Advanced Engineering Informatics 29 (2015) 930–939

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Automated registration of multi-view point clouds using sphere targets

Dongho Yun [a], Sunghan Kim [b], Heeyoung Heo [b], Kwang Hee Ko [a,*]

[a] School of Mechatronics, Gwangju Institute of Science and Technology, Gwangju 500-712, Republic of Korea
[b] Automation Research, SAMSUNG Heavy Industries Co., Ltd, Geoje, Republic of Korea

ARTICLE INFO

Article history:
Received 23 September 2014
Received in revised form 25 September 2015
Accepted 28 September 2015
Available online 21 October 2015

Keywords:
Point cloud
Fine registration
Rough registration
Iterative closest point registration
Model-based registration
No overlap

ABSTRACT

This paper addresses the problem of automated registration of multi-view point clouds generated by a 3D scanner using sphere targets. First, sphere targets are detected from each point cloud. The centroids of the detected targets in each point cloud are then used for rough registration. Congruent triangles are computed from the centroids for the correspondence among them, with which a rigid body transformation is obtained to bring the two point clouds together as closely as possible. After the initial registration, the two point clouds are further registered by refining the position and orientation of the point clouds using the underlying geometric shapes of the targets. These registration steps are integrated into one system that allows two input point clouds automatically registered with no user intervention. Real examples are used to demonstrate the performance of the point cloud registration.

1. Introduction

Measuring the 3D shape of an object is frequently considered in recent manufacturing processes to maintain the quality of products and to control the fabrication process. It is more important in fabrication processes when multiple parts are assembled. The actual 3D geometric shape of each part can be compared with its CAD (Computer Aided Design) model in advance to identify any potential problem in the assembly and correct it if necessary [1]. In the shipbuilding industry, such demand is growing. A ship or an offshore structure is a custom-ordered product, and few standardized parts are used to build a ship. This means that every part should be uniquely designed and fabricated, and the accurate assembly of the parts is critical for the quality of the final product. To achieve the desired quality, the actual shapes of the fabricated parts need to be evaluated against their CAD models to verify that the models have been made accurately. Here, an efficient method to acquire the 3D shape is to use a laser-based 3D scanner.

A 3D scanner is a *line-of-sight* device. Moreover, the field of view that the scanner covers is limited. Therefore, multiple scans in different positions and directions are typically required to scan a large object. The acquired point clouds must be merged into one data set for correct representation of the shape of the object. The merged point clouds, however, fail to represent the shape of the object because each point cloud is defined with respect to a different coordinate system, as illustrated in Fig. 1. Fig. 1(a) and (b) show two point clouds measured from two different locations and directions. Merging the measured point clouds naively with respect to one reference coordinate system produces misaligned point sets as illustrated in Fig. 1(c). The circled parts in Fig. 1(a) and (b), which should be the same object, are not aligned and instead produce two separate objects. Therefore, aligning point sets with respect to a reference coordinate system is necessary to obtain a set of points that correctly represent the shape of a target object; this process is called *registration*.

Registration is a problem that has attracted the attention of researchers from around the world, and diverse algorithms for the problem have been proposed in the related literature. With the significant growth of data in size and complexity due to the extensive use of high-performance scanners, it is difficult for a human to handle them efficiently for registration. Moreover, the linear extensions of the existing algorithms often fail to handle such data. Therefore, it is necessary to develop a method for registration that can handle data of large size robustly and efficiently and that can run automatically.

In this paper, the problem of registration of point clouds is addressed, and a novel method for automatic registration of two point clouds using sphere targets is proposed. The overall process for registration consists of filtering, sphere detection, initial registration and fine registration. The point clouds, each of which contains sphere targets, are produced by a 3D scanner. They are

* Corresponding author at: School of Mechatronics, Gwangju Institute of Science and Technology, Room 202, 1 Oryeongdong, Bukgu, Gwangju 500-712, Republic of Korea. Tel.: +82 62 715 3225; fax: +82 62 715 2384.

E-mail addresses: lipo123@gist.ac.kr (D. Yun), sh203.kim@samsung.com (S. Kim), heeyoung.heo@samsung.com (H. Heo), khko@gist.ac.kr (K.H. Ko).

http://dx.doi.org/10.1016/j.aei.2015.09.008
1474-0346/© 2015 Elsevier Ltd. All rights reserved.

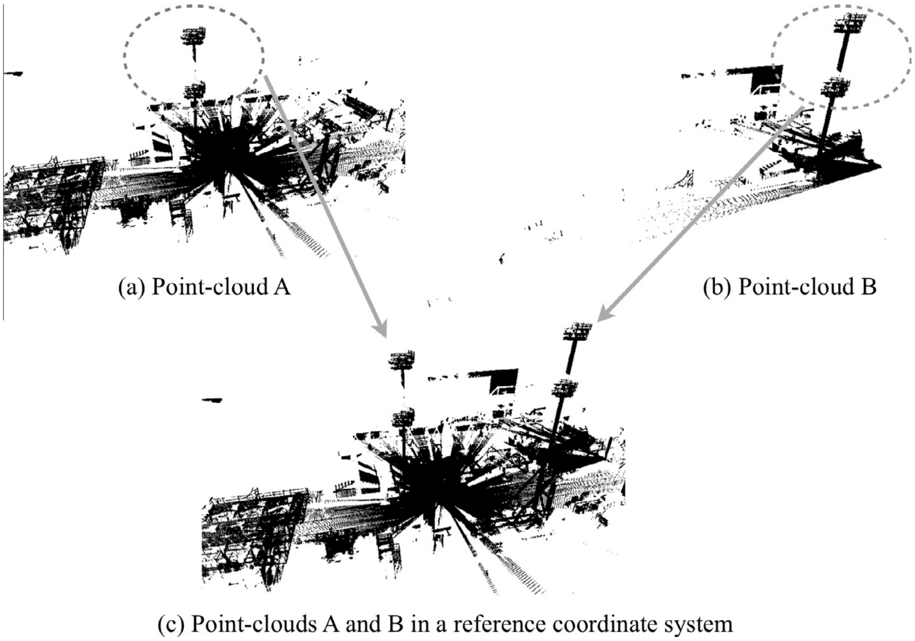


Fig. 1. Point clouds obtained by a 3D scanner. (a) Point cloud A, (b) point cloud B, and (c) point clouds A and B in a global coordinate system with no registration.

filtered to extract points for the targets using differential properties such as normal curvatures and normal vectors estimated from the point clouds. Next, the sphere targets are detected using a sphere detection method. The two point clouds are then registered using the centroids of the sphere targets. This registration is called the initial registration. Next, the positions of the point clouds are finely adjusted to minimize the registration error via a fine registration. The proposed method is different from that in [15]. First, a different filtering method is employed. In this work, differential properties are considered to be filtering criteria. This method is advantageous because more underlying geometric information is obtained, which is used for further registration and data evaluation. Second, a new method for fine registration is proposed. The method is designed to combine modeling and registration in an iterative manner, which produces an accurate registration result.

The primary contribution of this work is as follows. A novel method is proposed to register point clouds with high accuracy using a model-based iterative scheme. In most of the existing methods for the accurate registration of point clouds, the overlapping regions between two point clouds should be detected, and a registration method such as ICP is applied to those overlapping regions. However, the proposed method in this paper does not require such a condition. As long as points are determined to belong to a common object, then fine registration can be performed. Next, two minor improvements are introduced in this work. One is an improved process for detecting spheres from a point cloud. A method of filtering points using differential properties of the underlying geometry is proposed to single out candidate points for spheres, and a selective method is applied to detect spheres from the filtered points. The other is that the concept of parallel computation is introduced to accelerate the detection and registration of point clouds. The methods are implemented in such a way that parallel computation is fully exploited using multiple cores of a CPU, and a significant improvement in computation is achieved.

2. Literature review

Registration of two sets of point clouds consists of computation of the correspondence and the best rigid body transformation. Such computation can be performed using the iterative closest point method in [2]. Since its introduction in the literature, many variants, such as [3–10], have been proposed to improve the performance of the original ICP (Iterative Closest Point) method. There are other types of registration that do not belong to the category of ICP methods. Masuda [11] proposed a method for registration of multiple range images using signed distance fields. Pottmann et al. [12] proposed a method for registration based on the geometric optimization framework of squared distance minimization. A comparative study on the registration of range images was performed in [13]. However, computing the overlapped region and common features is a difficult task, and efficiency in registration can decrease as the number of points increases.

To solve this problem, a target-based approach can be considered. Pre-defined targets are attached to an object, and a scanner scans the object as well as the targets to produce point clouds. Next, the targets are detected from the point clouds and then used to find the correspondence for registration. In [14], three different types of targets (fixed paper, paddle, and spheres) considered in practice were analyzed for registration accuracy. One of the conclusions of the analysis was that the sphere targets are desirable for 3D laser scanning. In [15], a system was proposed for registering point clouds using sphere targets. The system consists of two steps: point processing and registration. The point processing unit filters input points to obtain candidate points on the targets. The points are then fitted with spheres using a quasi-Newton optimization procedure. The centroids of the detected spheres are then used for registration. This method, however, is limited because of three aspects. First, the input points should be pre-processed. Specifically, unstructured data points should be mapped to points on a regular grid, which is an additional step. Moreover, a relatively strong condition that there should exist some free space around the targets is imposed for sphere target placements. Finally, this algorithm only considers a rough registration because sphere detection may not be robust and accurate. This means that a high precision of registration is needed in actual applications.

Automated registration of two point clouds requires the step of finding correspondences between them. For this computation, PCA (Principal Component Analysis) values [16], geometric properties

[17,18], or statistical similarity [19] are considered to obtain correspondences, with which rough registration is performed. For improved accuracy, fine registration, such as ICP-based approaches or the surface interpenetration measure [20] based on a genetic algorithm, can be used. These methods, however, are not robust for scanned point clouds because the patterns of the scanned data are not well organized. Alternately, a semi-automatic registration of point clouds was proposed [21] for better robustness in registration. This is not a fully automatic approach because it requires user input for plane extraction. However, it can avoid the problems of complex and ill-posed registration.

There have been some attempts to develop automated registration algorithms for point clouds. However, full automatic registration without any targets is still problematic for use in practice. The problem can be reduced and made tractable by using sphere targets, and there exist a few papers addressing the problem of automatic registration using sphere targets. However, the current methods using targets are not complete in terms of robustness and registration accuracy because of the difficulty in target detection and registration. In particular, fine registration using targets, which would be advantageous over an approach using the entire point clouds, has not been significantly addressed. Therefore, a new method to handle such problems is needed.

3. Technical approaches

3.1. Target detection

Sphere targets with a fixed radius are used as features of the point clouds. In most cases, the number of points sampled on the sphere targets is less than 0.05% of an input point cloud. Therefore, extracting points on the targets robustly is an important task.

In this section, the process of detecting spheres from a point cloud is presented. It consists of two steps, point filtering and sphere detection. The point-filtering step removes points that do not belong to sphere targets using differential properties. The detection step estimates sphere targets from the filtered data points using a RANSAC (RANdom SAmpling Consensus)-based method.

3.1.1. Filtering point data

The fundamental approach of filtering is to eliminate points for which the underlying geometry is obviously different from that of a sphere target. The geometry at a point is estimated using differential properties, such as normal vectors and curvatures at the point. These properties can be computed by the method of [22]. For completeness of the presentation, a brief summary of this method is introduced.

3.1.2. Estimation of the normal vector and curvature

Consider a point, $\mathbf{p}_0$. First, the points in the neighborhood of $\mathbf{p}_0$ are selected. The number of these points is denoted as $2K$. Then, K sets of $(\mathbf{p}_i, \mathbf{p}_0, \mathbf{p}_j)$ are obtained. Here, $\mathbf{p}_i$ and $\mathbf{p}_j$ are chosen, which yield the largest angle between the two vectors $\mathbf{p}_i - \mathbf{p}_0$ and $\mathbf{p}_j - \mathbf{p}_0$. Then, a quadratic curve is computed for each set. A curve interpolating $(\mathbf{p}_i, \mathbf{p}_0, \mathbf{p}_j)$ is given by $\mathbf{r}_i(u) = \mathbf{a}_{i0} + \mathbf{a}_{i1}u + \mathbf{a}_{i2}u^2$. Here, $\mathbf{r}_i(0) = \mathbf{p}_i$, $\mathbf{r}_i(1) = \mathbf{p}_j$, $\mathbf{r}_i(u_{i0}) = \mathbf{p}_0$ and $u_{i0} = |\mathbf{p}_0 - \mathbf{p}_i|/(|\mathbf{p}_0 - \mathbf{p}_i| + |\mathbf{p}_0 - \mathbf{p}_j|)$, and the unknowns in $\mathbf{r}_i(u)$ can be determined using these conditions. The unit tangent vector, $\mathbf{v}_i$, of $\mathbf{r}_i(u)$ at $\mathbf{p}_0$ is

$$\mathbf{v}_i = \left(\frac{\mathbf{a}_{i1} + 2\mathbf{a}_{i2}u_{i0}}{|\mathbf{a}_{i1} + 2\mathbf{a}_{i2}u_{i0}|}\right). \tag{1}$$

Then, the normal vector $\mathbf{n}_0$ at $\mathbf{p}_0$ can be calculated using the property that $\mathbf{n}_0$ and $\mathbf{v}_i$ should be perpendicular. For all K curves,

$$s = \frac{\sum_{i=1}^{K} (\mathbf{n}_0 \cdot (\mathbf{v}_i - \bar{\mathbf{v}}))^2}{K - 1}. \tag{2}$$

Here, $\bar{\mathbf{v}} = \sum_{i=1}^{K} \mathbf{v}_i / K$. Then, the normal vector at $\mathbf{p}_0$ is given as $\mathbf{n}_0$, the vector that minimizes (2). Consider a $K \times 3$ matrix, $\mathbf{A}$, whose i-th row is $\mathbf{v}_i - \mathbf{v}$. Then, the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$ is the vector that minimizes (2).

The normal curvature of the underlying surface at $\mathbf{p}_0$ in the $\mathbf{v}_i$ direction is computed as follows. Suppose that $\mathbf{n}_i$ is the normal vector of $\mathbf{p}_i(u)$ at $\mathbf{p}_0$. The curvature value of the curve at $\mathbf{p}_0$ is computed to be

$$\kappa_i = \left(\frac{d\mathbf{p}_i}{du} \times \frac{d^2\mathbf{p}_i}{du^2}\right) \Big/ \left|\frac{d\mathbf{p}_i}{du} \times \frac{d^2\mathbf{p}_i}{du^2}\right|. \tag{3}$$

Then, a curvature vector $\kappa_i\mathbf{n}_i$ is obtained. The normal curvature of the surface in the $\mathbf{v}_i$ direction becomes

$$\kappa_{ni} = \kappa_i\mathbf{n}_i \cdot \mathbf{n}_0. \tag{4}$$

3.1.3. Processing point clouds

Using the differential properties at each point, the following three filtering steps are performed.

1. If any of the curvature values are close to zero, then the point is highly likely to be part of a plate or a cylinder. This point is removed.
2. If the signs of the curvature values are not the same, the point is removed because the underlying geometric shape near that point is hyperbolic.
3. If the curvature values are not close to the curvature of the sphere target, the point is removed.

Fig. 2 shows an example of filtered point data after the three filtering passes are completed. These filtering steps can efficiently eliminate points that do not belong to the given sphere targets.

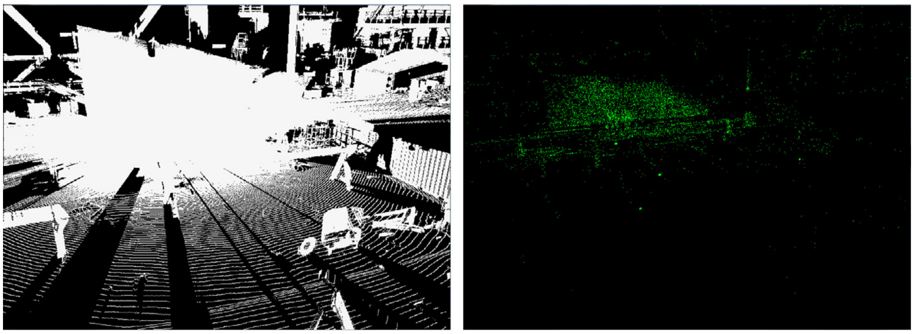
3.1.4. Sphere detection

Spheres can be detected from points by various methods such as least squares-based methods [23–26], a RANSAC-based methods [27,28] and the Hough Transform based method [29]. In this paper, a RANSAC-based method is considered. The RANSAC concept has various advantages [28]. This method is simple to implement and easily applicable to various applications. It is also robust against noise and outliers [30]. Roth and Levine [31] discussed how data with more than 50% of outliers could be processed. The downside of the RANSAC method is long computation time.

In sphere detection, four non-collinear sample points, which are the minimum number of points for modeling a sphere, are needed. Then, a sphere is modeled by computing the centroid and its radius from the four points. Suppose that the four non-collinear points are $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ and $\mathbf{P}_4$. Then, a matrix Eq. (5) is obtained as follows.

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} -\mathbf{P}_1 \cdot \mathbf{P}_1 \\ -\mathbf{P}_2 \cdot \mathbf{P}_2 \\ -\mathbf{P}_3 \cdot \mathbf{P}_3 \\ -\mathbf{P}_4 \cdot \mathbf{P}_4 \end{bmatrix}. \tag{5}$$

Here, $\mathbf{P}_i = (x_i, y_i, z_i),\ i = 1,\ 2,\ 3,\ 4$. The equation is solved using the singular value decomposition method, determining the unknowns A, B, C and D. The centroid and the radius are computed by $a = -A/2,\ b = -B/2,\ c = -C/2$ and $r = (a^2 + b^2 + c^2 - D)^{1/2}$ [27].



(a) Input point-cloud

(b) Filtered point-cloud

Fig. 2. Example of filtered point data. (a) An original input point cloud. (b) A filtered point cloud.

RANSAC-based sphere estimations are presented in [27,28]. Based on the two approaches, the following procedure is developed for this work.

1. Four points are randomly drawn from the input points.
2. A sphere is estimated by solving Eq. (5) using the sampled points.
3. The points in the neighborhood of the estimated sphere are gathered to construct a consensus set. These points are called the inliers to the sphere. Here, the distances from the inliers to the sphere are less than a user-defined value, t.
4. Verify that the number of points in the consensus set is larger than a user-defined tolerance, d.
   (a) If so, the estimated sphere is accepted as an underlying model, and an accurate sphere is estimated using all the inliers in the least squares sense. The procedure is terminated.
   (b) If not, proceed to Step 5.
5. Determine if the number of iterations is less than N_{iter}. If so, return to Step 1. Otherwise, the process stops with a failure notice.

A value of 30% of the designated sphere radius is employed for t, which is empirically determined after the measurement condition of the scanner and the distribution pattern of the point clouds are considered. Step 4 requires a user-defined number, d. If the number of points within the tolerance region t is larger than d, it is concluded that an appropriate sphere has been estimated. This value, d, should be determined in advance based on the point distribution and noise pattern of the points. In this process, a value of 20 was chosen for d. The number N_{iter} is determined based on the probability, p; at least one random sample contains no outliers with probability $p = 0.99$. In this work, N_{iter} is set to be thirty times the number of detected points belonging to the sphere.

The naive application of the proposed procedure would take a long time if the size of the point clouds is large. In this work, a divide-and-conquer approach, which subdivides the entire space enclosing the point cloud into smaller subspaces in an octree-like manner, is introduced to reduce the computation time.

4. Registration

4.1. Initial registration

The detection process may not accurately estimate the spheres that the points are intended to represent. Therefore, the initial registration using the centroids of the detected spheres can be inaccurate, but it is accurate enough for the two point clouds to be roughly aligned.

Consider that k_1 sphere targets from point cloud $\mathbf{P}_{C_1}$ and k_2 sphere targets from point cloud $\mathbf{P}_{C_2}$ have been detected. Suppose that $\mathbf{C}_1 = \{\mathbf{c}_{1f}, f = 1, \ldots, k_1\}$ and $\mathbf{C}_2 = \{\mathbf{c}_{2g}, g = 1, \ldots, k_2\}$; these are the sets containing the centroids of the detected spheres in $\mathbf{P}_{C_1}$ and $\mathbf{P}_{C_2}$, respectively, and $k_1 \leqslant k_2$. Here, $\mathbf{C}_1$ is assumed to be the base, and $\mathbf{C}_2$ is the target. The initial registration process is given as follows.

1. $\mathbf{C}_1^* = Corr_{pts}(\mathbf{C}_1, \mathbf{C}_2)$: This is the operator for computing the points $\mathbf{C}_1^*$ from $\mathbf{C}_1$, where $\mathbf{C}_1^*$ corresponds to $\mathbf{C}_2$.
2. $(\mathbf{R}, \mathbf{T}) = R_{tr}(\mathbf{C}_1^*, \mathbf{C}_2)$: This is the operator for computing the rotation matrix, $\mathbf{R}$, and the translation vector, $\mathbf{T}$ for registering $\mathbf{C}_2$ and $\mathbf{C}_1^*$.
3. $\mathbf{P}_{C_2}^* = Transf(\mathbf{P}_{C_2};\ \mathbf{R}, \mathbf{T})$: This function transforms $\mathbf{P}_{C_2}$ using $\mathbf{R}$ and $\mathbf{T}$. Namely, $\mathbf{P}_{C_2}^* = \mathbf{R} \cdot \mathbf{P}_{C_2} + \mathbf{T}$.

Step 1 is a critical step for the success of registration. Because no *a priori* information for the correspondence between $\mathbf{C}_1$ and $\mathbf{C}_2$ is available, the function $Corr()$ should establish the best correspondence between them. In this work, the following method is used. Suppose that $d(\mathbf{c}_{2i}, \mathbf{c}_{2j})$ is the function that computes the minimum distance between $\mathbf{c}_{2i}$ and $\mathbf{c}_{2j}$.

(1) Construct a set of pairs $\mathbf{J} = \{(\mathbf{c}_{2i}, \mathbf{c}_{2j}) : i, j = 1, \ldots, k_2, i \neq j\}$.
(2) Select a pair $(\mathbf{c}_{2im}, \mathbf{c}_{2jm})$, that yields the maximum distance d in $\mathbf{J}_d$. Here, $\mathbf{J}_d = \{d(\mathbf{c}_{2im}, \mathbf{c}_{2jm}), i, j = 1, \cdots, k_2, i \neq j\}$.
(3) Choose the third point $\mathbf{c}_{2km}$ from $\mathbf{C}_2$ such that the area of the triangle defined by $(\mathbf{c}_{2im}, \mathbf{c}_{2jm}, \mathbf{c}_{2km})$ is the largest.
(4) Search for three points $\mathbf{c}_{1i}^*, \mathbf{c}_{1j}^*$, and $\mathbf{c}_{1k}^*$ from $\mathbf{C}_1$ such that the triangle formed by the points is congruent with that of $(\mathbf{c}_{2im}, \mathbf{c}_{2jm}, \mathbf{c}_{2km})$.
  (4)-1 If such points are found, go to Step (5).
  (4)-2 Otherwise, $\mathbf{J} = \mathbf{J} - \{(\mathbf{c}_{2im}, \mathbf{c}_{2jm})\}$ and go to Step (2).
(5) Find the correspondence between the vertices of the two triangles.

In Step (4), an algorithm is designed to reduce the search domain. A pair of points yielding the maximum distance computed in Step 2 is searched in $\mathbf{C}_1$. This pair is denoted as $(\mathbf{c}_{1i}^*, \mathbf{c}_{1j}^*)$. Next, the angles θ_{2i} and θ_{2j} of the triangle $(\mathbf{c}_{2im}, \mathbf{c}_{2jm}, \mathbf{c}_{2km})$ at $\mathbf{c}_{2im}$ and $\mathbf{c}_{2jm}$ are computed. These angles are then used to restrict the search domain for the third point $\mathbf{c}_{1k}^*$ in $\mathbf{C}_1$ that forms a triangle congruent to the triangle of $(\mathbf{c}_{2im}, \mathbf{c}_{2jm}, \mathbf{c}_{2km})$ along with the distances $d(\mathbf{c}_{2im}, \mathbf{c}_{2km})$ and $d(\mathbf{c}_{2jm}, \mathbf{c}_{2km})$.

Once the correspondence between the two sets is established, a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{T}$ can be obtained using the method by Horn [32].

4.2. Fine registration

The initial registration step uses the centroids of the estimated sphere targets. Therefore, the registration quality that the initial registration method can guarantee is entirely dependent on the accuracy of the sphere estimation. The accuracy, however, cannot be assured because the estimation is performed using measured points that may contain noise. Therefore, after the initial registration, the point clouds should be finely adjusted to obtain more accurate registration. This second registration is called fine registration.

For fine registration, ICP-based methods are typically employed. However, to apply such methods, one condition should hold that two point clouds overlap, and the correspondence relations between the points in the overlapped region are utilized for registration. Without this condition, the ICP-based methods cannot be correctly performed. However, this condition cannot always be satisfied. It is difficult to extract the points in the overlapped region with no user-provided input because there exists no information on which regions of the two point clouds overlap despite the initial registration of the two point clouds. Moreover, the two point clouds are so generated that no overlap exists, or only a small fraction of points are obtained in the overlapped region. In this case, the registration methods based on the overlapped region cannot be used at all. Next, obtaining appropriate correspondences between two sets of points in the overlapped region is another challenge. Although an overlapped region has been identified and points in the region are extracted, the correct correspondence cannot be established because the traditional ICP only considers the positions of points and the minimum Euclidean distances without considering the underlying geometric shape. A typical example illustrating this situation is given in Fig. 3, which shows two sets of points in the overlapped region. One set contains points in the vertical direction; the other set contains points in a nearly horizontal direction. If the positions of the points are considered, no unique correspondence can be established. Translating one set of points horizontally would result in a similar situation. Therefore, the point-based ICP cannot correctly find the best correspondence for registration, leading to failed registration.

In this paper, a new registration method is proposed to solve these problems. The method does not require that two point clouds overlap. An analytical sphere model is estimated as its underlying geometry, and the registration is performed with respect to the estimated model. There are two papers addressing the integration of modeling and registration in one process [33,34]. Both papers present a similar approach in that modeling and registration are formulated as one optimization problem, but they differ in the solution method. In [33], the squared distance minimization (SDM) scheme was considered, and a quasi-Newton method was employed to find the parameters, whereas the Levenberg–Marquardt method was used in [34]. The proposed method, termed the model-based method, combines modeling and registration using the ICP framework. It does not require a complicated mathematical formulation for optimization including derivative calculation and is designed to be generalized for handling various geometric shapes other than spheres.

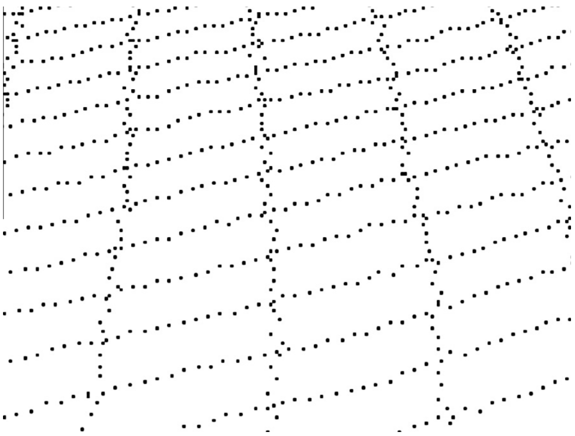


Fig. 3. Example of point distribution in the overlapped region.

4.2.1. Model-based method

The core idea of the model-based method is to estimate the underlying geometric shape from the updated registered point clouds and to register the point clouds to the shape iteratively. Unlike the ICP-based methods that use the point clouds directly, the proposed method considers the underlying geometric shape as a reference and refines it in each iteration. Fig. 4 illustrates the idea of the proposed method.

Suppose that two point clouds have been registered using the centroids of the detected spheres. $\mathbf{m}_{ai}$ is a set of points on the i-th detected sphere from the point cloud $\mathbf{P}_{C_a}$. $sp(\mathbf{m}_i)$ is the operator of sphere estimation from $\mathbf{m}_i$. $\mathbf{S}_i$ is the estimated sphere. The pseudo-code for the model-based registration method is given as follows.

Suppose that $c\mathbf{P}_1 = \{\mathbf{m}_{1i}, 1 \leqslant i \leqslant k_2\}$ and $c\mathbf{P}_2 = \{\mathbf{m}_{2j}, 1 \leqslant j \leqslant k_2\}$.

```
1. Compute $c\mathbf{P} = \{\mathbf{m}_{12i} = \mathbf{m}_{1i} \cup \mathbf{m}_{2i}, 1 \leqslant i \leqslant k_2\}$.
2. Compute $\mathbf{S}_i = sp(\mathbf{m}_{12i}),\ 1 \leqslant i \leqslant k_2$.
3. $(\mathbf{R},\mathbf{T}) = R_{tr}(\mathbf{S}_i, \mathbf{m}_{2i})$ for all $i$.
4. $\mathbf{m}_{2i} = Transf(\mathbf{m}_{2i};\ \mathbf{R},\mathbf{T}),\ 1 \leqslant i \leqslant k_2$.
5. $E = Err(\mathbf{S}_i;\ \mathbf{m}_{1i}, \mathbf{m}_{2i})$.
6. Go to 1 while $E > \delta$.
```

Lines 1 and 2 are the core steps in the proposed method. In Line 1, two point sets from $c\mathbf{P}_1$ and $c\mathbf{P}_2$ are merged into one point set with respect to one coordinate system. In Line 2, a model approximating the merged point set is computed, which is used as a reference in the registration calculation. This model is approximated by minimizing the error to the point set. In Line 3, the point set that produces a larger error to the model is selected and registered onto

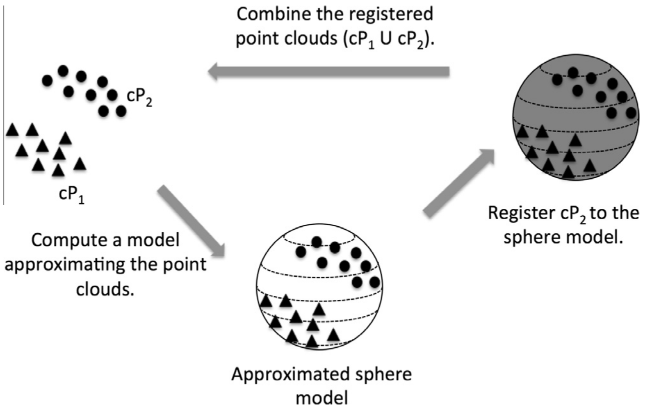

Combine the registered point clouds (cP$_1$ U cP$_2$).
cP$_2$
cP$_1$
Register cP$_2$ to the sphere model.
Compute a model approximating the point clouds.
Approximated sphere model

Fig. 4. Graphical illustration of the model-based method. An approximated sphere is computed from the point clouds $c\mathbf{P}_1$ and $c\mathbf{P}_2$. The point cloud $c\mathbf{P}_2$ is then registered to the sphere. The union of $c\mathbf{P}_1$ and the registered $c\mathbf{P}_2$ is then computed for the next iteration.

the model. Then, the point set is updated in Line 4. After the loop is terminated, $\mathbf{P}^*_{C_2} = Transf(\mathbf{P}_{C_2};\ \mathbf{R}, \mathbf{T})$ is computed. For $\mathbf{S}_i = sp(\mathbf{m}_{12i})$ the least squares fitting method is employed. One best sphere can be estimated in this step and is then used as a true underlying sphere in the registration process. R_{tr} is an operator for computing a rotation matrix and a translation vector that register the points onto the sphere. For this computation, an orthogonal projection method [35] and the quaternion-based method [32] are employed. In Step 6, δ is a user-defined tolerance.

4.2.2. Analysis

The iterative process of sphere estimation and registration is a main part of the proposed method. The convergence of the method is verified as follows.

Consider the value $E^b = Err(\mathbf{S}^b_i;\ \mathbf{m}_{1i}, \mathbf{m}_{2i})$, which is the sum of the Euclidean distances between the sphere, $\mathbf{S}^b_i$, and the points, $\mathbf{m}_{1i}$ and $\mathbf{m}_{2i}$, for the b-th iteration. After Step 2, E^b is obtained. Steps 3–5 adjust the position and orientation of $\mathbf{m}_{2i}$, yielding $\mathbf{m}^*_{2i}$ and subsequently resulting in a reduced E^b. This new E^b is denoted as E^{b*} with $E^{b*} \leqslant E^b$. At the $b+1$-th iteration, $\mathbf{m}_{2i} = \mathbf{m}^*_{2i}$ and a new sphere $\mathbf{S}^{b+1}_i$ is obtained based on $\mathbf{m}_{1i}$ and $\mathbf{m}_{2i}$. Next, the updated value of E^{b+1} is computed. Consider the relation between E^{b+1} and E^b. Because $\mathbf{S}^{b+1}_i$ is obtained using $\mathbf{m}_{1i}$ and $\mathbf{m}_{2i}$, which have smaller error values than those before transformation, it can be found that $E^{b+1} \leqslant E^{b*}$. Therefore, $E^{b+1} \leqslant E^b$ is satisfied. This relation indicates that the algorithm converges. The convergence of the proposed method is demonstrated using five examples.

Consider six point clouds, $S1$, $S2$, $S3$, $S4$, $S5$ and $S6$. The numbers of points in each point cloud are 14,748,777, 3,818,803, 5,059,272, 3,447,131, 4,516,864 and 4,385,237, respectively. In these tests, $\delta = 10(\text{mm})$ is used as a termination condition. The five test cases are $(S1,\ S3)$, $(S1, S4)$, $(S2, S3)$, $(S3, S5)$ and $(S4, S6)$.

The error values for each test are plotted as shown in Fig. 5, where the horizontal axis is the number of iterations and the vertical axis is the error at each iteration. As shown in the figure, the error values monotonically decrease as the iteration continues. However, they decrease considerably at the first few iterations and converge to certain values gradually afterward. Although the test results show a somewhat similar decrease pattern of error, the rate of convergence is different for each case. This convergence rate depends on the distribution patterns, the numbers of points detected on each sphere and the positions of the detected spheres, which are not analytically characterized. Therefore, it is hard to find a general model representing the convergence rate.

4.3. Extension to multiple-point-cloud registration

The proposed model-based registration method can be extended to the registration of multiple point clouds. Suppose that n^{pc} point clouds are given. The successive pair-wise registration may suffer from error accumulation; when two point clouds are registered, a certain amount of error is induced. When the next pair of point clouds is registered, the error in the first pair is directly passed on to the registration of the next pair. Therefore, the error of the first and the last point clouds may become larger than expected.

To solve this problem, all point clouds should be considered simultaneously in the registration process. The extension of fine registration for two point clouds to fine registration for multiple point clouds is straightforward. The algorithm remains the same except for Step 1. $c\mathbf{P}$ is the union of all the input point clouds. From Steps 2 to 4, the rigid body transformations for each point cloud against the computed sphere $\mathbf{S}_i$ are computed and applied. The overall error is computed at Step 5. This process is repeated until the error falls below the user-defined tolerance.

4.4. Implementation

The proposed procedure is implemented as a software system. The structure of the system is shown in Fig. 6. The arrows indicate the data flow between the modules.

The kernel module takes user inputs, reads and writes point clouds, maintains data structures, including Kd-trees for storing and managing point clouds, and provides various vector and matrix operators. In the filtering module, an efficient method [36] is employed to search for the nearest neighboring points. The visualization module provides interactive functions for displaying point clouds, navigating the points, zooming in/out on the points, and rotating/translating the points. For these functions, OpenGL was used. The sphere detection module detects spheres from the point clouds using the RANSAC-based method and the least-squares method introduced in this paper. The registration module performs the initial and fine registration for point clouds. For the implementation, Visual C++ 2010 was used as a development language on a computer system with an Intel Core i7 CPU (3.5 GHz) and 32 GB of RAM.

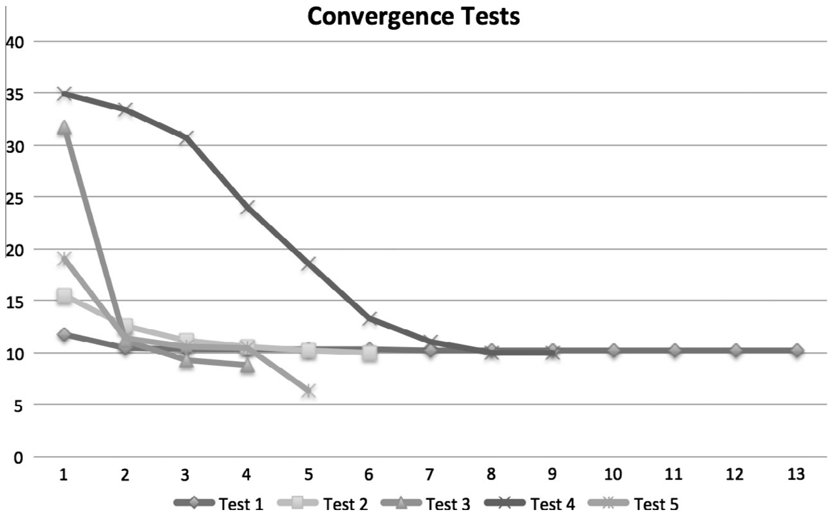


Fig. 5. Convergence of errors with respect to the number of iterations. The horizontal and vertical axes are the number of iterations and the error, respectively.

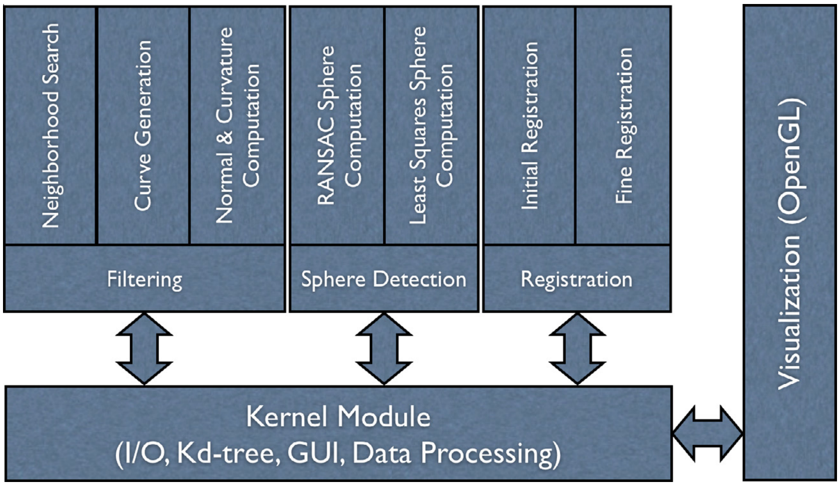


Fig. 6. Structure of the program for the registration of point clouds.

5. Examples

5.1. Validation

Various examples were used to test the proposed method. Two point clouds, *S*4 and *S*6, were selected, as shown in Fig. 7. The numbers of points of each point cloud were 3,447,131 and 4,358,042, respectively.

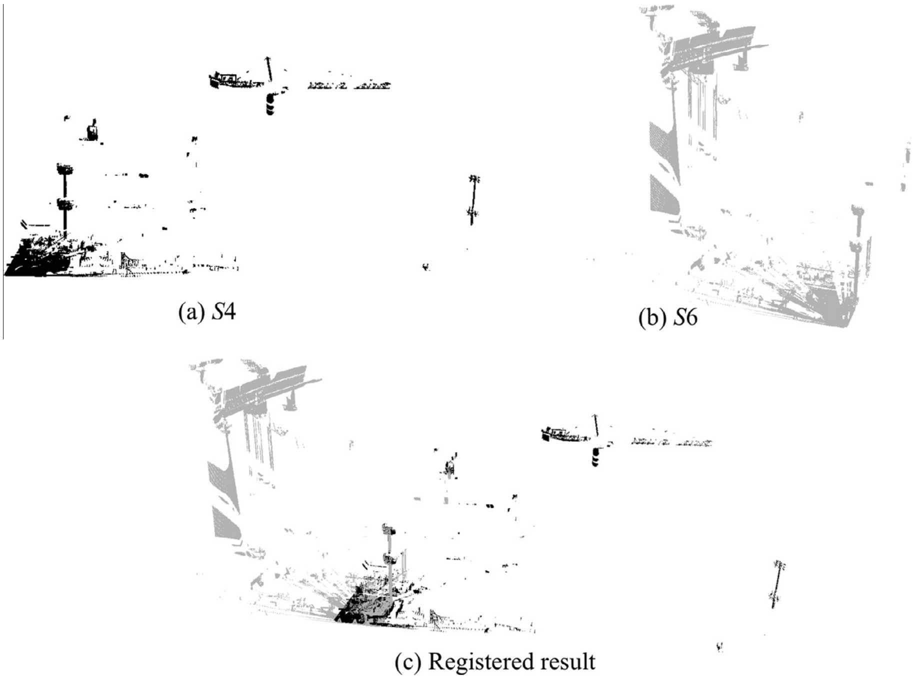
The input point clouds were filtered leaving, 1912 and 2970 points for *S*4 and *S*6, respectively. From *S*4, five spheres were detected. From point cloud *S*6, four spheres were detected. Examples of detected spheres of the registered point clouds are shown in Fig. 8.

The initial registration module selected three centroids for *S*4 and another three centroids for *S*6. These two sets of centroids were chosen by the algorithm in Section 3.1. Before the initial registration, the average distance between the two sets of centroids was 155061.32 mm. Based on the selected centroids, the initial registration computed a rotation matrix and a translation vector for registration, which were then applied to *S*6 to produce the registered centroids. The average error was reduced to 624.78 mm.

The registered spheres were provided as input to the fine registration module. In this registration step, the error is computed using the equation in Section 3.2.1. Before the fine registration the value was 1.77 mm. After the fine registration, the error was reduced to 0.71 mm.

The registered point clouds are shown in Figs. 7(c) and 8. For this registration, the total computation times for processing *S*5 and *S*6 were 63 and 87 s, respectively. The initial and fine registration times were less than 1 s.

As a next example, registrations of five point clouds (*S*1, *S*2, *S*3, *S*5 and *S*6) were tested. The numbers of points of each point cloud and the numbers of points after filtering are summarized in Table 1. Registration was performed for two pairs of point


(a) *S*4

(b) *S*6

(c) Registered result

Fig. 7. Example of registration. (a) Point cloud *S*4, (b) point cloud *S*6, and (c) registered point clouds.

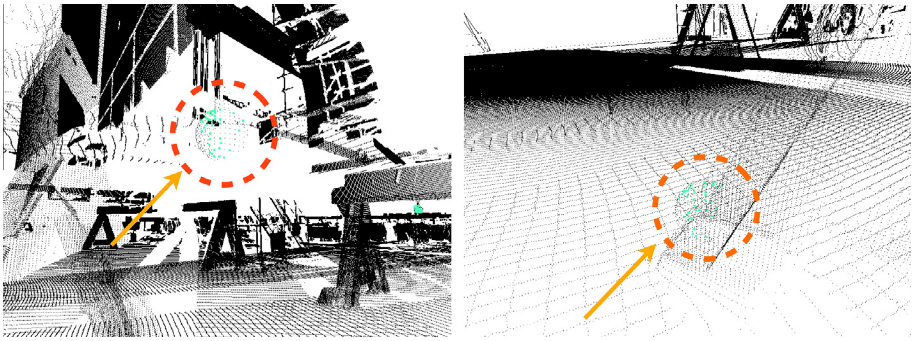


Fig. 8. Detected spheres in *S*4 and *S*6. The two point clouds have been registered using the spheres. The arrows indicate the detected sphere targets.

Table 1
The numbers of points of each point cloud before and after filtering.

No. of points	*S*1	*S*2	*S*3	*S*5	*S*6
Before	14,748,777	3,818,803	5,059,272	4,516,864	4,358,042
After	8899	2407	2031	3284	2970

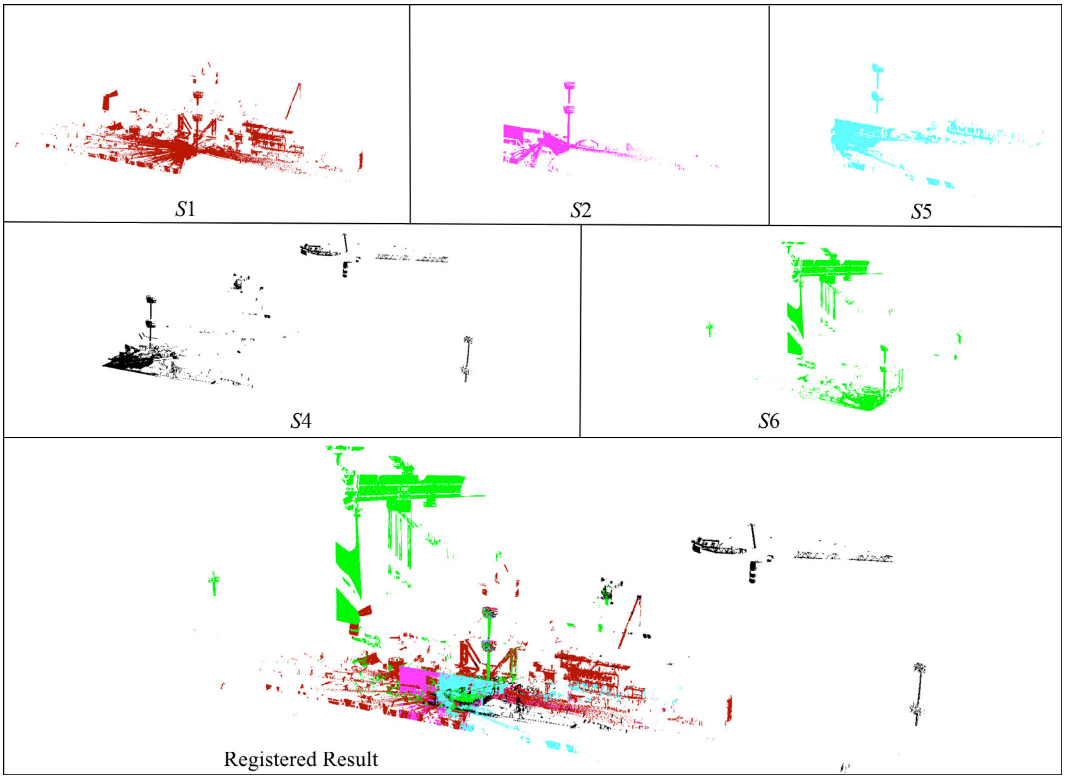


Fig. 9. Registration result of point clouds *S*1, *S*2, *S*4, *S*5 and *S*6.

clouds in a sequential manner. Namely, *S*1 was considered to be a base, and each remaining point cloud was registered with respect to *S*1.

The total computation time for registering the five point clouds was 600.75 s. Most of the time was spent on filtering; the registration took a few seconds. The registered result is given in Fig. 9.

5.2. Comparison

In this section, the proposed method is compared with two existing commercial programs bundled with Leica and FARO scanners. One is Leica Cyclone, a 3D point cloud processing program. The other is FARO Scene for FARO scanners. They provide functions for 3D point cloud processing and registration designed for the 3D

Table 2
Comparison of the proposed method with the existing commercial software for sphere detection.

Method	*S*1		*S*2		*S*3	
	No.	Time (s)	No.	Time (s)	No.	Time (s)
Proposed	5	35.3	3	9.6	4	12.11
Cyclone	5	49.12	3	67.06	4	78.19
Scene	5	61.20	3	20.14	2	39.03
	*S*4		*S*5		*S*6	
	No.	Time (s)	No.	Time (s)	No.	Time (s)
Proposed	5	9.73	5	15.48	5	30.12
Cyclone	5	90.09	5	63.18	4	63.08
Scene	5	19.04	5	23.12	3	17.20

scanners. All the programs register point clouds using sphere targets. Unfortunately, however, the core algorithms that the two commercial programs are based on cannot be disclosed because of company policy.

First, the performance of sphere detection is compared using six point clouds, *S*1, *S*2, *S*3, *S*4, *S*5, and *S*6. Note that Leica Cyclone does not provide a function for automatic sphere detection and makes the user select a point on a sphere manually. Table 2 summarizes the test results of the three programs. Note that the computation time for Leica Cyclone includes the time elapsed for manual selection. As shown in the table, the proposed method found more spheres than FARO Scene and the same number of spheres as Leica Cyclone. This means that the proposed method can detect spheres robustly. The computation times for sphere detection by the proposed method are better than those of the two products. It can be considered that the run time advantage of the proposed method is mainly due to the exploitation of parallel computing and data structures, not to the proposed algorithm itself. However, the proposed method is built upon parallel computation and data structures, which are tightly combined into one optimal package. Therefore, the performance of target detection becomes superior to that of existing commercial methods in terms of speed and detection rate.

Next, the time for registration and the registration errors are compared. In this comparison, pair-wise registration is considered. The accuracy of the registration software bundled with the commercial scanners is suitable for practical use. Therefore, achieving accuracy up to that level would be satisfactory. The final registration errors by the three methods, given in Table 3, show that the proposed method can produce reasonable registration results. In addition, the proposed method is the fastest among them.

These experiments show that the proposed method is equivalent to and/or superior to the existing software packages in terms of performance and robustness.

Table 3
Comparison of the proposed method with the existing commercial software for registration. Here, *E* stands for the average error.

Method	Test 1		Test 2		Test 3	
	E (mm)	Time (ms)	*E* (mm)	Time (ms)	*E* (mm)	Time (ms)
Proposed	9.28	11	7.04	12	20.06	11
Cyclone	9.17	13	6.85	13	20.26	26
Scene	9.16	12	7.16	24	21.50	16
	Test 4		Test 5			
	E (mm)	Time (ms)	*E* (mm)	Time (ms)		
Proposed	2.47	12	67.28	12		
Cyclone	1.46	13	67.23	13		
Scene	1.49	20	67.18	16		

6. Conclusion

In this paper, the problem of registration of point clouds is addressed, and a new method for registering point clouds using sphere targets is proposed. The procedure consists of filtering and registration. The filtering step processes a point cloud, eliminates unnecessary points and detects sphere targets. The registration step registers two input point clouds using the detected targets through the initial and fine registration. The initial registration uses the centroids of the targets and brings the point clouds together as closely as possible. The fine registration refines the registration to improve the accuracy of registration. In particular, an innovative method for fine registration is proposed utilizing the underlying component geometry during registration.

Theoretically, it would be possible for a CAD model to be used instead of sphere targets for registration. However, to use it as a replacement for sphere targets, the point clouds should be processed to detect the CAD model from an input point cloud. Once the model is detected, it could be used as features for registration. A CAD model may consist of various parts. Robustly detecting a CAD model from a point cloud is, however, a very difficult process in general because the model may contain complex shapes that are difficult to detect from a point cloud. In contrast, sphere targets are easy to detect robustly because of their well-defined geometric properties. Therefore, using sphere targets in the registration process, instead of using a CAD model, is preferable.

Three issues need to be considered before the proposed automated registration system can be employed in a practical application. First, the critical aspect of the target-based registration is robustness in the target detection. The automated registration will fail when targets are not properly detected. Particularly, if the measured point clouds do not contain the minimum number of targets necessary for registration, the registration cannot be performed because the proper correspondence between the targets cannot be established. To handle such a case, a manual registration method must be introduced as a supplemental method to the automated registration system. Next, the range of points for the evaluation of registration accuracy needs to be extended. In this work, the accuracy of the initial and fine registration was evaluated using the Euclidean distances between the centroids of targets and those distances of points to the targets. However, these methods only use the targets for evaluation and do not consider the accuracy of registration for the points far away from the targets. A reasonable measure that considers all the points is necessary. Finally, the number of points of a typical input point cloud is more than a few million, possibly reaching into the tens of millions. Therefore, the processing of such point clouds would take a long time, and a reduction of the processing time is required. These three issues are recommended for future work.

Acknowledgement

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0010099).

References

[1] J. Sun, K. Kiekata, H. Yamato, N. Nakagaki, A. Sugawara, Efficient point cloud data processing in shipbuilding: reformative component extraction method and registration method, J. Comput. Des. Eng. 1 (2014) 202–212.

[2] P.J. Besl, N.D. McKay, A method for registration of 3D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256.

[3] P. Bergstrom, O. Edlund, I. Soderkvist, Repeated surface registration for on-line use, Int. J. Adv. Manuf. Technol. 54 (2011) 677–689.

[4] C.S. Chua, R. Jarvis, 3D free-form surface registration and object recognition, Int. J. Comput. Vis. 17 (1) (1996) 77–99.

[5] C.S. Chua, R. Jarvis, Point signatures: a new representation for 3D object recognition, Int. J. Comput. Vis. 25 (1) (1997) 63–85.

[6] S. Du, N. Zheng, L. Xiong, S. Ying, J. Xue, Scaling iterative closest point algorithm for registration of m–d point sets, J. Vis. Commun. Image Represent. 21 (2010) 442–452.

[7] Y. Liu, Free form shape registration using the barrier method, Comput. Vis. Image Underst. 114 (2010) 1004–1016.

[8] G.C. Sharp, S.W. Lee, D.K. Wehe, ICP registration using invariant features, IEEE Trans. Pattern Anal. Mach. Intell. 24 (1) (2002) 90–102.

[9] Y.W. Sun, J.T. Xu, An optimal matching algorithm based on rough localization and exact adjustment, Key Eng. Mater. 291–292 (2005) 239–256.

[10] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, Int. J. Comput. Vis. 13 (2) (1994) 119–152.

[11] T. Masuda, Registration and integration of multiple range images by matching signed distance fields for objet shape modeling, Comput. Vis. Image Underst. 87 (2002) 51–65.

[12] H. Pottmann, S. Leopoldseder, M. Hofer, Registration without ICP, Comput. Vis. Image Underst. 95 (2004) 54–71.

[13] J. Salvi, C. Matabosch, D. Fofi, J. Forest, A review of recent range image registration methods with accuracy evaluation, Image Vis. Comput. 25 (5) (2007) 578–596.

[14] B. Becerik-Gerber, F. Jazizadeh, G. Kavulya, G. Calis, Assessment of target types and layouts in 3d laser scanning for registration accuracy, Autom. Constr. 20 (2011) 649–658.

[15] M. Franaszek, G.S. Cheok, C. Witgall, Fast automatic registration of range images from 3d imaging systems using sphere targets, Autom. Constr. 18 (2009) 265–274.

[16] C. Kim, H. Son, C. Kim, Fully automated registration of 3d data to a 3d cad model for project progress monitoring, Autom. Constr. 35 (2013) 587–594.

[17] K.-H. Bae, D.D. Lichti, A method for automated registration of unorganized point clouds, ISPRS J. Photogramm. Rem. Sens. 63 (2008) 36–54.

[18] P.W. Theiler, J.D. Wegner, K. Schindler, Keypoint-based 4-points congruent sets – automated marker-less registration of laser scans, ISPRS J. Photogramm. Rem. Sens. 96 (2014) 149–163.

[19] E.G. Parmehr, C.S. Fraser, C. Zhang, J. Leach, Automatic registration of optical imagery with 3d lidar data using statistical similarity, ISPRS J. Photogramm. Rem. Sens. 88 (2014) 28–40.

[20] L. Silva, O.R.P. Bellon, K.L. Boyer, Multiview range image registration using the surface interpenetration measure, Image Vis. Comput. 25 (1) (2007) 114–125.

[21] F. Bosche, Plane-based registration of construction laser scans with 3d/4d building models, Adv. Eng. Inform. 26 (2012) 90–102.

[22] D. OuYang, H.-Y. Feng, On the normal vector estimation for point-cloud data from smooth surfaces, Comput. Aid. Des. 37 (2005) 1071–1079.

[23] S.J. Ahn, W. Rauh, H.J. Warnecke, Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbolar, and parabola, Pattern Recogn. 34 (2001) 2283–2303.

[24] D. Clouse, C. Padgett, Estimating the position of a sphere from range images, in: IEEE, Aerospace Conference Proceedings, IEEE, 2002.

[25] H. Spath, Least-square fitting with spheres, J. Opt. Theory Appl. 96 (1998) 191–199.

[26] C. Witzgall, G.S. Cheok, A.J. Kearsley, Recovering circles and spheres from point data, in: alt, f. b., fu, m. c., golden, b. l., (eds.), Perspectives in Operations Research, Springer, New York, 2006.

[27] S. Garcia, Fitting Primitive Shapes to Point-Clouds for Robotic Grasping, Master's thesis, Royal Institute of Technology, Sweden, 2009.

[28] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point-cloud shape detection, Comput. Graph. Forum 26 (2) (2007) 214–226.

[29] A. Abuzaina, M.S. Nixon, J.N. Carter, Sphere detection in kinect point-clouds via the 3d hough transform, Lect. Notes Comput. Sci. 8048 (2013) 290–297.

[30] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.

[31] G. Roth, M.D. Levine, Extracting geometric primitives, CVGIP: Image Underst. 58 (1) (1993) 1–22.

[32] B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, J. Opt. Soc. Am., Ser. A 4 (4) (1987) 629–642.

[33] Y. Liu, H. Pottmann, W. Wang, Constrained 3d shape reconstruction using a combination of surface fitting and registration, Comput. Aid. Des. 38 (2006) 572–583.

[34] T. Rabbani, S. Dijkman, F. van den Heuvel, G. Vosselman, An integrated approach for modeling and global registration of point clouds, ISPRS J. Photogramm. Rem. Sens. 61 (2007) 355–370.

[35] K. Ko, T. Sakkalis, Orthogonal projection of points in CAD/CAM applications: an overview, J. Comput. Des. Eng. 1 (2) (2014) 116–127.

[36] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, A.Y. Wu, An optimal algorithm for approximate nearest neighbor searching in fixed dimensions, J. ACM 45 (6) (1998) 891–923.