Research Article

Representation of 3D Environment Map Using B-Spline Surface with Two Mutually Perpendicular LRFs

Rui-Jun Yan,¹ Jing Wu,¹ Ji Yeong Lee,² and Chang-Soo Han²

¹Department of Mechatronics Engineering, Hanyang University, Ansan, Gyeonggi-do 426-791, Republic of Korea

²Department of Robot Engineering, Hanyang University, Ansan, Gyeonggi-do 426-791, Republic of Korea

Correspondence should be addressed to Chang-Soo Han; cshan@hanyang.ac.kr

Received 19 November 2014; Revised 27 March 2015; Accepted 2 April 2015

Academic Editor: Yongsheng Ou

Copyright © 2015 Rui-Jun Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a map representation method of three-dimensional (3D) environment by using B-spline surfaces, which are first used to describe large environment in 3D map construction research. Initially, a 3D point cloud map is constructed based on extracted line segments with two mutually perpendicular 2D laser range finders (LRFs). Then two types of accumulated data sets are separated from the point cloud map according to different types of robot movements, continuous translation and continuous rotation. To express the environment more accurately, B-spline surface with covariance matrix is proposed to be extracted from each data set. Due to the random movements, there must be overlap between extracted B-spline surfaces. However, merging of two overlapping B-spline surfaces with different distribution directions of their control points is a complex problem, which is not well addressed by far. In our proposed method, each surface is divided into overlap and nonoverlap. Then generated sample points with propagated uncertainties from one overlap and their projection points located on the other overlap are merged using the product of Gaussian probability density functions. Based on this merged data set, a new surface is extracted to represent the environment instead of the two overlaps. Finally, proposed methods are validated by using the experimental result of an accurate representation of an indoor environment with B-spline surfaces.

1. Introduction

Two-dimensional (2D) features-based simultaneous localization and mapping (SLAM) is the problem of correcting a robot position and building an environment map by using the extracted features in unknown environment. In the past decade, researchers have investigated many issues in 2D SLAM such as feature characterization [1–3], data association [4–6], and loop closing [7–9]. Even though much work has increased the accuracy of constructed 2D environment map, only the 2D geometrical parameters of the objects in three-dimensional (3D) environment have been obtained.

Recently, several SLAM works have constructed a 3D point cloud map of a real environment to show the geometrical shape of the real objects [10, 11]. Based on constructed 3D point cloud map, navigation [12] and path planning [13] research have been done in a 3D environment. To build the 3D point cloud map, a 3D sensor is necessary to obtain the raw sensor data. Most of the 3D LRF system is composed

of a 2D LRF and a mechanism system, such as a vertical rotating system [14], a pitch motion system [15], and a spring-mounted system [16]. The obtained 3D raw sensor data from these sensors should be organized to represent the environment map. The iterative closest point (ICP) algorithm [17] is the most well-known method for registration of 3D shapes described either geometrically or with point clouds. Extended ICP algorithms [18, 19] have been used to represent outdoor terrain maps. However, since the environment is represented with scanned sensor data, large storage space is needed in the experimental process.

To represent the environment well, the most commonly used feature is a plane, which has been considered to be extracted from the point cloud map in current research about 3D map construction. There are many plane extraction methods [20, 21], in which the planes have been chosen as landmarks to build an environment map. Nevertheless, a plane is not a good choice to represent 3D map in consideration of the diversity of real objects.

FIGURE 1: Constructed 3D point cloud map by using two mutually perpendicular LRFs, which is expressed with different colors according to their height. Robot trajectory is plotted with green triangles, and 2D map is plotted with blue lines.

In this paper, two mutually perpendicular 2D LRFs are used to build the 3D point cloud map. To correct the position of a mobile robot, line segments are extracted from the sensor data obtained from the horizontal LRF. Improved extended Kalman filter (IEKF) SLAM algorithm is applied to update the position of robot by using matched feature pair. Based on the accurate position of robot, point cloud map is constructed by using the sensor data obtained from the vertical LRF, shown in Figure 1.

B-spline surfaces with a small number of parameters are extracted from the point cloud map to represent the 3D environment because of its powerful representation of various objects with complex geometrical shape. Only small storage space is needed to store the small number of parameters of B-spline surface instead of large amounts of point cloud. It can make the SLAM process more efficient, and it does not increase with even repeatedly scanning a same object. This is because extracted B-spline surfaces from different scans by scanning a similar object are merged as one, which lead to the small number of parameters. Comparing with planar surface-based 3D map construction, B-spline surface has a broad representation of complex environment. Not only the polyhedral objects can be expressed well, but also irregular and curved objects can be accurately expressed. If a polyhedral object is expressed with both of these two methods, the number of B-spline surface must be smaller than the number of planes because of its property of closures.

Even though B-spline surface is commonly used in computer aided design (CAD) [22], manufacturing, and reverse engineering [23], it is firstly considered to represent a large environment in the 3D map construction research in this paper. In order to extract the B-spline surfaces, two types of data sets are separated from the point cloud map according to different types of robot movements, continuous rotations, and continuous translations. An extended approximation algorithm is proposed to extract the raw data of each scan in every data set as a B-spline curve. The proposed method

generates a curve with as few control points as possible that approximates the raw data within a prespecified error tolerance by adding a new control point. We extract the initial B-spline curve with $p+1$ control points, where p is the degree of curve. A new knot point is inserted into the knot vector of the current curve at the position of a raw data point with the biggest divergence to the extracted curve. This iteration process is terminated until the error of the extracted curve is smaller than the error bound.

To extract the surface, control points of the curves are rearranged and considered as raw data points. Afterwards, the curve extraction process is repeated to find the control points of the B-spline surface. The covariance matrix of the control points is derived from the uncertainty of the raw data points. Due to the random robot movements, there must be overlap of two B-spline surfaces extracted from the different data sets. These overlaps should be merged as one to represent the environment. However, by far, the problem about merging two B-spline surfaces with overlaps is not well solved. This is because the distribution directions of the control points of these two overlapping surface patches are different. In our proposed method, each surface is first divided into the overlap and the nonoverlap. Then, one data set is generated from the overlap of one surface and the other data set is its projection locating on the other one. Merged data sets are obtained by using the product of two Gaussian probability density functions. Finally, a new B-spline surface is extracted from this merged data set to represent the environment instead of two overlaps.

The rest of this paper is organized as follows. The result of a 3D point cloud map construction is presented in Section 2. Definition, properties, fitting method, and covariance matrix derivation of B-spline surface are shown in Section 3. B-spline surface merging and the experiment results are discussed in Section 4 and Section 5, respectively. Finally, this paper is concluded in the last section.

2. 3D Point Cloud Map

To build the 2D and 3D map accurately, the position of a mobile robot should be accurately corrected after each movement. This can be realized by considering the extracted line segments as landmarks. Therefore, there are three subsections, line segments extraction, line segments-based 2D SLAM, and construction of 3D point cloud map.

2.1. Extraction of Line Segments. Landmarks play a key role in the update of robot pose. In this paper, line segments are considered as landmarks. These line segments are extracted from the segmented data groups of each sensor scan, obtained from a 2D LRF horizontally located on a mobile robot. The raw data of each sensor scan should be separated into many groups if the distance of two adjacent points is beyond a defined limit value. Moreover, the separated group is further divided if the angle of three sequential points is bigger than a limit angle. Then each data group is used to extract one line segment.

FIGURE 2: Extracted line segment (black line) from a group of raw sensor data (green points) has two geometrical parameters, expressed as intercept r and angle α in local coordinate system of mobile robot and R and ψ in absolute coordinate system. ρ_i and θ_i are the coordinates of a raw sensor data expressed in polar coordinate system.

For the extraction of line segments, each segment has two geometrical parameters, intercept r and angle α shown in Figure 2. In the figure, each raw sensor data is expressed as (ρ_i, θ_i) in polar coordinate system. The parameters of a line segment are expressed in a local coordinate system of mobile robot because the sensor scan is obtained in the local reference frame. To derive these parameters, the distance between the raw data points and the expected line segment is minimized. This distance is calculated by projecting the data point onto line segment, which is expressed as $d_i = \rho_i \cos(\theta_i - \alpha) - r$. The squared distance from the raw data point to the line are summed as follows:

$$S = \sum_{i=1}^N d_i^2 = \sum_{i=1}^N (\rho_i \cos(\theta_i - \alpha) - r)^2. \quad (1)$$

The least-square solution is found by setting the partial derivative of S with respect to r and α as zero, shown as follows:

$$\alpha = \frac{1}{2} \arctan \left[\frac{A}{B} \right], \quad (2)$$

$$A = \sum_{i=1}^N \rho_i^2 \sin(2\theta_i) - \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \rho_i \rho_j \sin(\theta_i + \theta_j), \quad (3)$$

$$B = \sum_{i=1}^N \rho_i^2 \cos(2\theta_i) - \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \rho_i \rho_j \cos(\theta_i + \theta_j),$$

$$r = \frac{\sum_{i=1}^N \rho_i \cos(\theta_i - \alpha)}{N}, \quad (4)$$

where A and B are symbols to represent the numerator and denominator in (2).

In order to match the new extracted line segments with the stored map features, these new segments should be

transformed in the global coordinate system. The parameters (R, ψ) of a line segment in the global coordinate system are shown in Figure 2, which are calculated based on the local parameters (r, α) and current robot position vector (x_r, y_r, φ_r) as in the following:

$$\begin{bmatrix} R \\ \psi \end{bmatrix} = \begin{bmatrix} r + \sqrt{x_r^2 + y_r^2} \cos \left(\varphi_r + \alpha - \arctan \left(\frac{y_r}{x_r} \right) \right) \\ \varphi_r + \alpha \end{bmatrix}. \quad (5)$$

2.2. IEKF-SLAM with PCA. In order to correctly localize the mobile robot and accurately build the 2D environment map, a data association method should be used to construct the correspondence between the stored line segments and the new extracted ones. Partial compatibility algorithm (PCA) [24] has been proposed as a new robust data association algorithm with a short computation time. The output of PCA is a vector storing best matching pairs. After this segments matching process by using PCA, each matching pair is used to update a state vector μ including the position of the mobile robot and the parameters of the line segments.

To consistently and efficiently update the state vector, an improved EKF (IEKF) SLAM algorithm was used. There are three parts in IEKF-SLAM algorithm, prediction, data association, and correction. In the n th step of prediction step, the inputs of this method at n th step are the state vector μ_{n-1} in (6) including the current pose (x_r, y_r, φ_r) of mobile robot and all the stored map $\{R_{\text{map}}^i\}$ segments with their covariance matrix $\{\Sigma_{\text{map}}^i\}$ at $(n-1)$ th step. Based on new odometry data $(\delta_{\text{trans}}^n, \delta_{\text{rot}}^n)$, the state vector μ_{n-1} with its covariance matrix Σ_{n-1} is predicted as $\bar{\mu}_n$ and $\bar{\Sigma}_n$ by using (7) to (9). After the prediction, new extracted segments and previously stored map features are matched by using PCA. A best matching vector can be obtained by considering the partial compatibility, which includes many unit matching pairs $[(R_{\text{new}}^k, \psi_{\text{new}}^k), (R_{\text{map}}^{j(k)}, \psi_{\text{map}}^{j(k)})]$. In correction step, each unit matching pair is used to update the state vector and its covariance matrix by using (12) to (15). To reduce the computation complexity, the large covariance matrix is decomposed as many block matrices. Finally, new updated state vector μ_n with its covariance matrix Σ_n is obtained, and unmatched new segments are stored as map segments. These whole steps iteratively process with the newly observed scans:

$$\mu_{n-1} = ((x_r, y_r, \varphi_r), (R_{\text{map}}^1, \psi_{\text{map}}^1), \dots, (R_{\text{map}}^N, \psi_{\text{map}}^N)). \quad (6)$$

(a) Prediction:

$$\bar{\mu}_n = \mu_{n-1} + F_n^T \underbrace{\begin{pmatrix} \delta_{\text{trans}}^n * \cos(\varphi_r^{n-1} + \delta_{\text{rot}}^n) \\ \delta_{\text{trans}}^n * \sin(\varphi_r^{n-1} + \delta_{\text{rot}}^n) \\ \delta_{\text{rot}}^n \end{pmatrix}}_{g(\mu_{n-1}, \mu_n)} \quad (7)$$

$$+ N(0, F_n^T R_n F_n),$$

$$F_n = (I_{3 \times 3}, 0_{3 \times 2}, \dots, 0_{3 \times 2}^N), \quad (8)$$

$$G_n = \frac{\partial g_{(1:3,1)}(\mu_{n-1}, u_n)}{\partial \mu_{n-1, (x_r, y_r, \varphi_r)}} = \begin{bmatrix} 1 & 0 & -\delta_{\text{trans}}^n * \sin(\varphi_r^{n-1} + \delta_{\text{rot}}^n) \\ 0 & 1 & \delta_{\text{trans}}^n * \cos(\varphi_r^{n-1} + \delta_{\text{rot}}^n) \\ 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

$$\bar{\Sigma}_n = \begin{pmatrix} G_n \Sigma_{n-1}^{3 \times 3} G_n^T + R_n & G_n \Sigma_{n-1}^{3 \times 2N} \\ (G_n \Sigma_{n-1}^{3 \times 2N})^T & \Sigma_{n-1}^{2N \times 2N} \end{pmatrix}. \quad (10)$$

(b) Data association:

$$\begin{pmatrix} \underbrace{(R_{\text{new}}^k, \psi_{\text{new}}^k)^T}_{z_n^k}, \underbrace{(R_{\text{map}}^{j(k)}, \psi_{\text{map}}^{j(k)})^T}_{z_n^{j(k)}} \\ \vdots \end{pmatrix} \leftarrow \text{PCA}(\mu_{n-1}, z_n). \quad (11)$$

(c) Correction:

$$K_n^k = \bar{\Sigma}_n [H_n^k]^T \psi_k^{-1},$$

$$H_n^k = ({}^0H_n^k, 0_{2 \times 2}, \dots, 0_{2 \times 2}, {}^iH_n^k, 0_{2 \times 2}, \dots, 0_{2 \times 2}),$$

$${}^0H_n^k = \begin{bmatrix} x_r c_\Phi - y_r s_\Phi & y_r c_\Phi + x_r s_\Phi & -L s_\Phi \\ L & 0 & 1 \end{bmatrix}, \quad (12)$$

$${}^iH_n^k = \begin{bmatrix} 1 & -L \sin \Phi \\ 0 & 1 \end{bmatrix},$$

where $L = \sqrt{x_r^2 + y_r^2}$, $\Phi = \varphi_r + \alpha = \text{atan}(x_r/y_r)$, $s_\Phi = \sin \Phi$, $c_\Phi = \cos \Phi$,

$$\bar{\Sigma}_n [H_n^k]^T = A_0 {}^1H_n^k + A_i {}^iH_n^k, \quad (13)$$

$$\bar{\Sigma}_n = (A_0, \dots, A_i, \dots, A_N),$$

where $A_0 = (2N+3) \times 3$, $A_{i:1 \rightarrow N} : (2N+3) \times 2$,

$$\psi_{j(k)} = {}^0H_n^k A_0^0 ({}^0H_n^k)^T + {}^iH_n^k A_i^i ({}^0H_n^k)^T + {}^0H_n^k A_i^0 ({}^iH_n^k)^T$$

$$+ {}^iH_n^k A_i^i ({}^iH_n^k)^T + C_{\text{Line}}^j,$$

$$A_i = (A_i^0 \ \dots \ A_i^j \ \dots \ A_i^N), \quad (14)$$

where $A_{i:0}^{j:0} = 3 \times 3$, $A_{i:0}^{j:1 \rightarrow N} = 2 \times 3$, $A_{i:1 \rightarrow N}^{j:1 \rightarrow N} : 2 \times 2$,

$$\mu_n = \bar{\mu}_n + K_n^k (z_n^k - z_n^{j(k)}), \quad (15)$$

$$\Sigma_n = \bar{\Sigma}_n - \bar{\Sigma}_n [H_n^{j(k)}]^T \psi_{j(k)}^{-1} (\bar{\Sigma}_n [H_n^{j(k)}]^T)^T.$$

2.3. Construction of a 3D Point Cloud Map. The experimental system established in our research is shown in Figure 3, in which the horizontal sensor obtains 760 raw data points with the measurement range of 270 degrees, and vertical sensor obtains 541 points with the same measurement range. Two laser sensors are mutually perpendicular with each other, and the horizontal sensor is parallel with ground. In the first subsection, the line segments are extracted from the raw sensor data, which are obtained from the horizontal LRF. By matching and updating these line segments, the pose of a mobile robot can be corrected accurately. Based on the updated robot position, the raw data points obtained from the vertical sensor are projected into the 3D space to build the 3D point cloud map.

Four reference frames are built in Figure 3 to express a laser point into 3D space. These are the absolute reference frame $\{O\text{-}XYZ\}$, reference frame of mobile robot $\{O_R\text{-}X_R Y_R Z_R\}$, reference frame of horizontal LRF $\{O_H\text{-}X_H Y_H Z_H\}$, and reference frame of vertical LRF $\{O_V\text{-}X_V Y_V Z_V\}$. Assume that the j th laser point obtained from the vertical LRF in a global reference frame is represented as ${}^{O\text{-}XYZ}P_j$, which can be transformed as follows:

$$\begin{bmatrix} {}^{O\text{-}XYZ}P_j \\ 1 \end{bmatrix} = {}_{O_R\text{-}X_R Y_R Z_R}^{O\text{-}XYZ}T {}_{O_V\text{-}X_V Y_V Z_V}^{O_R\text{-}X_R Y_R Z_R}T \begin{bmatrix} {}^{O_V\text{-}X_V Y_V Z_V}P_j \\ 1 \end{bmatrix}, \quad (16)$$

where

$${}_{O_R\text{-}X_R Y_R Z_R}^{O\text{-}XYZ}T = \begin{bmatrix} \cos \varphi_r & -\sin \varphi_r & 0 & x_r \\ \sin \varphi_r & \cos \varphi_r & 0 & y_r \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}_{O_V\text{-}X_V Y_V Z_V}^{O_R\text{-}X_R Y_R Z_R}T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & {}^R x_V \\ 1 & 0 & 0 & {}^R y_V \\ 0 & 0 & 1 & {}^R z_V \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (17)$$

$${}_{O_V\text{-}X_V Y_V Z_V}P_j = \begin{bmatrix} {}^V \rho_j \cos {}^V \theta_j \\ {}^V \rho_j \sin {}^V \theta_j \\ 0 \end{bmatrix}.$$

Finally, we get

$${}^{O\text{-}XYZ}P_j = \begin{bmatrix} -\sin \varphi_r {}^V \rho_j \cos {}^V \theta_j + \cos \varphi_r {}^R z_V - \sin \varphi_r {}^R y_V + x_r \\ \cos \varphi_r {}^V \rho_j \cos {}^V \theta_j + \sin \varphi_r {}^R z_V + \cos \varphi_r {}^R y_V + y_r \\ {}^V \rho_j \sin {}^V \theta_j - {}^R x_V \end{bmatrix}. \quad (18)$$

By using (16) to (18), each 2D scan obtained from the vertical LRF can be correctly expressed in the global reference

FIGURE 3: Experimental setup, a Pioneer mobile robot, a vertical SICK LMS 100 LRF, and a horizontal Hokuyo 08LX LRF.

frame. To have a robust 3D point cloud map, the mobile robot in an indoor environment is controlled to translate with a small interval or rotate with a small angle. In this paper, the spacing distance is 100 mm and the rotation angle is 0.087 rad in each time.

3. B-Spline Surface

In this section, some fundamental concepts and the fitting method of the B-spline surface are presented. The term *spline* is a sufficiently smooth polynomial function that is piecewise-defined, which can be applied in many scientific applications where approximation or interpolation of noisy data is required. A $p \times q$ th degree B-spline surface is obtained as the tensor product of a p th degree (order $p + 1$) B-spline curve and a q th degree B-spline curve.

3.1. Definition and Properties of the B-Spline Surface. A B-spline surface of the order $p \times q$ with a bidirectional net of control points $P_{i,j}$ ($i = 0, \dots, n$; $j = 0, \dots, m$), two knot vectors U, V , and the product of the univariate B-spline basis functions $N_{i,p}(u)N_{j,q}(v)$ can be expressed as

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v)}. \quad (19)$$

The basis functions of $N_{i,t}(u)$ ($t = 1, \dots, p$) are defined as

$$N_{i,1}(u) = \begin{cases} 1, & \text{if } u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

and for all $t > 1$

$$N_{i,p} = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1} + \frac{u_{i+p+1} - u_i}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}, \quad (21)$$

where $N_{i,p}$ is the simplified form of $N_{i,p}(u)$. $N_{j,q}(v)$ can be calculated using the same method. Two parameters u and v are defined as $u \in [a_1, b_1]$, $v \in [a_2, b_2]$. The two nondecreasing knot vectors are expressed as follows:

$$\begin{aligned} U : \underbrace{a_1 = \dots = a_1}_{p+1} \leq u_{p+1} \leq \dots \leq u_{r-p-1} \leq \underbrace{b_1 = \dots = b_1}_{p+1}, \\ V : \underbrace{a_2 = \dots = a_2}_{q+1} \leq v_{q+1} \leq \dots \leq v_{s-q-1} \leq \underbrace{b_2 = \dots = b_2}_{q+1}. \end{aligned} \quad (22)$$

The number of knot points in these two knot vectors is $n + p + 1$ and $m + q + 1$, respectively. An example of these parameters is illustrated in Figure 4.

There are many mathematical and geometrical properties of a B-spline surface, which are very useful for the remaining content. Four of them are listed as follows.

- (1) The minimum number of control points of a $p \times q$ th B-spline surface is $(p + 1)(q + 1)$.
- (2) For any value of the parameters $(u, v) \in [a_1, b_1] \times [a_2, b_2]$, the sum of its all the B-spline basis function is 1; that is,

$$\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) = 1. \quad (23)$$

- (3) In any given rectangle, at most $(p + 1)(q + 1)$ basis functions are nonzero if $(u, v) \in [u_s, u_{s+1}] \times [v_t, v_{t+1}]$. This means that the value of $S(u, v)$ depends only on $(p + 1)(q + 1)$ of the coefficients

$$S(u, v) = \sum_{i=s-p}^s \sum_{j=t-q}^t N_{i,p}(u) N_{j,q}(v) P_{i,j}. \quad (24)$$

FIGURE 4: Example of a bicubic B-spline surface ($p = q = 3$) with control points and the basis functions with two knot vectors (U, V).

- (4) Let (u, v) be fixed, the partial derivative of $S(u, v)$ can be obtained by computing the derivatives of the basis functions:

$$\frac{\partial^{(k+l)} S(u, v)}{\partial^{(k)} u \partial^{(l)} v} = \sum_{i=0}^{n-k} \sum_{j=0}^{m-l} N_{i,p-k}^{(k)} N_{j,q-l}^{(l)} P_{i,j}, \quad (25)$$

$$N_{i,p-k}^{(k)} = \frac{(p-k) N_{i,p-k-1}^{(k-1)}}{u_{i+p-k} - u_i} - \frac{(p-k) N_{i+1,p-k-1}^{(k-1)}}{u_{i+p-k+1} - u_{i+1}}. \quad (26)$$

More information about the properties of the B-spline surface can be seen in [25].

3.2. Fitting of the B-Spline Surface. Assume that there is a data set $Q_{k,l}$ ($k = 0, \dots, M; l = 0, \dots, N$) with $N + 1$ sensor scans, and each includes $M + 1$ raw data points. The least square approximation method is used to find the control points of the B-spline surface by minimizing the error between the raw sensor data and the extracted surface, which is expressed as

$$\sum_{l=0}^N \sum_{k=0}^M |Q_{k,l} - S(\bar{u}_k, \bar{v}_l)|^2, \quad (27)$$

where the parameter set $\{\bar{u}_k\}$ in the l th data group is calculated by using the centripetal method [25]. Let

$$d_l = \sum_{k=0}^M \sqrt{Q_{k,l} - Q_{k-1,l}}. \quad (28)$$

Then

$$\bar{u}_0 = a_1, \quad \bar{u}_M = b_1, \quad \bar{u}_k = \bar{u}_{k-1} + \frac{\sqrt{|Q_{k,l} - Q_{k-1,l}|}}{d_l}. \quad (29)$$

The first step in proposed B-spline surface extraction algorithm is the B-spline curve extraction from each sensor

scan. This means that the error between the raw data in each scan and the curve is minimized, which is done by fixing the parameter l in (27). In the beginning, $p + 1$ control points are extracted from the raw data with $2p + 2$ knot points, $p + 1$ zeroes, and $p + 1$ ones. The control points T_l of the curve are calculated by setting the partial derivative of the sum of the error with respect to all the control points as zero:

$$T_l = (N_u^T N_u)^{-1} N_u^T Q_l, \quad N_{\bar{u}} = \begin{pmatrix} N_{0,p}(\bar{u}_0) & \cdots & N_{n,p}(\bar{u}_0) \\ \vdots & \ddots & \vdots \\ N_{0,p}(\bar{u}_M) & \cdots & N_{n,p}(\bar{u}_M) \end{pmatrix}, \quad (30)$$

$$T_l = [T_{0,l}, \dots, T_{m,l}]^T, \quad Q_l = [Q_{0,l}, \dots, Q_{M,l}]^T.$$

If the error between the raw data point and this extracted curve is bigger than the limit value, a new knot is added in the common knot vector of these curves at the knot position of a sensor point which has the maximum value of all the biggest divergence between the sensor data to the corresponding extracted curve in each scan. This process is terminated until all the error of the curves is located within the error bound.

To calculate the control points of the B-spline surface, the control points of the curves in the first step is rearranged and considered as raw data:

$$T = \begin{bmatrix} T_{0,0} & \cdots & T_{m,0} \\ \vdots & \ddots & \vdots \\ T_{0,N} & \cdots & T_{m,N} \end{bmatrix}. \quad (31)$$

New control points from these data can be obtained using the same method as in (25):

$$P_{i,:} = (N_v^T N_v)^{-1} N_v^T T_{i,:}, \quad (32)$$

where $P_{i,:}$ is the i th column of the control points of the extracted B-spline surface with the dimension of $3n \times 1$ in

FIGURE 5: Two types of extracted B-spline surfaces according to two different types of robot movements, continuous translation and continuous rotation (different scans are plotted with different shapes of points).

three-dimensional space. Matrix $N_{\bar{v}}$ and parameters $\{\bar{v}_k\}$ are calculated using the same principle as in (24) and (26). To express the uncertainty of extracted surface, the covariance matrix C_P of the control points is propagated from the covariance matrix C_Q of the raw data points using first-order Taylor expansion:

$$\begin{aligned} C_{T_i} &= \left[(N_{\bar{u}}^T N_{\bar{u}})^{-1} N_{\bar{u}}^T \right] C_{Q_i} \left[(N_{\bar{u}}^T N_{\bar{u}})^{-1} N_{\bar{u}}^T \right]^T, \\ C_{P_{i,j}} &= \left[(N_{\bar{v}}^T N_{\bar{v}})^{-1} N_{\bar{v}}^T \right] C_{T_{i,j}} \left[(N_{\bar{v}}^T N_{\bar{v}})^{-1} N_{\bar{v}}^T \right]^T. \end{aligned} \quad (33)$$

In our research, the raw data from the constructed 3D point cloud map are divided into two types according to the different types of robot movements, continuous rotation and continuous translation. All the combined movements of rotation and translation can be analyzed by dividing these movements into the two types of defined movements. An example of the extracted B-spline surface from the two simulated types of raw data is shown in Figure 5. In addition, due to the random robot movements, there may be overlap between the two different extracted B-spline surfaces. The overlap of the two surfaces should be merged as one to represent the environment, which is done in the following section.

FIGURE 6: An example of two B-spline surfaces with overlapping parts.

4. Merging of the B-Spline Surfaces

An example of the two surfaces with overlap is illustrated in Figure 6, in which the distribution directions of the control points of the two surfaces are different and intersected. To represent the environment without overlapping surfaces, two steps are executed, which are the B-spline surface division and merging of the two overlapping patches of the two surfaces.

4.1. Division of B-Spline Surface. To merge the overlaps of the two surfaces, the overlapping part of each surface should first be found. Each surface is separated into two parts, the

overlap and the nonoverlap. This is done by projecting the boundary points of one B-spline surface onto that of the other one. Projection of a point Q^P onto a B-spline surface S is an iteration process, which begins with the closest generated sample point Q_{init}^P (with knot points $(u_{\text{init}}, v_{\text{init}})$) of the surface S to the point Q^P . Because of the finite number of sample points, the projection point of Q^P cannot be exactly the same as a sample point. The knot points (u_{i+1}, v_{i+1}) of the much closer points located on the surface S are calculated as follows:

$$J_i \begin{bmatrix} u_{i+1} - u_i \\ v_{i+1} - v_i \end{bmatrix} = - \begin{bmatrix} r(u_i, v_i) \cdot S_u(u_i, v_i) \\ r(u_i, v_i) \cdot S_v(u_i, v_i) \end{bmatrix},$$

$$r(u_i, v_i) = S(u_i, v_i) - Q^P, \quad (34)$$

$$J_i = \begin{bmatrix} |S_u|^2 + rS_{uu} & S_u S_v + rS_{uv} \\ S_u S_v + rS_{uv} & |S_v|^2 + rS_{vv} \end{bmatrix},$$

where $S_u, S_v, S_{uu}, S_{vv}, S_{uv}$ are the derivative of surface S with respect to u and v at the knot position u_i, v_i , calculated by using (20). This iteration is terminated when at least one of the following two conditions is satisfied:

$$\frac{|S_*(u_i, v_i) r(u_i, v_i)|}{|S_*(u_i, v_i)| |r(u_i, v_i)|} \leq \varepsilon_1; \quad (* : u, v), \quad (35)$$

$$|(u_{i+1} - u_i) S_u + (v_{i+1} - v_i) S_v| \leq \varepsilon_2.$$

By repeating the projection process of the generated sample points, the boundaries of the overlapping part and nonoverlapping part in each B-spline surface of Figure 6 can be found. The boundary points of these two surfaces are shown in the top part of Figure 7, in which the nonoverlap part is separated into two parts because the distribution of the knot points is discontinued in some places. Separation of the nonoverlapping part is done by fixing one of the two knot points of the corner point locating on the boundary. According to these boundaries, the sample points are generated and assembled into different groups. To maintain the boundary of the original surface, an interpolation algorithm is used to calculate the control points of a new surface after obtaining the control points of the curves in each group of the sample data. The segmented B-spline surface patches are shown in the bottom part of Figure 7, where the number of control points increased due to the maintenance of the shape.

4.2. Merging of the Overlapping B-Spline Surfaces. To merge the overlapping surface patches, two surface patches should be selected correctly from all the patches. An example with six B-spline surface patches is shown in Figure 7. We find the goal patches by projecting the middle point with the knot points $(u_{\text{mid}}, v_{\text{mid}})$ of the sequential patches from one original surface to that from the other one. Here $(u_{\text{mid}}, v_{\text{mid}})$ are set as $(0.5 * (a_1^i + b_1^i), 0.5 * (a_2^i + b_2^i))$, where $a_1^i, b_1^i, a_2^i, b_2^i$ are the start knot point and end knot point of the i th B-spline surface patch in the U and V direction, respectively. This procedure is terminated if the projection point of the middle point is not located on the edge of the objective patch. These two surface

FIGURE 7: Boundary points of the divided B-spline surface patches (column 1), the sample points of these patches (column 2), and the corresponding extracted B-spline surface patches (column 3). The boundary points and the sample points are plotted using different colors and different shape of point for each patch.

patches cannot be merged by updating the control points directly because not only is the number of control points from the two surface patches different but also the distribution directions of the control points of the two are different.

Two surface patches are merged by operating the generated sample points. Due to the different distribution directions of the sample points of the two surfaces, it is difficult to group the combined sample points of the two B-spline surface patches. To solve this problem, one group of sample points is generated from one B-spline surface patch. By projecting these sample points onto the objective patch, these projection points located on this surface patch are considered as the second group. The covariance matrix of these two groups of sample points is propagated from the covariance matrix of the two B-spline surface patches, respectively. The covariance matrix Σ_S of the sample point S is propagated from the covariance matrix C_P of the control points as follows:

$$\Sigma_S = \frac{N}{3 \times 3} \frac{C_P}{3 \times K} \frac{N^T}{K \times 3} \quad (36)$$

with

$$P = (P_{0,0}, \dots, P_{m,0}, P_{1,0}, \dots, P_{m,n}),$$

$$N = (N_{0,p} N_{0,q} I_{3 \times 3}, \dots, N_{m,p} N_{n,q} I_{3 \times 3}), \quad (37)$$

$$K = 3(m+1)(n+1).$$

The merging process of the overlapping parts in the example of Figure 7 is shown in Figure 8, in which the two groups of sample points are from two different patches. Each sample point and its projection point are merged by using

FIGURE 8: Merged surface (right) of two overlapping patches by merging the sample points (left star points) from one and their projection points (middle circular points) from the other one.

the product of two Gaussian PDFs. Assume that there is a state vector μ_s of a sample point with its covariance matrix Σ_s and the state vector μ_p of the projection point with a covariance matrix Σ_p ; the state vector of the combined point with its covariance matrix are calculated as follows

$$\begin{aligned}\mu &= (\Sigma_s^{-1} + \Sigma_p^{-1})^{-1} (\Sigma_s^{-1} \mu_s + \Sigma_p^{-1} \mu_p), \\ \Sigma &= (\Sigma_s^{-1} + \Sigma_p^{-1})^{-1}.\end{aligned}\quad (38)$$

Finally, the merged surface is extracted from all the merged points of the sample points and their projection points using previously described surface extraction method. All the B-spline surface patches of the example in Figure 6 are shown in Figure 9.

FIGURE 9: B-spline surface patches after surface division and merging of overlapping patches.

5. Experiment Results

An experiment was performed with real data obtained using the experimental tools in Figure 3 in order to validate the methodologies presented in this paper. To represent an environment with continuous B-spline surface patches, a 3D indoor experimental environment was selected without any meshes and holes. Initially, a 3D environment map is represented with the point clouds. Then the accuracy and effectiveness of an extracted B-spline surface from the point clouds was analyzed. To show the process of map construction, the surface division and merging are explained by using an example with real sensor data. Finally, the entire environment map represented with B-spline surfaces is shown.

5.1. 2D Map and 3D Point Cloud Map. A two-dimensional map of the real environment is built as shown in Figure 10, where the robot position is corrected by matching the new measured line segments with stored segments. To show the detailed matching process, the number of new measurements and stored ones with each robot movement are illustrated in Figure 11. It can be seen that the number of new extracted segments from each scan is not bigger than 14, and the number of the stored segments generally increases. However, it decreases in some steps, especially in the beginning steps. This is because some discontinuous line segments actually

belonging to the same line are extracted initially. With the accumulated environment information, the remaining parts of the line are observed and merged with the stored segments to represent the environment map.

By using the information from the horizontal LRF, the robot position is corrected, and the 2D map is simultaneously constructed. Based on the accurate robot pose, the observed information from the vertical LRF is transformed into the 3D coordinate system to build the 3D point cloud map, shown in Figures 1 and 12. It can be seen that the discrete points are arranged regularly. In each vertical sensor scan, the cross section of the environment at the current robot pose is scanned except for part of the ground due to the limited scan range of a sensor. The measuring distances of all the laser points in every vertical scan are continuous without large fluctuations.

5.2. Accuracy and Efficiency of Extracted B-Spline Surface. As mentioned in Section 4, the entire data set of the 3D point cloud map is divided into two types according to the continuous translation and continuous rotation of a mobile robot. The 3D point cloud map in Figure 12 is separated into seven data sets with these two different types of robot movements. The raw sensor data in each data set is used to be extracted as one B-spline surface patch. The detailed process of the bicubic B-spline surface extraction from the 7th data set with continuous robot translation is shown in

FIGURE 10: Real experiment environment (a), corrected position of mobile robot, and the constructed 2D map (b).

FIGURE 11: Number of new extracted line segments and stored line segments in each step.

Figure 13. There are two types of control points, control points of the extracted B-spline curves from the raw sensor data in U direction, and the control points of extracted B-spline surface (control points of the B-spline curves in V direction) from the control points in U direction. To show the propagation of the uncertainties, the uncertainty ellipsoids of the raw sensor data and these two types of control points are plotted with different scales according to their corresponding covariance matrix. The covariance matrices of these control points are calculated in (33).

In addition, the B-spline surfaces are extracted from the same data set with different degrees to show the accuracy and efficiency of the bicubic B-spline surface. The extracted B-spline surfaces with $(p = 0, q = 0)$, $(p = 0, q = 3)$, and $(p = 3, q = 3)$ are shown in Figure 14. It can be seen that the B-spline surface with $q = 3$ in the middle and right of the figure is much smoother than the surface with $q = 0$ in the left.

FIGURE 12: Another view of constructed 3D point cloud map.

Furthermore, the error between all these raw data points in this data set and these three B-spline surfaces are calculated and shown in Figure 15. The limit values of the error in these three surface extraction processes are set as the same in the U direction and V direction, which are 0.01 m and 0.05 m, respectively. In this data set, there are twenty 2D scans from the vertical LRF. The error between all the 541 sensor points of one 2D scan and the surface is plotted with a continuous curve. Therefore, in all these three figures of Figure 15, there are twenty curves.

By comparing the error range of these three extracted surfaces, the B-spline surface with $(p = 0, q = 0)$ has the largest error range. In four 2D scans, in particular, the average error between the raw data and this surface is about 0.05 m. The errors of part of the scans at the 350th step to the 480th step are bigger than 0.05 m. For the surface with $(p = 0, q = 3)$, the error range at all the 541 steps is about 0 to 0.04 m. Even though the error range of this surface is more stable than the surface with $(p = 0, q = 0)$, this error range is bigger than that of the surface with $(p = 3, q = 3)$. The error range at most of the steps of this bicubic B-spline surface is about 0 to 0.02 m. However, the error range at the 60th step, 260th step, 350th step, and 470th step is a little bigger. This is because there are four sharp corners at these steps in a real experiment environment. Bigger errors exist between the raw sensor data scanned from the sharp corners and the extracted B-spline surface because of its property of continuity. This can be proven by the uncertainty propagation in Figure 13 (column 6) where the size of the uncertainty ellipsoid of the control points located in the sharp corners is bigger than other parts. However, even though the maximum error in the corners' part is about 0.06 m, which is relatively small with respect to the measuring error of LRF, ± 0.03 m.

To know the accuracy of the extracted B-spline surface, the average errors between all the sensor data in this data set and three surfaces with different degrees are calculated. These average errors are plotted with red dashed lines in Figure 15, which are 0.0325 m, 0.0192 m, and 0.0131 m for the surface with $(p = 0, q = 0)$, $(p = 0, q = 3)$, and $(p = 3, q = 3)$, respectively. It can be seen that the bicubic B-spline surface has the minimum error in all these three extracted surfaces. Furthermore, the number of control points in the U and V direction of these three surfaces is compared in

FIGURE 13: Example of raw sensor data (column 1) and their uncertainty ellipsoid (column 2) with the continuous translation of mobile robot. Extracted B-spline curves ($p = 3$) in U direction with the control points (column 3) and the ellipsoid of propagated uncertainty from the raw sensor data (column 4). Extracted B-spline curves ($q = 3$) in V direction with the control points (column 5) and the ellipsoid of propagated from the control points in U direction (column 6). These three types of uncertainty ellipsoid are plotted with the ratio of 50:1, 100:1, and 100:1 with respect to their corresponding real sizes.

FIGURE 14: Comparison of extracted B-spline surface with different degrees: ($p = 0, q = 0$) (a), ($p = 0, q = 3$) (b), and ($p = 3, q = 3$) (c).

Figure 16, which shows that the bicubic B-spline surface has the smallest number of control points. Less control points means that less iteration is used in the process of surface extraction. Moreover, the storage space is also saved by representing the surface extracted 541×20 raw data points with only 26×5 control points. In summary, the bicubic B-spline surface can accurately represent the real environment with a fewer number of control points.

5.3. Division and Merging of B-Spline Surface. Due to the seven data sets separated from the constructed 3D point cloud map, there must be seven extracted B-spline surfaces. To represent the environment using B-spline surface patches without overlap between any two patches, the overlapping parts should be merged as one. In Section 4, the detailed

process of B-spline surface division and merging is demonstrated with simulations. In this subsection, the second data set obtained with continuous robot rotation and the fifth data set obtained with continuous translation are selected from the constructed 3D point cloud map to show the same process but with real sensor data. The raw sensor data of these two data sets are shown in the first row of Figure 17, in which the overlapping parts can be clearly seen from the top view in the right of this row. By using the same principle of Figure 5, the extracted B-spline surfaces are shown in the second row of Figure 17. It can be seen that the concave part of the ceiling in the environment is well represented with B-spline surfaces extracted from both of these two data sets.

To merge the overlapping parts of these two B-spline surfaces, one of them is divided into five patches, P_{51} , P_{52} , P_{53} , and P_{54} , and the other one is divided into four patches,

FIGURE 15: Error between the raw sensor data in each 2D scan from the vertical LRF and the extracted B-spline surfaces with different degrees: ($p = 0, q = 0$) (a), ($p = 0, q = 3$) (b), and ($p = 3, q = 3$) (c). The average errors between all the sensor data and these extracted B-spline surfaces are plotted with red dashed line.

FIGURE 16: Number of control points in U and V direction of the extracted B-spline surfaces with different degrees: ($p = 0, q = 0$) (left), ($p = 0, q = 3$) (middle), and ($p = 3, q = 3$) (right).

P_{21} , P_{22} , P_{23} , P_{24} , and P_{25} , as shown in Figure 18. The detailed process of the B-spline surface division is presented with simulations in Figure 7. Here only the division result is shown in the figure, where the surface patches P_{52} and P_{22} represent the same part of the real environment but are extracted from two different data sets. Similar to these two patches, P_{54} and P_{24} also represent the same part of the ground. Due to the different distribution direction of the control points of these two surface patches in each pair, these two patches cannot be directly merged by operating the control points. To merge these two patches in each pair, one group of sample points is generated from one patch, and they are projected onto the other one. Because the sample points and their projection points have the same distribution direction, each sample point and its projection point are combined by using the product of two Gaussian PDFs in (38). Finally, the merged B-spline surface patches and other nonoverlapping surface patches are assembled as in the right-bottom of Figure 18.

5.4. Whole Constructed Environment Map. The entire constructed environment map from the 3D point cloud map is represented with bicubic B-spline surface patches shown in Figure 19, in which the control points and mesh grids are not plotted for simplicity and clarity. Some walls are expressed with many small surface patches because of the surface division for the merging of the overlapping parts. In the figure, there are only tiny cracks between two adjacent B-spline surface patches. It can also be seen that the corners where the curvature changes sharply in the environment are expressed correctly. The concave rectangle located in the ceiling is clearly represented after many robot movements. In addition, the walls and ground are smoothly constructed without any bumpiness.

6. Conclusions

A novel methodology for 3D map representation of the environment with the characteristics of complicated geometry has been proposed and experimentally validated. In consideration of the limitation of a traditional 3D point cloud map, the B-spline surface has superior advantages of the expression of complex structure. To represent a map with B-spline surfaces, the 3D point cloud map was constructed initially by using two mutually perpendicular LRFs. To build this point cloud map, an accurate robot position is corrected based on the extracted line segments from the horizontal LRF. The IEKF SLAM algorithm was used to update the robot position with the features pairs of the line segments and their matched stored segments. Raw sensor data obtained from the vertical LRF forms the point cloud map of the 3D environment map.

To extract the B-spline surface from the point cloud map, two types of data sets were segmented from the point cloud map according to two different types of robot movements, continuous translation and continuous rotation. Due to

FIGURE 17: Raw sensor data of the data set with the robot movements of continuous rotation (red points in row 1, column 1) and raw sensor data of the data set with the robot movements of continuous translation (blue points in row 1, column 1). Overlapping parts of these two data sets also can be seen in the top view (row 1, column 2). Extracted B-spline surfaces from these two data sets are shown in row 2, column 1, and row 2, column 2, respectively.

FIGURE 18: Divided B-spline surface patches (P_{21} , P_{22} , P_{23} , P_{24} , P_{25} and P_{51} , P_{52} , P_{53} , P_{54}) from the two data sets, and assembly of all these B-spline surface patches after merging the overlapping patches.

the existing overlap between the two B-spline surfaces, a B-spline surface division method was proposed to divide the surface into two types, overlaps and nonoverlaps. Then a merging method was presented to merge the overlapping surface patches with different distribution directions of their control points by operating the generated sample points and

FIGURE 19: Front view (a) and back view (b) of the whole 3D environment map expressed using bicubic B-spline surface patches.

their projection points. Simulations of the two B-spline surfaces with overlaps were used to show this detailed process. Finally, a real experimental environment was successfully constructed with B-spline surface patches, which validated the accuracy, efficiency, and feasibility of proposed methods.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the Ministry of Trade, Industry and Energy (MOTIE), KOREA, through the Education Support Program for Creative and Industrial Convergence. (Grant no. N0000717).

References

- [1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.
- [2] S.-Y. Hwang and J.-B. Song, "Monocular vision-based global localization using position and orientation of ceiling features," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 3785–3790, IEEE, Karlsruhe, Germany, May 2013.
- [3] R. J. Yan, J. Wu, M. L. Shao, K. S. Shin, J. Y. Lee, and C. S. Han, "Mutually converted arc-line segment-based SLAM with summing parameters," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2014.
- [4] L. M. Paz, J. D. Tardós, and J. Neira, "Divide and conquer: EKF SLAM in $O(n)$," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1107–1120, 2008.
- [5] W. S. Wijesoma, L. D. L. Perera, and M. D. Adams, "Toward multidimensional assignment data association in robot localization and mapping," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 350–365, 2006.
- [6] J.-L. Blanco, J. González-Jiménez, and J.-A. Fernández-Madrigal, "An alternative to the Mahalanobis distance for determining optimal correspondences in data association," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 980–986, 2012.
- [7] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1611–1626, 2013.
- [8] J. Folkesson and H. I. Christensen, "Closing the loop with graphical SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 731–741, 2007.
- [9] K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos, "Learning to close loops from range data," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1728–1754, 2011.
- [10] R. Valencia, E. H. Teniente, E. Trulls, and J. Andrade-Cetto, "3D mapping for urban service robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, pp. 3076–3081, IEEE, St. Louis, Mo, USA, October 2009.
- [11] K. Granström and T. B. Schön, "Learning to close the loop from 3D point clouds," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 2089–2095, October 2010.
- [12] M. Liu and R. Siegwart, "Navigation on point-cloud—a Riemannian metric approach," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '14)*, pp. 4088–4093, IEEE, Hong Kong, May 2014.
- [13] F. Colas, S. Mahesh, F. Pomerleau, M. Liu, and R. Siegwart, "3D path planning and execution for search and rescue ground robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 722–727, Tokyo, Japan, November 2013.
- [14] J. Welle, D. Schulz, T. Bachran, and A. B. Cremers, "Optimization techniques for laser-based 3D particle filter SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 3525–3530, May 2010.
- [15] J. Weingarten and R. Siegwart, "EKF-based 3D SLAM for structured environment reconstruction," in *IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 3834–3839, August 2005.
- [16] M. Bosse, R. Zlot, and P. Flick, "Zebedee: design of a spring-mounted 3-D range sensor with application to mobile mapping," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1104–1119, 2012.
- [17] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [18] P. Pfaff, R. Triebel, and W. Burgard, "An efficient extension to elevation maps for outdoor terrain mapping and loop closing," *International Journal of Robotics Research*, vol. 26, no. 2, pp. 217–230, 2007.
- [19] X. Zhu, C. Qiu, and M. A. Minor, "Terrain inclination aided three-dimensional localization and mapping for an outdoor mobile robot," *International Journal of Advanced Robotic Systems*, vol. 10, article 76, 2013.
- [20] A. J. B. Trevor, J. G. Rogers III, and H. I. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 3041–3048, IEEE, Saint Paul, Minn, USA, May 2012.
- [21] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 5182–5189, May 2013.
- [22] S. Xu, W. Jahn, and J.-D. Müller, "CAD-based shape optimisation with CFD using a discrete adjoint," *International Journal for Numerical Methods in Fluids*, vol. 74, no. 3, pp. 153–168, 2014.
- [23] R. Bénére, G. Subsol, G. Gesquière, F. Le Breton, and W. Puech, "A comprehensive process of reverse engineering from 3D meshes to CAD models," *Computer Aided Design*, vol. 45, no. 11, pp. 1382–1393, 2013.
- [24] R. J. Yan, J. Wu, Q. Yuan et al., "Natural corners-based SLAM with partial compatibility algorithm," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 228, no. 8, pp. 591–611, 2014.
- [25] L. Piegl and W. Tiller, *The NURBS Book*, Springer, New York, NY, USA, 2nd edition, 1997.