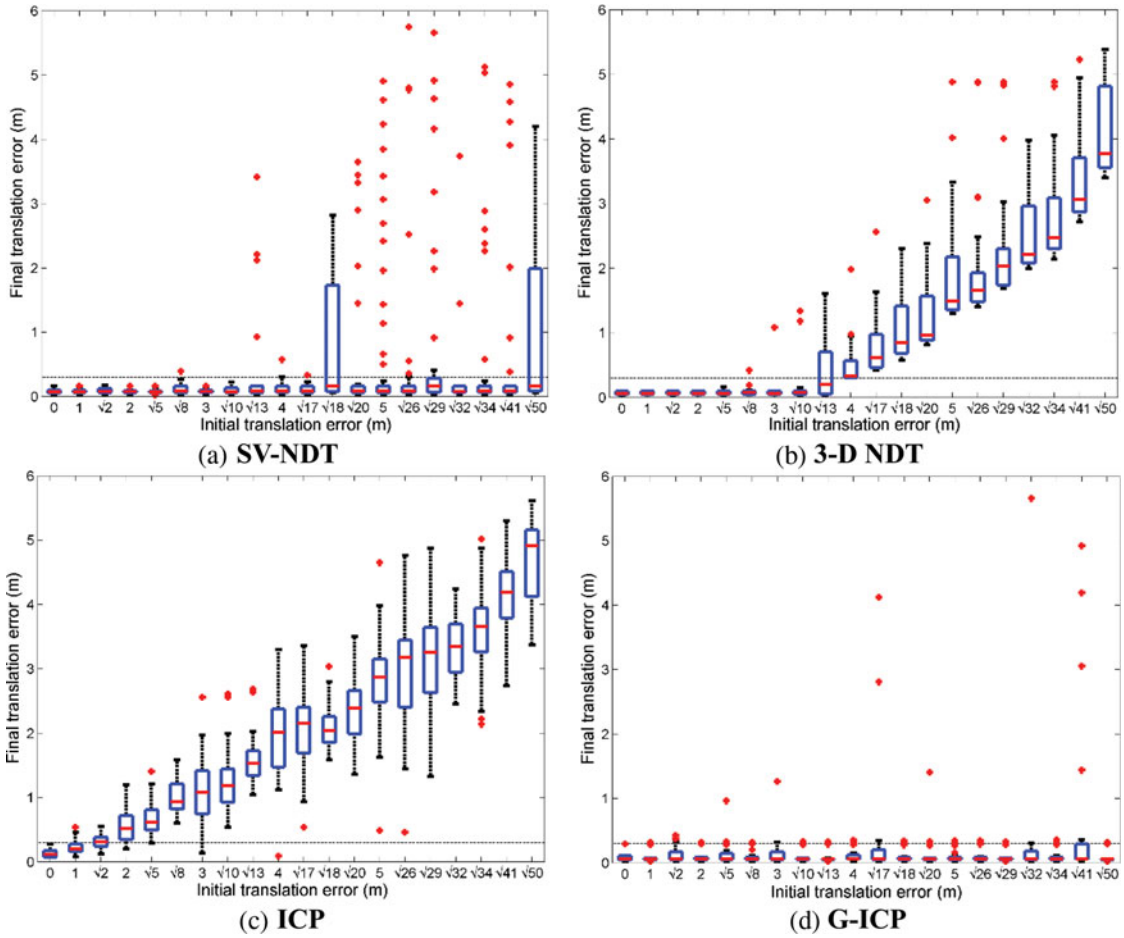


Fig. 19. The boxplots of the final translation errors of each registration algorithm according to the initial translation errors when the initial transformation errors consist of only the initial translation errors. (a) SV-NDT (b) 3-D NDT (c) ICP (d) G-ICP.

initial translation error at intervals of 1 m along the x and y axes and from -50 to 50° of the initial rotation error at intervals of 10° about z axis. In other words, each algorithm runs a total of 1331 registration per scan pair. Boxplots of the translation and the rotation part of the results are shown in Figs. 21 and 22, respectively. The SV-NDT outperforms the other scan registration algorithms. Because the initial translation and rotation errors were combined, the overall magnitude of the final translation or rotation errors and the numbers of outliers are increased for all algorithms. This stems from the differences in the sizes of the basins of attraction of each registration algorithm. Because the sizes of the basins of attraction of the 3-D NDT and the ICP are small, the estimated transformations are close to the initial transformation. Thus, the final transformation errors increase with the initial transformation errors, but they are narrowly distributed. As a result, the numbers of outliers of the 3-D NDT and the ICP are small. However, in that the SV-NDT and the G-ICP have large basins of attraction of the correct solution, these algorithms can overcome large initial transformation errors, but in the cases with converging local minima, the distributions of the final transformation errors are wide. This is why the SV-NDT and the G-ICP have many outliers, as shown in Figs. 21 and 22. In spite of that, the medians of the final translation errors of the SV-NDT are under the dashed line when the initial translation errors are less than 5 m. Although the 25th percentiles of the final translation errors of the G-ICP are as low as those of the SV-NDT, the medians increase with the initial translation error. The success rate of the 3-D NDT is greater than 25% when the initial translation error is as high as 4 m, but the 3-D NDT fails in registration when the initial translation errors are larger than 4 m. Because the ICP is not robust to the initial rotation errors, the robustness decreases drastically regardless of the initial translation error. According to Fig. 22, the SV-NDT succeeded in most of the

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

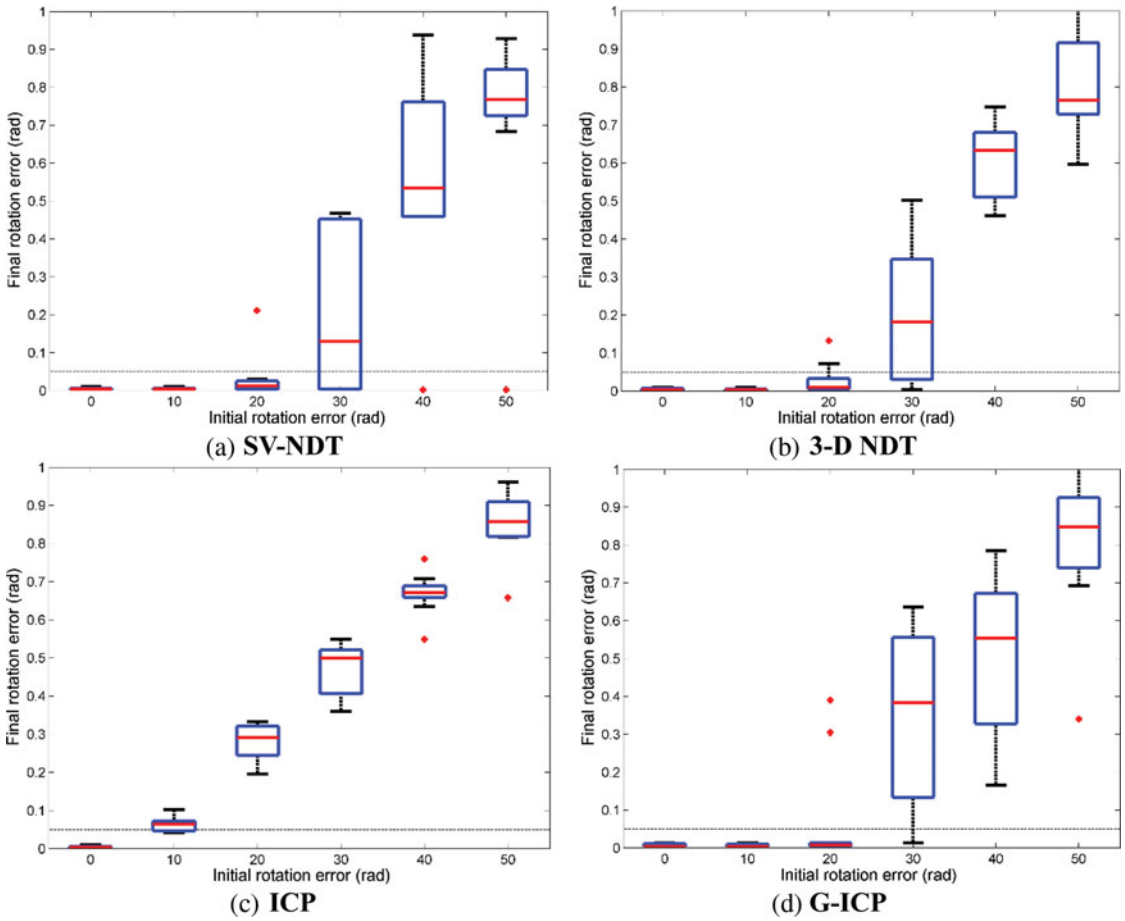


Fig. 20. The boxplots of the final rotation errors of each registration algorithm according to the initial rotation errors when the initial transformation errors consist of only the initial rotation errors. (a) SV-NDT (b) 3-D NDT (c) ICP (d) G-ICP.

registrations up to an initial rotation error of 20°, and the 3-D NDT and G-ICP achieved success up to an initial rotation error of 10°. In addition, the success rate of the SV-NDT is greater than 50% when the initial rotation error is 30°.

The success rate of each algorithm is summarized in Fig. 23 and in the first row of Table III. The SV-NDT far surpasses the 3-D NDT and is superior to the other algorithms on average although the G-ICP is better than the SV-NDT on scan pairs 3 and 4, which were collected at intersections. The overall success rates are somewhat low, as the initial translation errors are two-dimensional. When magnitude of the initial translation error is larger, there exist more cases of the initial translation error with the same magnitude. When the initial transformation error is larger than 5 m for translations and larger than 30° for rotations, no algorithm works well. There are 848 of these cases, representing 63.7% out of the 1331 cases in total. The second row in Table III shows the partial averages of the success rates, apart from those cases.

For a comparison of the accuracy levels, the final transformation errors of the successful registration cases were evaluated. Boxplots of the translation part and the rotation part of these results are shown in Fig. 24, and medians in the boxplots are summarized in the third and fourth row in Table III. In the translation part, the 3-D NDT and the G-ICP show the best results on the basis of the median values, followed by the SV-NDT and the ICP. The difference between the median values of the SV-NDT and the NDT is approximately 1 cm, which is tiny with respect to the scale of the scans, about 120 m, showing that the translation accuracy levels of the SV-NDT, the 3-D NDT, and the G-ICP are nearly identical. In the rotation part, the SV-NDT is slightly superior to the other algorithms, followed by the G-ICP, the ICP, and the 3-D NDT, but they are also about the same.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

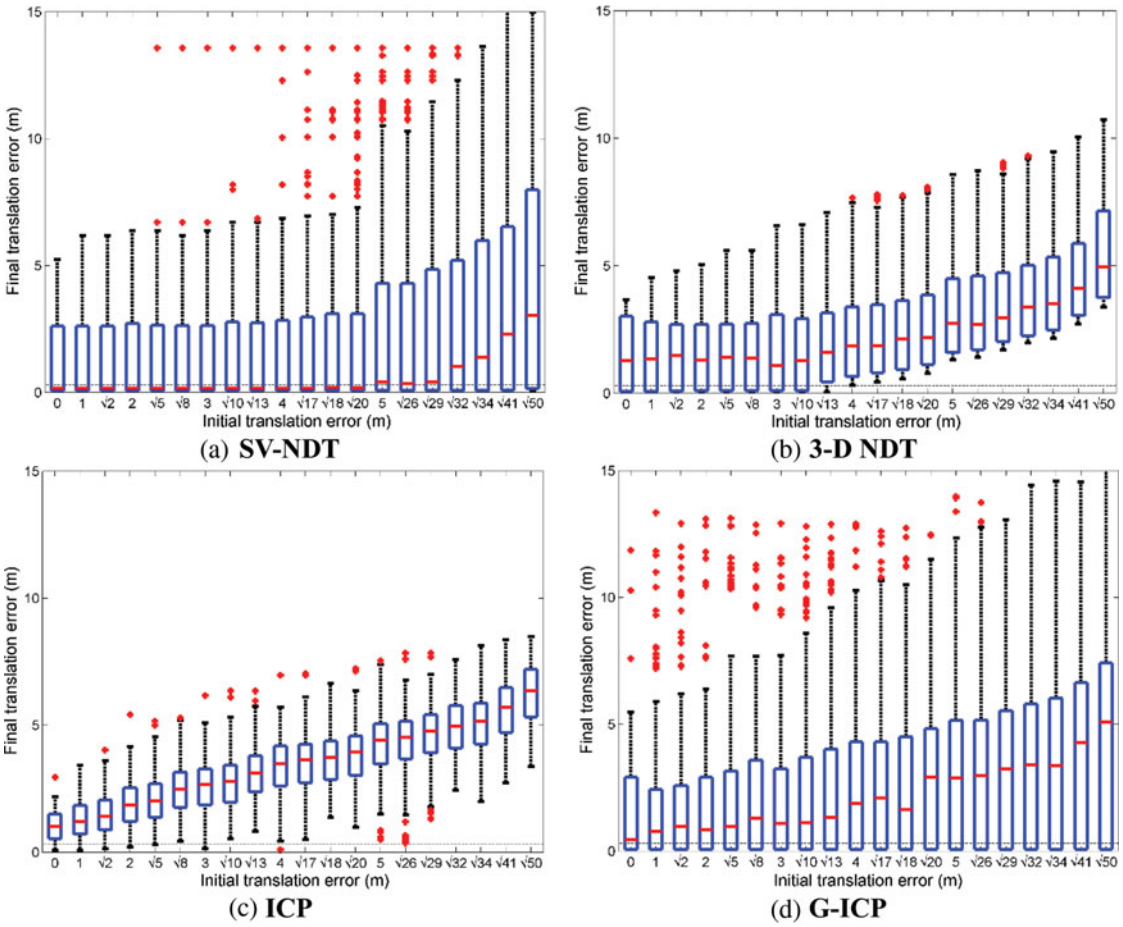


Fig. 21. The boxplots of the final translation errors of each registration algorithm according to the initial translation errors when the initial transformation errors consist of both initial translation and initial rotation errors. (a) SV-NDT (b) 3-D NDT (c) ICP (d) G-ICP.

Finally, the runtimes of each registration algorithm were assessed. Figure 25 shows boxplots of the runtimes of the successful registration cases. The ICP is the fastest, followed by the SV-NDT, the G-ICP, and the 3-D NDT. The ICP succeeded in registrations with only minor initial transformation errors; thus, its runtime value is much lower than that of the others. The median values of the runtimes of each algorithm are shown in the last row of Table III. The SV-NDT is about 1.96 times faster than the G-ICP and is about 5.92 times faster than the 3-D NDT. Additionally, the runtimes of the supervoxel-generating algorithm for each scan pair are presented in Table IV. These runtimes are about 68.4 milliseconds on average, and they are only 1.70 percent of the average of the overall runtimes of the SV-NDT registration algorithm.

6. Conclusions

In this paper, a novel algorithm based on the 3-D NDT registration algorithm called the SV-NDT is proposed. It reduces the information loss of local surface structures with a supervoxel-generating algorithm when transforming the model scan into normal distributions, and it utilizes the local surface structures of the data scan through the use a newly proposed distance function which selects the corresponding distribution of each point in the data scan rather than the Euclidean distance. The results of an evaluation of the supervoxel-generating algorithm on synthetic scan data show that the algorithm greatly increases the modeling accuracy of the normal distributions. Furthermore, experiments which ran a performance evaluation of the SV-NDT on synthetic and real-world datasets demonstrated that the robustness and speed of the SV-NDT significantly exceed those of previous algorithms while maintaining comparable levels of accuracy.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

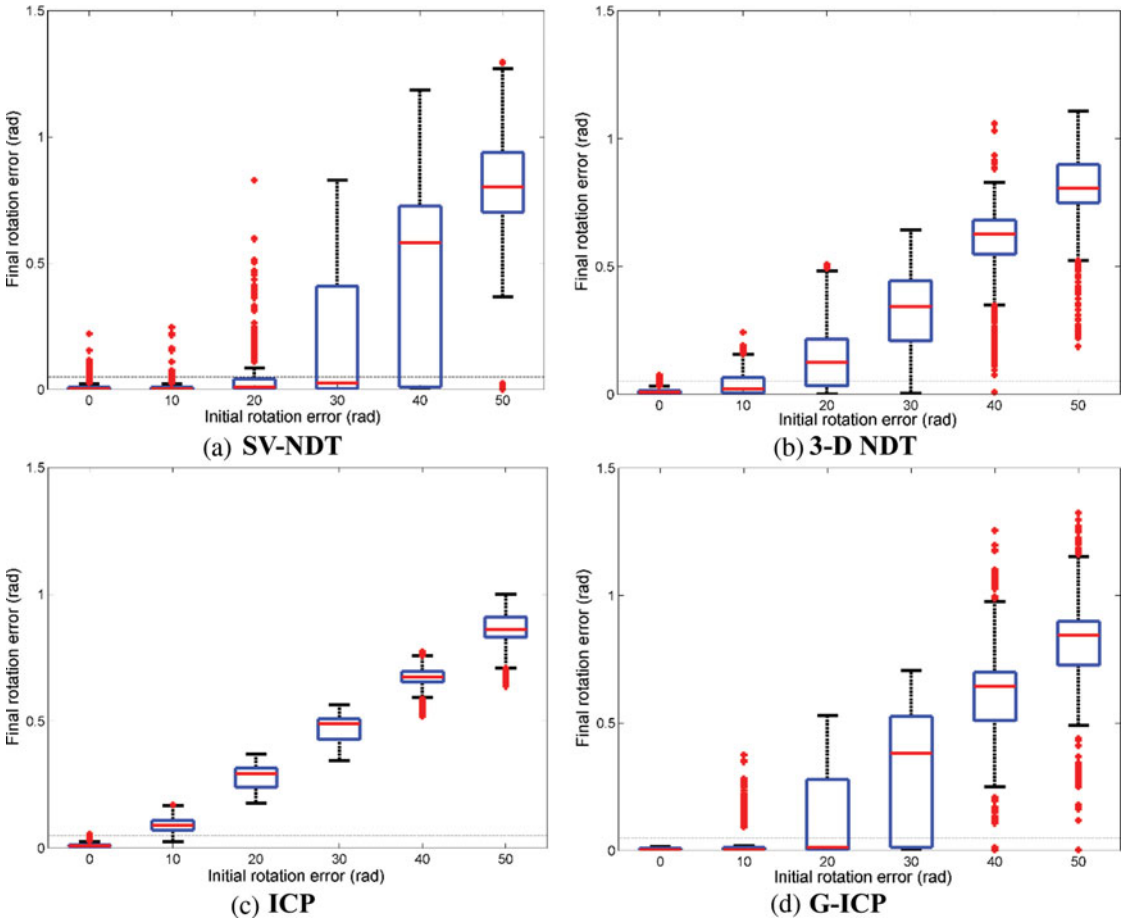


Fig. 22. The boxplots of the final rotation errors of each registration algorithm according to the initial rotation errors when the initial transformation errors consist of both initial translation and initial rotation errors. (a) SV-NDT (b) 3-D NDT (c) ICP (d) G-ICP.

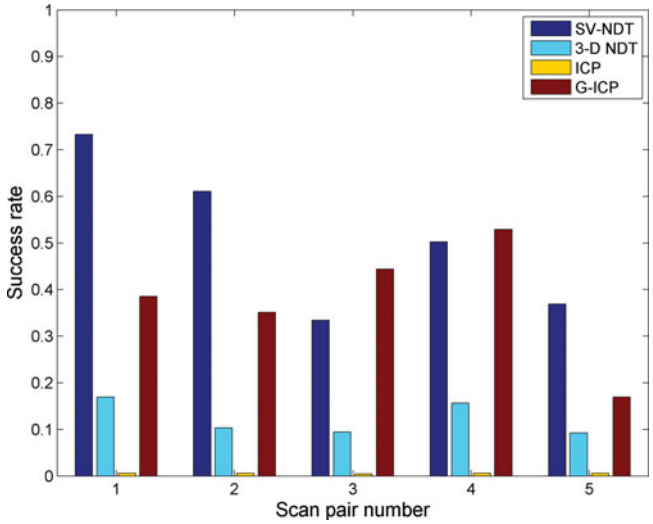


Fig. 23. Success rate of each algorithm for each scan pair.

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

Table III. Performances of each algorithm.

	SV-NDT	3-D NDT	ICP	G-ICP
Success rate average	0.5098	0.1231	0.0057	0.3757
Success rate partial average	0.7342	0.3391	0.0157	0.5921
Translation accuracy (mm)	73.523	61.471	190.58	61.31
Rotation accuracy (mrad)	3.017	4.183	3.502	3.50
Runtime (s)	4.025	23.835	1.862	7.88

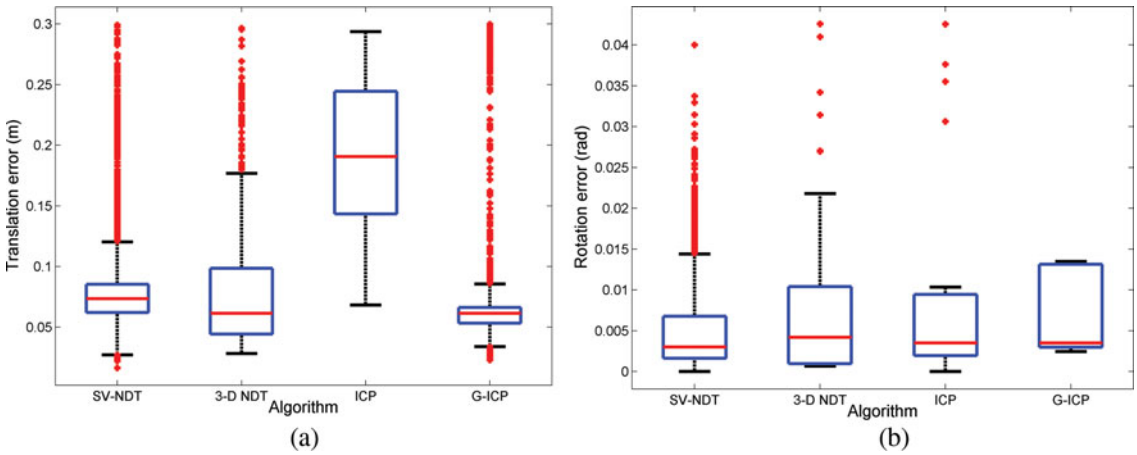


Fig. 24. Accuracy of each registration algorithm (a) translation part (b) rotation part.

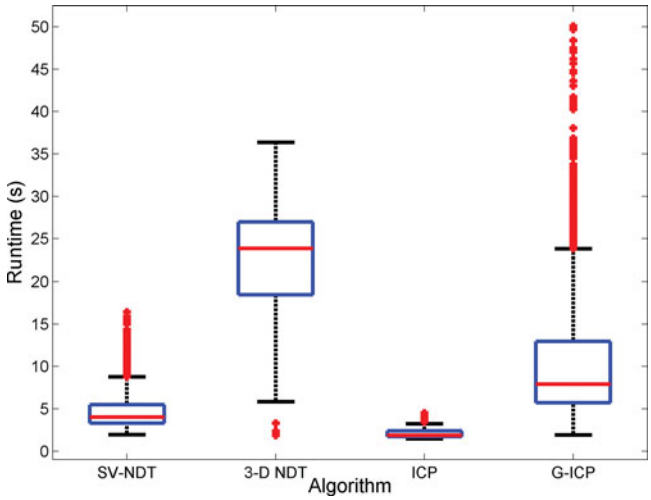


Fig. 25. Runtime of each algorithm.

Table IV. Runtimes of the supervoxel-generating algorithm for each scan pair.

Scan pair number	1	2	3	4	5
Runtime (ms)	62	65	72	74	69

https://doi.org/10.1017/S0263574714002483 Published online by Cambridge University Press

In future work, performance evaluations of the SV-NDT on datasets which are obtained by various sensors or in different environments can be carried out. In addition, to improve the performance of the proposed algorithm more, a method that generates supervoxels more consistently can be investigated to exploit supervoxels for loop detection and initial pose estimation purposes. Finally, the SV-NDT can be applied to 3-D SLAM from 3-D scan registration to the graph optimization.

Acknowledgements

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2013R1A2A1A05005547), in part by the Brain Korea 21 Plus Project, in part by ASRI, in part by the Industrial Foundation Technology Development Program of MOTIE/KEIT [Development of CIRT (Collective Intelligence Robot Technologies)], and in part by Bio-Mimetic Robot Research Center funded by Defense Acquisition Program Administration [UD130070ID].

References

1. P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992).
2. P. Biber and W. Straber, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, Las Vegas, USA, (2003) pp. 2743–2748.
3. A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," *Int. J. Robot. Res.* **26**(10), 1125–1153 (2007).
4. A. Censi and S. Carpin, "HSM3D: Feature-less Global 6DOF Scan-matching in the Hough/Radon Domain," *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, (2009) pp. 3899–3906.
5. R. B. Rusu, N. Blodow and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, (2009) pp. 3212–3217.
6. H. Bulow and A. Birk, "Spectral registration of noisy sonar data for underwater 3D mapping," *Auton. Robots* **30**(3), 307–331 (2011).
7. J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM* **18**(9), 509–517 (1975).
8. M. Greenspan and M. Yurick, "Approximate K-D Tree Search for Efficient ICP," *4th International Conference on 3-D Digital Imaging and Modeling*, Banff, Canada, (2003) pp. 442–448.
9. F. Lu and M. Evangelos, "Robot pose estimation in unknown environments by matching 2D range scans," *J. Intell. Robot. Syst.* **18**(3), 249–275 (1997).
10. L. Armesto, J. Minguez and L. Montesano, "A Generalization of the Metric-Based Iterative Closest Point Technique for 3D Scan Matching," *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, USA, (2010) pp. 1367–1372.
11. A. V. Segal, D. Haehnel and S. Thrun, "Generalized-ICP," *Proceedings of the 2009 Robotics: Science and System Conference*, Seattle, USA, (2009).
12. M. Magnusson, R. Elsrud, L.-E. Skagerlund and T. Duckett, "3D Modelling for Underground Mining Vehicles," *Proceedings of the Conference on Modeling and Simulation for Public Safety*, Linkping, Sweden, (2005).
13. M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal and J. Hertzberg, "Evaluation of 3D Registration Reliability and Speed - A Comparison of ICP and NDT," *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, (2009) pp. 3907–3912.
14. A. Das, J. Servos and S. L. Waslander, "3D Scan Registration Using the Normal Distributions Transform with Ground Segmentation and Point Cloud Clustering," *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, (2013) pp. 2207–2212.
15. T. Stoyanov, M. Magnusson, H. Andreasson and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations," *Int. J. Robot. Res.* **31**(12), 1377–1393 (2012).
16. E. Takeuchi and T. Tsubouchi, "A 3-D Scan Matching Using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, (2006) pp. 3068–3073.
17. C. Ulas and H. Temeltas, "3D multi-layered normal distribution transform for fast and long range scan matching," *J. Intell. Robot. Syst.* **71**(1), 85–108 (2013).
18. B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton and A. Frenkel, "On the Segmentation of 3D LIDAR Point Clouds," *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, (2011) pp. 2798–2805.

19. C. Tongtong, D. Bin, L. Daxue, Z. Bo and L. Qixu, "3D LIDAR-based Ground Segmentation," *Proceedings of the 2011 First Asian Conference on Pattern Recognition*, Beijing, China, (2011) pp. 446–450.
20. F. Moosmann, O. Pink and C. Stiller, "Segmentation of 3D Lidar Data in Non-flat Urban Environments Using a Local Convexity Criterion," *Proceedings of the 2009 IEEE Intelligent Vehicles Symposium*, Xi'an, China, (2009) pp. 215–220.
21. J. Lam, K. Kusevic, P. Mrstik, R. Harrap and M. Greenspan, "Urban Scene Extraction from Mobile Ground Based LIDAR Data," *5th International Symposium on 3D Data Processing, Visualization and Transmission*, Paris, France, (2010) pp. 1–8.
22. Y. Zhou, Y. Yu, G. Lu and S. Du, "Super-segments based classification of 3D urban street scenes," *Int. J. Adv. Robot. Syst.* **9**(248), 1–8 (2012).
23. J. Papon, A. Abramov and M. Schoeler, "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds," *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, Oregon, Portland, (2013) pp. 2027–2034.
24. E. H. Lim and D. Suter, "3D terrestrial LIDAR classifications with super-voxels and multi-scale conditional random fields," *Comput.-Aided Des.* **41**(10), 701–710 (2009).
25. M. Magnusson, "The Three-Dimensional Normal-Distributions Transform - an Efficient Representation for Registration, Surface Analysis, and Loop Detection," *Ph.D. Dissertation*, (Orebro University, Dec. 2009).
26. P. Biber, S. Fleck and W. Strasser, "A Probabilistic Framework for Robust and Accurate Matching of Point Clouds," *26th Pattern Recognition Symposium*, Tübingen, Germany, (2004).
27. J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, (2012) pp. 573–580.
28. B. Douillard, A. Quadros, P. Morton, J. P. Underwood, M. De Deuge, S. Hugosson, M. Hallstrom and T. Bailey, "Scan Segments Matching for Pairwise 3D Alignment," *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, Saint Paul, USA, (2012) pp. 3033–3040.
29. S.-W. Yang, C.-C. Wang and C.-H. Chang, "RANSAC Matching: Simultaneous Registration and Segmentation," *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, USA, (2010) pp. 1905–1912.
30. G. Pandey, J. R. McBride and R. M. Eustice, "Ford campus vision and lidar data set," *Int. J. Robot. Res.* **30**(13), 1543–1552 (2011).