

The Efficient Extension of Globally Consistent Scan Matching to 6 DoF

Dorit Borrmann Jan Elseberg Kai Lingemann Andreas Nüchter Joachim Hertzberg
University of Osnabrück
Knowledge-Based Systems Research Group
Albrechtstraße 28 D-49069 Osnabrück, Germany

Abstract

Over ten years ago, Lu and Milios presented a probabilistic scan matching algorithm for solving the simultaneous localization and mapping (SLAM) problem with 2D laser range scans, a standard in robotics. This paper presents an extension to this GraphSLAM method. Our iterative algorithm uses a sparse network to represent the relations between several overlapping 3D scans, computes in every step the 6 degrees of freedom (DoF) transformation in closed form and exploits efficient data association with cached k - d trees. Our approach leads to globally consistent 3D maps, precise 6D pose and covariance estimates, as demonstrated by various experimental results.

1 Introduction

Complex 3D digitalization without occlusions requires multiple 3D scans. Globally consistent scan matching is the problem of aligning the poses of n partially overlapping 3D scans such that the resulting model does not show any inconsistencies. Incremental methods like ICP [3] lead to inconsistencies, due to limited precision of each matching and accumulation of registration errors. Examples are *pairwise matching*, i.e., each new scan is registered against the scan with the largest overlapping area, and *metascan matching*, i.e., the new scan is registered against a so-called *metascan* [5], which is the union of the previously acquired and registered scans. However, for a point cloud consisting of several scans, a new scan might contain information that improves previous registrations. Globally consistent scan matching methods regard this fact: They transform all scans by minimizing an error function that takes all 3D scans into account.

Lu and Milios [11] proposed a probabilistic approach that builds a network of laser scans and their corresponding pose differences to form a linear equation system. Its solution corresponds to the optimal pose estimates of all laser scans. As most approaches, this algorithm is restricted to

2D laser scan data. In this paper, we present an extension to 3D scans and poses with 6 DoF, following the style of argumentation in [11].

2 State of the Art

Bergevin et al. [2], Benjema and Schmitt [1], and Pulli [14] presented iterative approaches for 3D scan matching. Based on networks representing overlapping parts of images, they used the ICP algorithm for computing transformations that are applied after all correspondences between all views have been found. However, the focus of research is mainly 3D modelling of small objects using a stationary 3D scanner and a turn table; therefore, the used networks consist mainly of one loop [14]. A probabilistic approach was proposed by Williams et al., where each scan point is assigned a Gaussian distribution in order to model statistically the errors made by laser scanners [18]. In practice, this causes high computation time due to the large amount of data. Krishnan et al. presented a global registration algorithm that minimizes the global error function by optimization on the manifold of 3D rotation matrices [10].

In robotics, many researchers consider similar problems when solving the SLAM (simultaneous localization and mapping) problem [15]. Here an autonomous vehicle builds a map of an unknown environment while processing inherently uncertain sensor data. So-called GraphSLAM techniques represent the global robotic map in a flexible graph structure [8, 9, 11, 13]. However, most of these approaches consider 2D scans and pose estimates with 3 DoF, i.e., motion in a planar environment.

We extend this state of the art by a GraphSLAM method, similar to the approach presented in [17]. Their work, however, is based on a gradient-descent algorithm to minimize the global error function, instead of a closed-form solution. In addition, poses, local point correspondences and global constraints are estimated iteratively, thus increasing the computation requirements of their algorithm and rendering it impractical for a large amount of data.

3 The Estimation Problem

3.1 Problem Formulation

Consider a scanning system traveling along a path, and traversing the $n + 1$ poses V_0, \dots, V_n . At each pose V_i , it takes a laser scan of its environment. By matching two scans made at two different poses, we acquire a set of relations between those poses. In the resulting network, poses are represented as nodes, and relations between them as edges. Given such a network with nodes X_0, \dots, X_n and directed edges $D_{i,j}$, the task is to estimate all poses optimally to build a consistent map of the environment. For simplification, the measurement equation is assumed to be linear:

$$D_{i,j} = X_i - X_j.$$

The observation $\bar{D}_{i,j}$ of the true underlying difference is modeled as $\bar{D}_{i,j} = D_{i,j} + \Delta D_{i,j}$, where the error $\Delta D_{i,j}$ is a Gaussian distributed random variable with zero mean and a known covariance matrix $C_{i,j}$.

Maximum likelihood estimation is used to approximate the optimal poses X_i . Under the assumption that all errors in the observations are Gaussian and distributed independently, maximizing $P(D_{i,j}|\bar{D}_{i,j})$ is equivalent to minimizing the following Mahalanobis distance:

$$\mathbf{W} = \sum_{(i,j)} (D_{i,j} - \bar{D}_{i,j})^T C_{i,j}^{-1} (D_{i,j} - \bar{D}_{i,j}). \quad (1)$$

3.2 Solution as given by Lu and Milios

We consider the simple linear case of the estimation problem. Without loss of generality we assume that the network is fully connected, i.e., each pair of nodes X_i, X_j is connected by a link $D_{i,j}$. In the case of a missing link $D_{i,j}$ we set the corresponding $C_{i,j}^{-1}$ to 0. Eq. (1) unfolds to:

$$\mathbf{W} = \sum_{0 \leq i < j \leq n} (X_i - X_j - \bar{D}_{i,j})^T C_{i,j}^{-1} (X_i - X_j - \bar{D}_{i,j}). \quad (2)$$

To minimize the Eq. (2), a coordinate system is defined by setting one node as a reference point. Setting $X_0 = 0$, the n free nodes X_1, \dots, X_n denote the poses relative to X_0 . Using the signed incidence matrix \mathbf{H} , the concatenated measurement equation \mathbf{D} is written as

$$\mathbf{D} = \mathbf{H}\mathbf{X},$$

with \mathbf{X} the concatenation of X_1 to X_n . The Mahalanobis distance equation can be written as:

$$\mathbf{W} = (\bar{\mathbf{D}} - \mathbf{H}\mathbf{X})^T \mathbf{C}^{-1} (\bar{\mathbf{D}} - \mathbf{H}\mathbf{X}).$$

The concatenation of all observations $\bar{D}_{i,j}$ forms the vector $\bar{\mathbf{D}}$, while \mathbf{C} is a block-diagonal matrix comprised of the covariance matrices $C_{i,j}$ as submatrices. The solution \mathbf{X} that minimizes the equation (2) and its covariance $\mathbf{C}_\mathbf{X}$ is given by

$$\begin{aligned} \mathbf{X} &= (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C}^{-1} \bar{\mathbf{D}} \\ \mathbf{C}_\mathbf{X} &= (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1}. \end{aligned}$$

The matrix $\mathbf{G} = \mathbf{H}^T \mathbf{C}^{-1} \mathbf{H}$ and the vector $\mathbf{B} = \mathbf{H}^T \mathbf{C}^{-1} \bar{\mathbf{D}}$ simplify the solution. \mathbf{G} consists of submatrices

$$G_{i,i} = \sum_{j=0}^n C_{i,j}^{-1} \quad (3)$$

$$G_{i,j} = C_{i,j}^{-1} \quad (i \neq j). \quad (4)$$

The entries of \mathbf{B} are obtained by:

$$B_i = \sum_{\substack{j=0 \\ j \neq i}}^n C_{i,j}^{-1} \bar{D}_{i,j}. \quad (5)$$

Solving the linear optimal estimation problem is equivalent to solving the following linear equation system:

$$\mathbf{G}\mathbf{X} = \mathbf{B}. \quad (6)$$

3.3 The Extension to 6 DoF

The solution of Sec. 3.2 requires the linearization of the pose difference equation. The 3 DoF case, i.e., $(x, y, \theta)^T$ poses, was solved by Lu and Milios [11]. Our algorithm derives relations for 6 DoF poses, i.e., $(x, y, z, \theta_x, \theta_y, \theta_z)^T$, by matching data obtained by a 3D laser range finder. The challenges of this extension are:

1. The amount of data: A 3D laser range finder scans the environment with a large number of samples.
2. Linearization of the rotation must regard the 3 DoF. The rotation consist of the three Euler angles $(\theta_x, \theta_y, \theta_z)$, and the multiplication of the corresponding three rotation matrices result in the desired overall rotation. By using linearization of the Euler angles, we enforce valid rotation matrices.
3. The additional three DoF result in an exponentially larger solution space. The solution is computationally more complex.

We define a 6D pose relation as follows: Assume the first pose to be $V_b = (x_b, y_b, z_b, \theta_{x_b}, \theta_{y_b}, \theta_{z_b})^T$, the second $V_a = (x_a, y_a, z_a, \theta_{x_a}, \theta_{y_a}, \theta_{z_a})^T$, with a pose change of $D = (x, y, z, \theta_x, \theta_y, \theta_z)^T$ of V_a relative to V_b . The poses V_a

and V_b are related by the compound operation $V_a = V_b \oplus D$. Similarly, a 3D position vector $u = (x_u, y_u, z_u)$ is compounded with the pose V_b by $u' = V_b \oplus u$:

$$\begin{aligned} x'_u &= x_b - z_u \sin \theta_{y_b} + \cos \theta_{y_b} (x_u \cos \theta_{z_b} - y_u \sin \theta_{z_b}) \\ y'_u &= y_b + z_u \cos \theta_{y_b} \sin \theta_{x_b} + \cos \theta_{x_b} (y_u \cos \theta_{z_b} + x_u \sin \theta_{z_b}) \\ &\quad + \sin \theta_{x_b} \sin \theta_{y_b} (x_u \cos \theta_{z_b} - y_u \sin \theta_{z_b}) \\ z'_u &= z_b - \sin \theta_{x_b} (y_u \cos \theta_{z_b} + x_u \sin \theta_{z_b}) \\ &\quad + \cos \theta_{x_b} (z_u \cos \theta_{y_b} + \sin \theta_{y_b} (x_u \cos \theta_{z_b} - y_u \sin \theta_{z_b})) \end{aligned}$$

This operation is used to transform a non-oriented point (from the range finder) from its local to the global coordinate system.

Scan matching computes a set of m corresponding point pairs u_k^a, u_k^b between two scans, each representing a single physical point. The positional error made by these point pairs is described by:

$$F_{ab}(V_a, V_b) = \sum_{k=1}^m \|V_a \oplus u_k^a - V_b \oplus u_k^b\|^2 \quad (7)$$

$$= \sum_{k=1}^m \|(V_a \ominus V_b) \oplus u_k^a - u_k^b\|^2. \quad (8)$$

Based on these m point pairs, the algorithm computes the matrices $\bar{D}_{i,j}$ and $C_{i,j}$ for solving Eq. (1). $\bar{D}_{i,j}$ is derived as follows:

Let $\bar{V}_a = (\bar{x}_a, \bar{y}_a, \bar{z}_a, \bar{\theta}_{x_a}, \bar{\theta}_{y_a}, \bar{\theta}_{z_a})$ and $\bar{V}_b = (\bar{x}_b, \bar{y}_b, \bar{z}_b, \bar{\theta}_{x_b}, \bar{\theta}_{y_b}, \bar{\theta}_{z_b})$ be close estimates of V_a and V_b . If the global coordinates of a pair of matching points $u_k = (x_k, y_k, z_k)$ then (u_k^a, u_k^b) fulfill the equation

$$u_k \approx V_a \oplus u_k^a \approx V_b \oplus u_k^b.$$

For small errors $\Delta V_a = \bar{V}_a - V_a$ and $\Delta V_b = \bar{V}_b - V_b$, a Taylor expansion leads to:

$$\begin{aligned} \Delta Z_k &= V_a \oplus u_k^a - V_b \oplus u_k^b := F_k(V_a, V_b) \\ &\approx F_k(\bar{V}_a, \bar{V}_b) - [\nabla_{\bar{V}_a} (F_k(\bar{V}_a, \bar{V}_b)) \Delta V_a \\ &\quad - \nabla_{\bar{V}_b} (F_k(\bar{V}_a, \bar{V}_b)) \Delta V_b] \\ &= \bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b - [\nabla_{\bar{V}_a} (\bar{V}_a \oplus u_k^a) \Delta V_a \\ &\quad - \nabla_{\bar{V}_b} (\bar{V}_b \oplus u_k^b) \Delta V_b] \end{aligned} \quad (10)$$

where $\nabla_{\bar{V}_a} (F_k(\bar{V}_a, \bar{V}_b))$ is the gradient of the pose compounding operation. By matrix decomposition

$$\begin{aligned} M_k H_a &= \nabla_{\bar{V}_a} (F_k(\bar{V}_a, \bar{V}_b)) \\ M_k H_b &= \nabla_{\bar{V}_b} (F_k(\bar{V}_a, \bar{V}_b)), \end{aligned}$$

Eq. (10) simplifies to:

$$\begin{aligned} \Delta Z_k &\approx \bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b - M_k [H_a \Delta V_a - H_b \Delta V_b] \\ &= \bar{Z}_k - M_k D. \end{aligned}$$

with

$$\begin{aligned} \bar{Z}_k &= \bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b \\ D &= (H_a \Delta V_a - H_b \Delta V_b) \\ M_k &= \begin{pmatrix} 1 & 0 & 0 & 0 & -y_k & -z_k \\ 0 & 1 & 0 & z_k & x_k & 0 \\ 0 & 0 & 1 & -y_k & 0 & x_k \end{pmatrix}. \end{aligned} \quad (11)$$

H_a as per Eq. (9), H_b respectively. D , defined by the Eq. (11), is the new linearized measurement equation. To calculate both \bar{D} and C_D , (8) is rewritten in matrix form

$$F_{ab}(D) \approx (\mathbf{Z} - \mathbf{M}D)^T (\mathbf{Z} - \mathbf{M}D).$$

\mathbf{M} is the concatenated matrix consisting of all M_k 's, \mathbf{Z} the concatenated vector consisting of all Z_k 's. The vector \bar{D} that minimizes F_{ab} is given by

$$\bar{D} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{Z}. \quad (12)$$

Since minimizing F_{ab} constitutes a least squares linear regression, we model the Gaussian distribution of the solution with mean \bar{D} and standard covariance estimation

$$C_D = s^2 (\mathbf{M}^T \mathbf{M}). \quad (13)$$

s^2 is the unbiased estimate of the covariance of the identically, independently distributed errors of Z_k , given by:

$$s^2 = \frac{(\mathbf{Z} - \mathbf{M}\bar{D})^T (\mathbf{Z} - \mathbf{M}\bar{D})}{2m - 3} = \frac{F_{ab}(\bar{D})}{2m - 3}.$$

The error term W_{ab} , corresponding to the pose relation, is defined by:

$$W_{ab} = (\bar{D} - D)^T C_D^{-1} (\bar{D} - D).$$

3.4 Transforming the Solution

Solving the linear equation (6) leads to an optimal estimate of the new measurement equation of D (Eq. (11)). To yield an optimal estimation of the poses, it is necessary to transform D and compute a set of solutions via $X_i = H_i \Delta V_i$, each corresponding to a node in the network. Assuming that the reference pose $V_0 = 0$, the pose V_i and its covariance C_i are updated by:

$$\begin{aligned} V_i &= \bar{V}_i - H_i^{-1} X_i, \\ C_i &= (H_i^{-1}) C_i^X (H_i^{-1})^T. \end{aligned}$$

If V_0 is non-zero, the solutions have to be transformed by:

$$\begin{aligned} V_i' &= V_0 \oplus V_i \\ C_i' &= K_0 C_i K_0^T \end{aligned}$$

with

$$K_0 = \begin{pmatrix} R_{\theta_{x_0}, \theta_{y_0}, \theta_{z_0}} & 0 \\ 0 & I_3 \end{pmatrix},$$

$R_{\theta_{x_0}, \theta_{y_0}, \theta_{z_0}}$ denoting a rotation matrix.

$$H_a = \begin{pmatrix} 1 & 0 & 0 & 0 & \bar{z}_a \cos(\bar{\theta}_{x_a}) + \bar{y}_a \sin(\bar{\theta}_{x_a}) & \bar{y}_a \cos(\bar{\theta}_{x_a}) \cos(\bar{\theta}_{y_a}) - \bar{z}_a \cos(\bar{\theta}_{y_a}) \sin(\bar{\theta}_{x_a}) \\ 0 & 1 & 0 & -\bar{z}_a & -\bar{x}_a \sin(\bar{\theta}_{x_a}) & -\bar{x}_a \cos(\bar{\theta}_{x_a}) \cos(\bar{\theta}_{y_a}) - \bar{z}_a \sin(\bar{\theta}_{y_a}) \\ 0 & 0 & 1 & \bar{y}_a & -\bar{x}_a \cos(\bar{\theta}_{x_a}) & \bar{x}_a \cos(\bar{\theta}_{y_a}) \sin(\bar{\theta}_{x_a}) + \bar{y}_a \sin(\bar{\theta}_{y_a}) \\ 0 & 0 & 0 & 1 & 0 & \sin(\bar{\theta}_{y_a}) \\ 0 & 0 & 0 & 0 & \sin(\bar{\theta}_{x_a}) & \cos(\bar{\theta}_{x_a}) \cos(\bar{\theta}_{y_a}) \\ 0 & 0 & 0 & 0 & \cos(\bar{\theta}_{x_a}) & -\cos(\bar{\theta}_{y_a}) \sin(\bar{\theta}_{x_a}) \end{pmatrix} \quad (9)$$

3.5 The Algorithm

Iterative execution of Algorithm 1 yields a successive improvement of the global pose estimation. Step 2 is sped up by component-wise computation of \mathbf{G} and \mathbf{B} . The components $C_{i,j}^{-1} = (\mathbf{M}^T \mathbf{M})/s^2$ and $C_{i,j}^{-1} \bar{D}_{i,j} = (\mathbf{M}^T \mathbf{Z})/s^2$ are expanded into simple summations, as shown in the appendix A.2. The most expensive operation are solving the linear equation system $\mathbf{G}\mathbf{X} = \mathbf{B}$ and the computation of correspondences. Since \mathbf{G} is a positive definite, symmetric $6n \times 6n$ matrix, this is done by Cholesky decomposition.

Algorithm 1 Optimal estimation algorithm (LUM)

1. Compute the point correspondences u_k^a, u_k^b .
 2. For any link (i, j) in the given graph, compute the measurement vector \bar{D}_{ij} by (12) and its covariance C_{ij} by (13).
 3. From all \bar{D}_{ij} and C_{ij} form the linear system $\mathbf{G}\mathbf{X} = \mathbf{B}$, with \mathbf{G} and \mathbf{B} as given in (3)–(5), and solve it for \mathbf{X} .
 4. Update the poses and their covariances, as explained in Sec. 3.4.
-

3.6 Invertibility of \mathbf{G}

The proposed algorithm depends on the invertibility of matrix \mathbf{G} , which is the case if:

1. All covariances are positive or negative definite, and
2. The pose graph is connected, i.e., there exist no two separate subgraphs.

The second condition is met trivially in practice since at least all consecutive poses are linked. The inductive proof over the number of nodes is given in appendix A.1.

4 Performance

The large amount of data to be processed makes computing time an issue in globally consistent range scan

matching. The first step in reducing the computing time is achieved by replacing matrix multiplications by simple summations, as explained in appendix A.2. Again our algorithm benefits from the network structure. Each scan has to be aligned to only few neighbors in the graph. Links exist between consecutive scans in the robot's path and additionally those scans that are spatially close. Consequently, most entries of matrix \mathbf{G} are zero, e.g., \mathbf{G} is sparse (cf. Fig. 1). Since \mathbf{G} is also positive definite (cf. appendix A.1), we apply a sparse Cholesky decomposition to speed up the matrix inversion [6]. Alternative approaches are described in [9, 13].

4.1 Sparse Cholesky Decomposition

By symbolic analysis of the non-zero pattern of matrix \mathbf{G} , a fill-reducing permutation P and an elimination tree are calculated. We use the minimum degree algorithm [7], a heuristic brute force method, to determine the fill-reducing permutation, e.g., a matrix P for which the factorization $P\mathbf{G}P^T$ contains a minimal number of non-zero entries. The elimination tree, a pruned form of the graph G_L that is equivalent to the Cholesky decomposition L , gives the non-zero pattern of L (cf. Fig. 1). The sparse system $\mathbf{G}\mathbf{X} = \mathbf{B}$ becomes $P\mathbf{G}P^T P\mathbf{X} = P\mathbf{B}$, which is solved efficiently with the Cholesky factorisation $LL^T = P\mathbf{G}P^T$ [7]. After solving the equation system $Ly = P\mathbf{B}$ for y , $L^T z = y$ is solved for z , resulting in the solution $\mathbf{X} = P^T z$. Sparse Cholesky factorisation is done in $\mathcal{O}(\text{FLOPS})$, i.e., is linear in the number of graph edges.

4.2 Fast Computation of Correspondences

According to our experience, the most computing time was spent in step 2 of Algorithm 1. While our matching algorithm spends $\mathcal{O}(n)$ time on matrix computation, calculating the corresponding points for a link needs $\mathcal{O}(N \log N)$, using cached k -d tree search [12]. N denotes the number of points per 3D scan, $n \ll N$. To avoid recomputing the k -d trees in each iteration, the query point is transformed into the local coordinate system according to the current scan

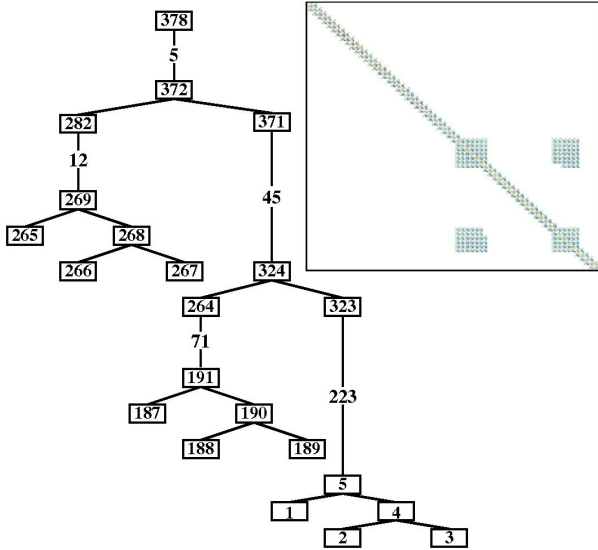


Figure 1. Sparse Matrix G (top right) and its elimination tree generated for the University building data set (cf. Fig. 2, 3). Blue dots in the matrix represent negative, yellow dots positive entries. White spaces are filled with zeros. The numbers on the edges of the tree indicate the number of linearly arranged nodes, removed to improve visibility. The nodes represent the columns of the matrix.

pose. By this means, the k -d trees only have to be computed once at the beginning of global optimization, caching proceeds as described in [12].

5 Experiments and Results

The proposed algorithm has been tested in various experiments. The test strategy consist of two parts: First, 3D environment data has been acquired, collected in a planar indoor environment, enabling a comparison with easily obtainable ground truth. As a second test, we present results from mapping outdoor environments, showing full functionality of the algorithm in all 6 degrees of freedom.

5.1 3D Mapping of Indoor Environments

We used data from the Kurt3D robot, acquired in a University building in Osnabrück. The robot’s path, following the shape of an eight, is shown in the upper left corner of Fig. 3. Fig. 2 displays a top view of the floor. The left side shows the prearranged laser scans, the right shows the resulting alignment after correction with 900 iterations of our algorithm. A more detailed view of the map is presented in Fig. 3. The initial error of up to 18 degrees is reduced



Figure 2. Top view of 3D laser scans before (left) and after correction (right) with 900 iterations of LUM. The data results from 64 scans each containing 81225 (225×361) data points.

Table 1. Comparison of computing times of different matrix inversion and point pair search techniques.

	Absolute time in <i>ms</i>	Relative time in %	Absolute time in <i>ms</i>	Relative time in %
Simple matrix inversion	11709	100	1867	100
Cholesky decomposition	4437	37.89	912	48.84
Sparse Chol. decomposition	1010	8.63	336	17.99
Standard k -d tree search	263551	100	375606	100
k -d trees as of Sec. 4.2	160151	60.78	193066	51.4
	University building		Bridge	

significantly during global registration. The total error is distributed over all laser scans rather than to be summed up with each additional laser scan as is the case in iterative scan matching approaches.

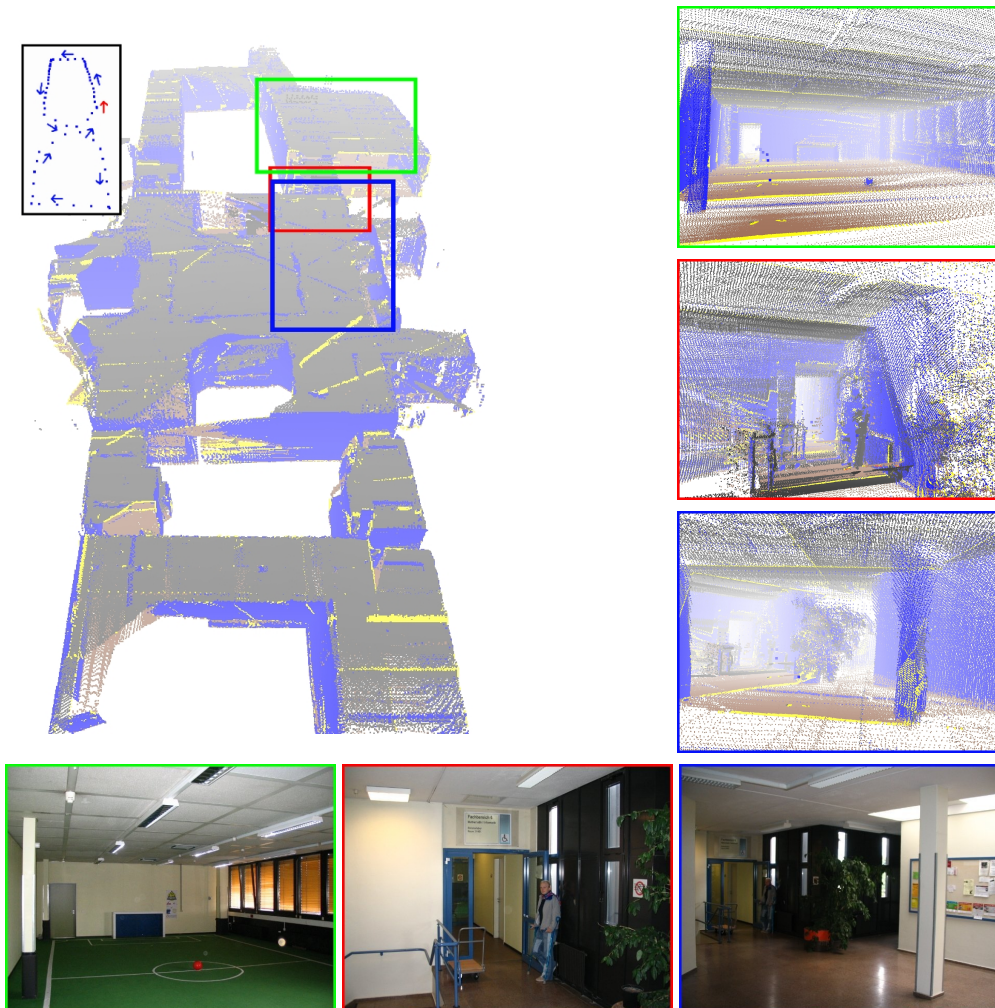


Figure 3. Final 3D map. 3D points are classified as ground (brown), walls (blue) and ceiling (grey). Non-classified points are painted yellow. The closeups on the right correspond to the rectangles marked in the map. Top: Robot soccer field. Middle: Start and end of the robot path, showing a stair rail and a person standing in a doorway. Bottom: Part of the hall where the robot path intersects. The blue squares indicate the robot poses. In the top left corner, the complete robot path of approximately 127 meters is shown, starting at the red arrow.

5.2 6D SLAM in Outdoor Environments

The difference between incremental ICP based mapping and full 6D SLAM becomes obvious in outdoor environments, where the robot motion occurs in all 6 degrees of freedom. The path of the robot led over a bridge (Fig. 4, right), down the hill, underneath the bridge and back up the hill to the starting point. Fig. 4 presents the results of matching the 36 3D scans. In addition to sequential scan matching as with ICP, the network contains links connecting the scans taken on the bridge to those taken from below. ICP scan matching shows poor results, i.e., the bridge appears twice in the resulting point cloud. Our algorithm maps the bridge correctly, including the real thickness of the bridge

(Fig. 4). Further experimental results are reported in [4].

Table 1 shows the computing time needed for matching the two data sets, with different techniques. Of the three approaches to matrix inversion, the sparse Cholesky decomposition is substantially the fastest. Use of the improved k -d tree based point pair calculation further accelerates the algorithm by approximately 40 percent.

6 Conclusions

This paper has presented a method to register laser range scans globally consistent, i.e., without accumulated errors. After building a network of pairwise relations between several laser scans, the solution of a linear system of equations

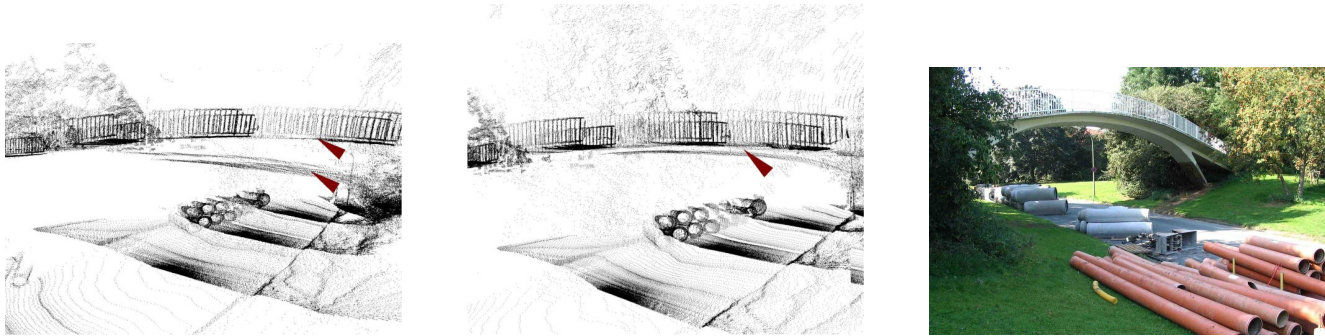


Figure 4. Left and Middle: Closeups on parts of the resulting 3D maps after ICP (left) and LUM (middle) scan matching. LUM has shifted the bridge bottom to the correct distance of the bridge surface. Right: Photo of the outdoor scene that was mapped.

of distance measurements between those scans optimizes the poses by minimizing the distances. As all poses are modified simultaneously, accumulations of local errors are eliminated.

ICP is unable to recover from an incorrect scan matching. In addition, errors in laser scan data and imprecise matching methods lead to accumulated errors in the progress of building large maps, causing inaccuracies in regions where loops are closed. In contrast, our algorithm proved to be more robust due to multiple scan connections in the graph. The instabilities reported in [16] for 2D scans and 3D poses did not occur in the 3D scan/6D pose case.

Acknowledgements

Financial support for this research and development project is provided by the German Federal Ministry for Education and Research (BMBF) as part of the framework program "Forschung für die Produktion von morgen" (Research for tomorrow's production; grants no. 02PB2170-02PB2177). Its administration is handled by the Project Management Agency Forschungszentrum Karlsruhe (PTKA-PFT).

References

- [1] R. Benjema and F. Schmitt. A Solution For The Registration Of Multiple 3D Point Sets Using Unit Quaternions. *Computer Vision – ECCV ’98*, 2:34 – 50, 1998.
- [2] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE TPAMI*, 18(5):540 – 547, 1996.
- [3] P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE TPAMI*, 14(2):239 – 256, 1992.
- [4] D. Borrmann and J. Elseberg. Global Konsistente 3D Kartierung am Beispiel des Botanischen Gartens in Osnabrück. Bachelor’s thesis, Universität Osnabrück, 2006.
- [5] Y. Chen and G. Medioni. Object Modelling by Registration of Multiple Range Images. *Image Vision Comput.*, 10(3):145–155, 1992.
- [6] T. A. Davis. Algorithm 849: A concise sparse Cholesky factorization package. *ACM TOMS*, 31(4), 2005.
- [7] T. A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.
- [8] J. Folkesson and H. Christensen. Graphical SLAM - a self-correcting map. In *Proc. ICRA*, 2004.
- [9] U. Frese. Efficient 6-DOF SLAM with Treemap as a Generic Backend. In *Proc. ICRA*, 2007.
- [10] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian. Global Registration of Multiple 3D Point Sets via Optimization on a Manifold. In *Eurographics SGP*, 2000.
- [11] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4:333 – 349, April 1997.
- [12] A. Nüchter, K. Lingemann, and J. Hertzberg. Cached k -d tree search for icp algorithms. In *Proc. 3DIM*, pages 419 – 426, 2007.
- [13] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proc. ICRA*, 2006.
- [14] K. Pulli. Multiview Registration for Large Data Sets. In *Proc. 3DIM*, pages 160 – 168, 1999.
- [15] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [16] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [17] R. Triebel and W. Burgard. Improving Simultaneous Localization and Mapping in 3D Using Global Constraints. In *Proc. AAAI*, 2005.
- [18] J. Williams and M. Bannamoun. Multiple View 3D Registration using Statistical Error Models. In *VMV*, 1999.

A Appendix

A.1 Proof of Invertibility

The algorithm described in this paper is based on the inversion of matrix \mathbf{G} . We prove that \mathbf{G} is positive definite and therefore invertible using complete induction. In order to simplify the proof, we show that changing the reference pose does not change the positive definite properties of \mathbf{G} . Without loss of generality \mathbf{G} is

a positive definite matrix of the form (3), (4), with the reference node X_0 . Switching the reference node to X_i results in the matrix \mathbf{G}' . These two matrices are related by

$$\mathbf{G}' = \mathbf{I}_i \mathbf{G}$$

where \mathbf{I}_i is an identity matrix of size $(dn \times dn)$ with a row of negative $(d \times d)$ identity matrices of the form:

$$\mathbf{I}_i = \begin{pmatrix} I_{d(i-1)} & 0 & 0 \\ -I_d & \dots & -I_d \\ 0 & 0 & I_{d(n-i+1)} \end{pmatrix}.$$

Multiplication with \mathbf{I}_i corresponds to replacing the submatrices at (i, j) with the negative sum of all submatrices at row j . Since \mathbf{I}_i is invertible, \mathbf{G}' remains positive definite.

Induction base $k = n$ Assuming a graph with $n + 1$ nodes and n links. The matrix \mathbf{G} is transformed into the block diagonal matrix \mathbf{G}' , composed of covariance matrices by

$$\mathbf{G}' = \mathbf{I}_D \mathbf{G} \mathbf{I}_D^T,$$

with an upper-right triangular matrix \mathbf{I}_D of d -dimensional identity matrices

$$\mathbf{I}_D = \begin{pmatrix} I_d & \dots & I_d \\ & \ddots & \vdots \\ 0 & & I_d \end{pmatrix}.$$

Since \mathbf{G}' is given by

$$\begin{aligned} G'_{i,i} &= C_{i-1,i}^{-1} \\ G'_{i,j} &= 0 \quad (i \neq j) \end{aligned}$$

and all covariances are positive definite, \mathbf{G}' itself is positive definite. The same holds for \mathbf{G} , as \mathbf{I}_D is invertible.

Inductive step $k \rightarrow k + 1$ Let \mathbf{G} be a positive definite matrix that corresponds to a graph with $n + 1$ nodes and k links. An additional link between the nodes X_i and X_j is inserted, with positive definite covariance $C_{i,j}$. Without restriction, X_i is the reference node of the given graph, since the reference pose is arbitrary. Thus, the resulting matrix \mathbf{G}' is changed only at the $d \times d$ submatrix $\mathbf{G}'_{j,j}$:

$$G'_{j,j} = G_{j,j} + C_{i,j}^{-1}.$$

In case $C_{i,j}$ is positive definite, \mathbf{G}'^* is positive definite, too, iff

$$\mathbf{X}^T \mathbf{G}' \mathbf{X} > 0 \quad \mathbf{X} \in \mathbb{R}^{d \cdot n} \quad \mathbf{X} \neq \mathbf{0},$$

which is equivalent to

$$\sum_{k,l=1}^n X_k^T G'_{k,l} X_l > 0, \quad (14)$$

where X_k are the d -dimensional subvectors of \mathbf{X} . Expanding Eq. (14) to

$$\begin{aligned} \sum_{k,l=1}^n X_k^T G'_{k,l} X_l &= X_j^T G'_{j,j} X_j + \sum_{\substack{k,l=1 \\ k \neq l \neq j}}^n X_k^T G_{k,l} X_l \\ &= X_j^T C_{i,j}^{-1} X_j + \sum_{k,l=1}^n X_k^T G_{k,l} X_l \\ &= X_j^T C_{i,j}^{-1} X_j + \mathbf{X}^T \mathbf{G} \mathbf{X} > 0. \end{aligned}$$

\mathbf{G}' is a positive definite matrix. \square

A.2 Fast construction of the linear equation system

To solve the linear equation system $\mathbf{G} \mathbf{X} = \mathbf{B}$,

$$C_D^{-1} = (\mathbf{M}^T \mathbf{M}) / s^2 \quad \text{and} \quad C_D^{-1} \bar{D} = (\mathbf{M}^T \mathbf{Z}) / s^2$$

are needed. To calculate these efficiently, summations are substituted for matrix multiplication by using the regularities in the matrix \mathbf{M} . $\mathbf{M}^T \mathbf{M}$ is represented as a sum over all corresponding point pairs: $\mathbf{M}^T \mathbf{M} =$

$$\sum_{k=1}^m \begin{pmatrix} 1 & 0 & 0 & 0 & -y_k & -z_k \\ 0 & 1 & 0 & z_k & x_k & 0 \\ 0 & 0 & 1 & -y_k & 0 & x_k \\ 0 & z_k & -y_k & y_k^2 + z_k^2 & x_k z_k & -x_k y_k \\ -y_k & x_k & 0 & x_k z_k & y_k^2 + x_k^2 & y_k z_k \\ -z_k & 0 & x_k & -x_k y_k & y_k z_k & x_k^2 + z_k^2 \end{pmatrix}.$$

Similarly, $\mathbf{M}^T \mathbf{Z}$ is calculated as follows:

$$\mathbf{M}^T \mathbf{Z} = \sum_{k=0}^m \begin{pmatrix} \Delta x_k \\ \Delta y_k \\ \Delta z_k \\ -z_k \cdot \Delta y_k + y_k \cdot \Delta z_k \\ -y_k \cdot \Delta x_k + x_k \cdot \Delta y_k \\ z_k \cdot \Delta x_k - x_k \cdot \Delta z_k \end{pmatrix}$$

with

$$\begin{pmatrix} \Delta x_k \\ \Delta y_k \\ \Delta z_k \end{pmatrix} = \bar{Z}_k = \bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b$$

and an approximation for each point:

$$\begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} = u_k \approx (\bar{V}_a \oplus u_k^a + \bar{V}_b \oplus u_k^b) / 2.$$

Finally s^2 is a simple summation using the observation of the linearized measurement equation $\bar{D} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{Z}$:

$$\begin{aligned} s^2 &= \sum_{k=0}^m (\Delta x_k - (\bar{D}_0 - y_k \cdot \bar{D}_4 + z_k \cdot \bar{D}_5))^2 \\ &\quad + (\Delta y_k - (\bar{D}_1 - z_k \cdot \bar{D}_3 + x_k \cdot \bar{D}_4))^2 \\ &\quad + (\Delta z_k - (\bar{D}_2 + y_k \cdot \bar{D}_3 - x_k \cdot \bar{D}_5))^2. \end{aligned}$$

\bar{D}_i denotes the i -th entry of the vector \bar{D} . Summation of C_D^{-1} and $C_D^{-1} \bar{D}$ yields \mathbf{B} and \mathbf{G} .