Linearization of Rotations for Globally Consistent n-Scan Matching

Andreas Nüchter, Jan Elseberg, Peter Schneider, and Dietrich Paulus

***Abstract*— The ICP (Iterative Closest Point) algorithm is the de facto standard for geometric alignment of three-dimensional models when an initial relative pose estimate is available. The basis of the algorithm is the minimization of an error function that takes point correspondences into account. While four closed-form solution methods are known for minimizing this function, linearization seems necessary for solving the global scan registration problem. This paper presents such linear solutions for registering n-scans in a global and simultaneous fashion. It studies parameterizations for the rigid body transformations of the n-scan registration problem.**

I. Introduction

Registering 3D models is a crucial step in 3D model construction and 3D mapping. Many applications use the ICP algorithm: Nowadays precise 3D scanners are available that are used in architecture, industrial automation, agriculture, cultural heritage conservation, and facility management. Other applications of point cloud registration algorithms include medical data processing, art history, archaeology, and rescue and inspection robotics. The advent of 3D TOF cameras is likely to generate another burst of ICP applications in the near future [2], [1].

The ICP algorithm registers two independently acquired 3D scans or 3D point clouds into a common coordinate system. Here the algorithm relies on minimizing an error function over closest point correspondences. The following analogy is often used to depict the minimization issue: The correspondences represent a system of springs that forces the scan to be aligned to the correct position. Four closed form solution methods are known for minimizing the ICP error function [18]. The difficulty in minimizing the ICP error function is to ensure the orthonormality constraint of the included rotation matrix. This paper presents linearized solution methods, where the optimal rotation is approximated by solving a system of linear equations. The linearization methods are the helix transform, the small angle approximation, and the uncertainty based registrations. For the latter one we model the scan pose, i.e., position and orientation, as Gaussian distributions and use the Euler and the quaternion representation. Our objective is to compare different gradient-descent algorithms based upon different rotation parameterizations to solve the bundle adjustment problem that arises with 3D point clouds produced by range finders. In [24] the recommendation concerning the parameterization choice is stated as: "*Similarly, experience suggests that quasi-global 3 parameter rotation parameterizations such as Euler angles cause numerical problems unless one can be certain to avoid their singularities and regions of uneven coverage. Rotations should be parameterized using either quaternions subject to $|\dot{q}|^2 = 1$, or local perturbations $\mathbf{R}\delta\mathbf{R}$ or $\delta\mathbf{R}\mathbf{R}$ of an existing rotation $\mathbf{R}$, where $\delta\mathbf{R}$ can be any well-behaved 3 parameter small rotation approximation, e.g. $\delta\mathbf{R} = (\mathbf{I}+[\delta\mathbf{r}]_\times)$, the Rodriguez formula, local Euler angles, etc.*". Recently, this was used by by Grisetti et al. [14] for 3D mapping. This paper studies alternative methods.

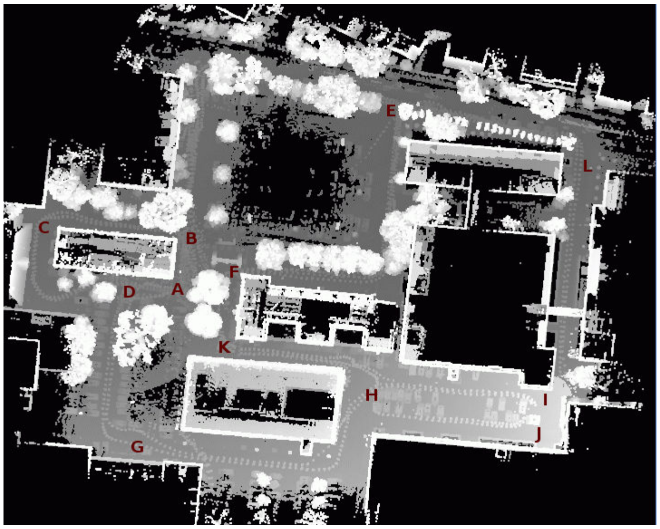

Fig. 1. Registration of n-scans. Top: 3D scene HANNOVER2 of an urban environment in a bird's-eye view. The scans have been taken according to the sequence: A-B-C-D-A-B-E-F-A-D-G-H-I-J-H-K-F-E-L-I-K-A. A video of the scan matching process can be found under `http://plum.eecs.jacobs-university.de/download/icra2010/large_slam.mpg`

If n-scans have to be registered, any sequential application of ICP will accumulate errors, and therefore the ICP algorithm has to be extended. Locally consistent algorithms retain the analogy of the spring system [13] and the resulting algorithms need iterations for minimizing the global error function. However, globally consistent algorithms minimize the error function in one step. Therefore, it is more likely to reach the global and hopefully the correct optimum. Fig. 1 presents an urban scene, digitalized by roughly 1000 terrestrial 3D scans and Fig. 2 shows the accumulated error when performing sequential ICP scan matching and the result of globally consistent scan matching. Fig. 3 stresses the difference between global and local optimal results and gives a schematic illustration of the difference.

This paper presents novel global optimal solutions to the global n-scan registration problem and stresses the role of linearization. The presented methods are examined with respect to computational requirements and to approximation and implementation issues.

A. Nüchter and J. Elseberg are with the Jacobs University Bremen gGmbH, Germany. {a.nuechter}@jacobs.university.de

P. Schneider and D. Paulus are with the Institute of Computervisualistics at the University of Koblenz-Landau, Germany.

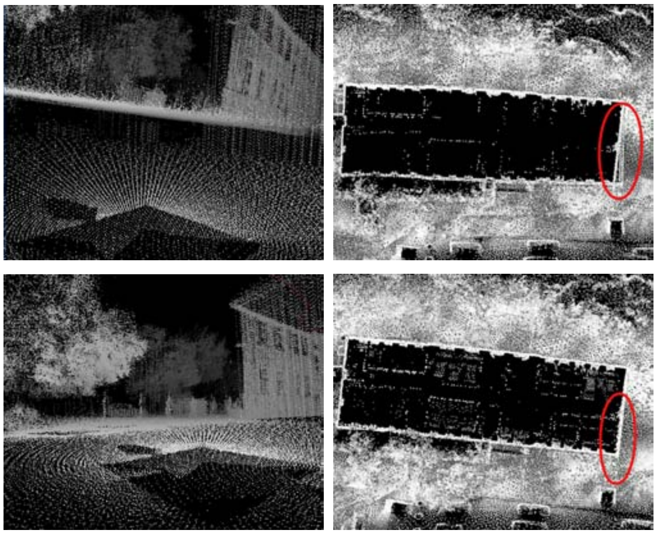

Fig. 2. Top left: 3D view of the scene with the accumulated error. Top right: Scene in a bird's-eye view. Bottom row: Consistent registration results. Left: Globally consistent registration in 3D view. Right: Bird's eye view.

II. State of the Art

a) The ICP Algorithm.: The following method is the de-facto standard for registration of two point sets. The complete algorithm was invented at the same time in 1991 by Besl and McKay [8], by Chen and Medioni [11] and by Zhang [27]. The method is called the *Iterative Closest Points (ICP) algorithm.*

Given two independently acquired sets of 3D points, $\hat{M}$ (model set) and $\hat{D}$ (data set) which correspond to a single shape, we want to find the transformation $(\mathbf{R}, \mathbf{t})$ consisting of a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$ which minimizes the following cost function:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1} ||\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})||^2 , \qquad (1)$$

All corresponding points can be represented in a tuple $(\mathbf{m}_i, \mathbf{d}_i)$ where $\mathbf{m}_i \in M \subset \hat{M}$ and $\mathbf{d}_i \in D \subset \hat{D}$. Two things have to be calculated: First, the corresponding points, and second, the transformation $(\mathbf{R},\ \mathbf{t})$ that minimizes $E(\mathbf{R}, \mathbf{t})$ on the basis of the corresponding points. The ICP algorithm uses closest points as corresponding points. A sufficiently good starting guess enables the ICP algorithm to converge to the correct minimum.

Current research in the context of ICP algorithms mainly focuses on fast variants of ICP algorithms [22]. If the input are 3D meshes then a point-to-plane metric can be used instead of Eq. (1). Minimizing using a point-to-plane metric outperforms the standard point-to-point one, but requires the computation of normals and meshes in a preprocessing step.

b) Globally Consistent n-Scan Matching.: Chen and Medioni [12] aimed at globally consistent range image alignment when introducing an incremental matching method, i.e., all new scans are registered against the so-called metascan, which is the union of the previously acquired and registered scans. This method does not spread out the error and is order-dependent. Bergevin et al. [7], Stoddart and Hilton [23], Benjemaa and Schmitt [5], [6], and Pulli [21] present iterative approaches. Based on networks representing overlapping parts of images, they use 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 modeling of small objects using a stationary 3D scanner and a turn table. Therefore, the used networks consist mainly of one loop [21]. These solutions are locally consistent algorithms that retain the analogy of the spring system [13] whereas true globally consistent algorithms minimize the error function in one step (cf. Fig. 3).

A probabilistic approach was proposed by Williams et al. [25], where each scan point is assigned a Gaussian distribution in order to model the statistical errors made by laser scanners. This causes high computation time due to the large amount of data in practice. Krishnan et al. [17] presented a global registration algorithm that minimizes the global error function by optimization on the manifold of 3D rotation matrices.

III. Linear Solutions for n-Scan Matching

To register scans globally consistent a network of poses is formed, i.e., a graph. Every edge represents a link $j \to k$ of matchable poses. The error function is extended to include all links and to minimize for all rotations and translations at the same time.

$$E = \sum_{j \to k} \sum_{i=1} ||(\mathbf{R}_j\mathbf{m}_i + \mathbf{t}_j) - (\mathbf{R}_k\mathbf{d}_i + \mathbf{t}_k)||^2 , \qquad (2)$$

Four methods can be used to minimize the error function (2) that differ in linearization of the rotation, namely the helix transform, which is due to Hofer and Pottman [16], a small angle approximation and two uncertainty based methods that represent the scan poses (rotation and translation) as Gaussian distributed random variables. These methods are discussed next.

A. Registration using the Helix Transform

Under the assumption that the transformations $(\mathbf{R}, \mathbf{t})$ that have to be calculated are small, one can approximate the solution by applying instantaneous kinematics. Instantaneous

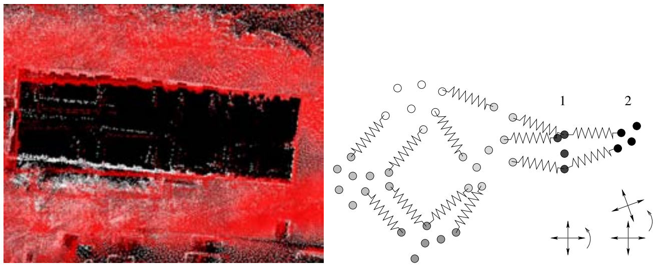

Fig. 3. Left: Locally consistent algorithms reduce the registration errors at the loop closing point but fail to distribute the error. Here in comparison with the global optimal result in bird's eye view. (cf. Fig. 2). Right: Registration of 6 Scans, whose points are represented by different gray values. Algorithms working locally optimal move every scan separately, while global optimal algorithms minimize all distances at the same time. This includes especially, that the transformation of scan 1 is continued to scan 2 (the rightmost one), i.e., incorporating a ”‘leverage effect”’.

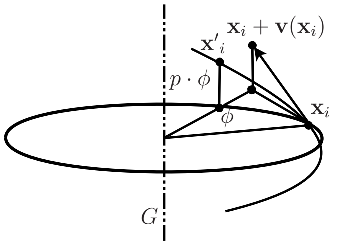

Fig. 4. The affine position of a 3D point $\mathbf{x}_i + \mathbf{v}(\mathbf{x}_i)$ is different from the rigid transformation that results in point $\mathbf{x}'_i$. Based on [16].

kinematics computes the displacement of a 3D point by an affine transformation via a so-called helical motion [20]. A 3D point $\mathbf{d}_i$ is mapped by adding a small displacement $\mathbf{v}(\mathbf{d}_i)$, i.e.,

$$\mathbf{v}(\mathbf{d}_i) = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{d}_i \tag{3}$$

This displacement corresponds to an affine motion. Fig. 4 presents the displacement of a point using the affine transformation and rigid transformation. Therefore in a post processing step a rigid transformation $(\mathbf{R}, \mathbf{t})$ is calculated from $(\bar{\mathbf{c}}, \mathbf{c})$ as follows: If $\mathbf{c} = \mathbf{0}$ only a translation is present. In this case $\mathbf{t} = \bar{\mathbf{c}}$ holds. Otherwise an axis G consisting of a direction vector $\mathbf{g}$ and a momentum vector $\bar{\mathbf{g}}$ can be computed. The tuple $(\mathbf{g}, \bar{\mathbf{g}})$ are the Plücker coordinates of the axis G:

$$\mathbf{g} = \frac{\mathbf{c}}{||\mathbf{c}||}, \quad \bar{\mathbf{g}} = \frac{\mathbf{c} - p\mathbf{c}}{||\mathbf{c}||}, \quad p = \frac{\mathbf{c}^T \cdot \bar{\mathbf{c}}}{\mathbf{c}^2}$$

Based on G the point $\mathbf{d}_i$ has to be transformed as follows:

$$\mathbf{d}'_i = \mathbf{R}(\mathbf{d}_i - \mathbf{p}) + (p\ \phi)\mathbf{g} + \mathbf{p} \tag{4}$$

Here $\mathbf{R}$ is the rotation matrix $\mathbf{R} =$

$$\frac{1}{b_0^2+b_1^2+b_2^2+b_3^2} \begin{pmatrix} b_0^2 + b_1^2 - b_2^2 - b_3^2 & 2(b_1 b_2 + b_0 b_3) & 2(b_1 b_3 - b_0 b_2) \\ 2(b_1 b_2 - b_0 b_3) & b_0^2 - b_1^2 + b_2^2 - b_3^2 & 2(b_2 b_3 + b_0 b_1) \\ 2(b_1 b_3 + b_0 b_2) & 2(b_2 b_3 - b_0 b_1) & b_0^2 - b_1^2 - b_2^2 + b_3^2 \end{pmatrix},$$

where $b_0 = \cos(\phi/2)$, $b_1 = g_x \sin(\phi/2)$, $b_2 = g_y \sin(\phi/2)$, and $b_3 = g_z \sin(\phi/2)$. $\mathbf{g} = (g_x, g_y, g_z)$ is the above mentioned direction vector of the axis G. Furthermore in Eq. (4) $\mathbf{p}$ is an arbitrary point on the axis G. Note: The above Eq. is similar to the term for computing a rotation matrix from a unit quaternion. Please refer to [16] for more details.

Next, we minimize the error function for global consistent scan matching, i.e., we improve the approach given in [20] which is only locally consistent to include off-diagonal elements in the resulting system of equations [15]. To minimize the error function using the new displacement Eq. (2) is rewritten as follows:

$$E = \sum_{j \to k} \sum_i (\mathbf{m}_i - \mathbf{d}_i + (\bar{\mathbf{c}}_j + \mathbf{c}_j \times \mathbf{m}_i) - (\bar{\mathbf{c}}_k + \mathbf{c}_k \times \mathbf{m}_i))^2$$

Reformulating E as

$$E = \mathbf{X}^T \cdot \mathbf{B} \cdot \mathbf{X} + 2\mathbf{A} \cdot \mathbf{X} + \sum_i (\mathbf{m}_i - \mathbf{d}_i)^2,$$

allows us to solve for the optimal poses $\mathbf{X} = (\mathbf{c}_2^T, \bar{\mathbf{c}}_2^T, \ldots, \mathbf{c}_n^T, \bar{\mathbf{c}}_n^T)^T$ in the linear equation system:

$$\mathbf{B}\mathbf{X} = \mathbf{A}.$$

Note that the first pose is fixed to the origin of the coordinate system, i.e. $\mathbf{c}_1 = \bar{\mathbf{c}}_1 = 0$. $\mathbf{A}$ is attained as follows:

$$\begin{aligned} 2\mathbf{A}\mathbf{X} &= \sum_{j \to k} \sum_i 2((\mathbf{m}_i - \mathbf{d}_i)(\bar{\mathbf{c}}_j + \mathbf{c}_j \times \mathbf{m}_i) \\ &\quad + (\mathbf{d}_i - \mathbf{m}_i)(\bar{\mathbf{c}}_k + \mathbf{c}_k \times \mathbf{m}_i)) \\ \mathbf{A}_j &= \sum_{j \to k} \sum_i \begin{pmatrix} \mathbf{m}_i \times (\mathbf{m}_i - \mathbf{d}_i) \\ \mathbf{m}_i - \mathbf{d}_i \end{pmatrix} \\ &\quad + \sum_{k \to j} \sum_i \begin{pmatrix} \mathbf{m}_i \times (\mathbf{d}_i - \mathbf{m}_i) \\ \mathbf{d}_i - \mathbf{m}_i \end{pmatrix} \end{aligned}$$

The matrix $\mathbf{B}$ is defined by

$$\mathbf{X}^T \cdot \mathbf{B} \cdot \mathbf{X} = \sum_{j \to k} \sum_i ((\bar{\mathbf{c}}_j + \mathbf{c}_j \times \mathbf{m}_i) - (\bar{\mathbf{c}}_k + \mathbf{c}_k \times \mathbf{m}_i))^2,$$

and given as

$$\mathbf{B}_{j,j} = \sum_{\substack{j \to k \\ k \to j}} \sum_i \mathbf{M}_i, \quad \mathbf{B}_{j,k} = \sum_{j \to k} \sum_i -\mathbf{M}_i,$$

where

$$\mathbf{M}_i = \begin{pmatrix} m_{y,i}^2 + m_{z,i}^2 & -m_{x,i} m_{y,i} & -m_{x,i} m_{z,i} & 0 & -m_{z,i} & m_{y,i} \\ -m_{x,i} m_{y,i} & m_{x,i}^2 + m_{z,i}^2 & -m_{y,i} m_{z,i} & m_{z,i} & 0 & -m_{x,i} \\ -m_{x,i} m_{z,i} & -m_{y,i} m_{z,i} & m_{x,i}^2 + m_{y,i}^2 & -m_{y,i} & m_{x,i} & 0 \\ 0 & m_{z,i} & -m_{y,i} & 1 & 0 & 0 \\ -m_{z,i} & 0 & m_{x,i} & 0 & 1 & 0 \\ m_{y,i} & -m_{x,i} & 0 & 0 & 0 & 1 \end{pmatrix}.$$

After the linear equation system is solved the computed tuples $(\mathbf{c}_j, \bar{\mathbf{c}}_j)$ are used to calculate the corresponding rotations and translations $(\mathbf{R}_j, \mathbf{t}_j)$ by the post processing as explained above.

B. Exploiting the Small Angle Approximation

Given a rotation matrix $\mathbf{R}$ based on the Euler angles

$$\mathbf{R} = \mathbf{R}_{\theta_x} \mathbf{R}_{\theta_y} \mathbf{R}_{\theta_z} \tag{5}$$

we use the first-order Taylor series approximation that is valid for small angles

$$\sin\theta \approx \theta - \frac{\theta^3}{3} + \frac{\theta^5}{5} - \cdots, \quad \cos\theta \approx 1 - \frac{\theta^2}{2} + \frac{\theta^4}{4} - \cdots$$

and apply it to (5). As a second approximation we assume that the result of a multiplication of small angles yields even smaller values that can be omitted as well. This eliminates second order and higher combination terms and Eq. (5) becomes:

$$\mathbf{R} \approx \begin{pmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{pmatrix} \tag{6}$$

We notice that this term is closely related to the Rodrigues formula but is no longer orthonormal.

Replacing this approximation (6) in $\mathbf{R}\mathbf{d}_i$ and rearranging the unknown variables in a vector yields:

$$\begin{aligned} \mathbf{R}_j \mathbf{d}_i &= \begin{pmatrix} 0 & d_{z,i} & -d_{y,i} \\ -d_{z,i} & 0 & d_{x,i} \\ d_{y,i} & -d_{x,i} & 0 \end{pmatrix} \cdot \begin{pmatrix} \theta_z \\ \theta_y \\ \theta_x \end{pmatrix} + \mathbf{d}_i && (7) \\ &= \mathbf{D}_i \cdot \mathbf{X}_j + \mathbf{d}_i. && (8) \end{aligned}$$

Surprisingly, the resulting equation is equal to the notation of Eq. (3) by Hofer and Pottmann [16], [20]. Therefore, the result has to be interpreted as a so-called helical motion and not using the small angle assumption. The small angle approximation fails, since the rotation calculated by the ICP algorithm always refers to the global coordinate system, i.e., represents a rotation about the origin $(0, 0, 0)$. While the helical motion takes care of this by calculating a rotation axis, our approximation does not regard this. To make the small angle approximation Eq. (6) work, we have to apply the centroid trick. Using the centroids $\mathbf{c}_j$ and $\mathbf{c}_k$, where $\mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_k$, $\mathbf{d}'_i = \mathbf{d}_i - \mathbf{c}_j$, Eq. (2) is restated as:

$$\begin{aligned}E = &\sum_{j\rightarrow k} \sum_i \left|\mathbf{R}_j\mathbf{m}'_i + \mathbf{R}_j\mathbf{c}_k + \mathbf{t}_j - (\mathbf{R}_k\mathbf{d}'_i + \mathbf{R}_k\mathbf{c}_j + \mathbf{t}_k)\right|^2 \\ = &\sum_{j\rightarrow k} \left(\sum_i \left|\mathbf{R}_j\mathbf{m}'_i - \mathbf{R}_k\mathbf{d}'_i\right|^2 \right. \\ &\quad -2\sum_i (\mathbf{t}_k - \mathbf{t}_j + \mathbf{R}_k\mathbf{c}_j - \mathbf{R}_j\mathbf{c}_k)(\mathbf{R}_j\mathbf{m}'_i - \mathbf{R}_k\mathbf{d}'_i) \\ &\quad \left. + \sum_i \left|\mathbf{t}_k - \mathbf{t}_j + \mathbf{R}_k\mathbf{c}_j - \mathbf{R}_j\mathbf{c}_k\right|^2 \right).\end{aligned}$$

The term $-2\sum_i(\mathbf{t}_k - \mathbf{t}_j + \mathbf{R}_k\mathbf{c}_j - \mathbf{R}_j\mathbf{c}_k)(\mathbf{R}_j\mathbf{m}'_i - \mathbf{R}_k\mathbf{d}'_i)$ equates to zero, because all values refer to the centroids. This separates the rotation from the translation and enables us to solve for the rotation of all poses independent of their translation,

1) Computing the Rotation: Minimizing the first term of the restated error metric allows us to derive the optimal rotation of all poses. Each rotation $\mathbf{R}_j$ is represented as $\mathbf{X}_j = (\theta_{z,j}, \theta_{y,j}, \theta_{x,j})^T$. The following rotational error will be minimized:

$$\begin{aligned}E_R = &\sum_{j\rightarrow k} \sum_i \left(\mathbf{M}_i \cdot \mathbf{X}_j - \mathbf{D}_i \cdot \mathbf{X}_k - (\mathbf{m}_i - \mathbf{d}_i)\right)^2 \\ = &\sum_{j\rightarrow k} \sum_i \left(\mathbf{M}_i \cdot \mathbf{X}_j - \mathbf{D}_i \cdot \mathbf{X}_k\right)^2 + (\mathbf{m}_i - \mathbf{d}_i)^2 \\ &- 2\left(\mathbf{M}_i \cdot \mathbf{X}_j - \mathbf{D}_i \cdot \mathbf{X}_i\right) \cdot (\mathbf{m}_i - \mathbf{d}_i),\end{aligned}$$

where $\mathbf{M}_i$ and $\mathbf{D}_i$ are given as in Eq. (8). The error term is rewritten with the rotations concatenated in $\mathbf{X}$

$$E_R = \mathbf{X}\mathbf{B}\mathbf{X} + 2\mathbf{A}\mathbf{X} + (\mathbf{m}_k - \mathbf{d}_k)^2$$

in order to solve the linear equation system

$$\mathbf{B}\mathbf{X} + \mathbf{A} = 0.$$

With $\mathbf{B}$ given by:

$$\mathbf{B}_{j,j} = \sum_{j\rightarrow k} \sum_i \mathbf{D}_i^T \cdot \mathbf{D}_i + \sum_{k\rightarrow j} \sum_i \mathbf{M}_i^T \cdot \mathbf{M}_i$$

case $j < k$:

$$\mathbf{B}_{j,k} = -\sum_{j\rightarrow k} \sum_i \mathbf{M}_i^T \cdot \mathbf{D}_i$$

case $j > k$:

$$\mathbf{B}_{j,k} = -\sum_{j\rightarrow k} \sum_i \mathbf{D}_i^T \cdot \mathbf{M}_i,$$

and $\mathbf{A}$ by:

$$\begin{aligned}\mathbf{A}_j = &\sum_{k\rightarrow j} \sum_i \begin{pmatrix} (m_{z,i} - d_{z,i}) \cdot d_{y,i} - (m_{y,i} - d_{y,i}) \cdot d_{z,i} \\ (m_{x,i} - d_{x,i}) \cdot d_{z,i} - (m_{z,i} - d_{z,i}) \cdot d_{x,i} \\ (m_{y,i} - d_{y,i}) \cdot d_{x,i} - (m_{x,i} - d_{x,i}) \cdot d_{y,i} \end{pmatrix} \\ &-\sum_{j\rightarrow k} \sum_i \begin{pmatrix} (m_{z,i} - d_{z,i}) \cdot m_{y,i} - (m_{y,i} - d_{y,i}) \cdot m_{z,i} \\ (m_{x,i} - d_{x,i}) \cdot m_{z,i} - (m_{z,i} - d_{z,i}) \cdot m_{x,i} \\ (m_{y,i} - d_{y,i}) \cdot m_{x,i} - (m_{x,i} - d_{x,i}) \cdot m_{y,i} \end{pmatrix}\end{aligned}$$

2) Computing the Translation: With the optimal rotations successfully calculated, the optimal translation is determined by minimizing the term:

$$E_T = \sum_{j\rightarrow k} \sum_i \left(\mathbf{t}_k - \mathbf{t}_j + \mathbf{R}_k\mathbf{c}_j - \mathbf{R}_j\mathbf{c}_k\right)^2.$$

Let $\mathbf{R}_k\mathbf{c}_j - \mathbf{R}_j\mathbf{c}_k$ be abbreviated as $\mathbf{R}_{j,k}$ and E_T is restated in matrix notation:

$$E_T = \mathbf{T}^T\mathbf{B}\mathbf{T} + 2\mathbf{A}\mathbf{T} + \sum_{j\rightarrow k} \sum_i \mathbf{R}_{j,k}^2.$$

This is minimized by solving the linear equation system:

$$\mathbf{B}\mathbf{T} + \mathbf{A} = 0$$

Here, $\mathbf{B}$ is given by:

$$\mathbf{B}_{j,j} = \sum_{\substack{j\rightarrow k \\ k\rightarrow j}} \mathbf{I}, \quad \mathbf{B}_{j,k} = -\sum_{j\rightarrow k} \mathbf{I}$$

and $\mathbf{A}$ by:

$$\mathbf{A}_j = \sum_{j\rightarrow k} \mathbf{R}_j\mathbf{c}_k - \mathbf{R}_k\mathbf{c}_j - \sum_{k\rightarrow j} \mathbf{R}_j\mathbf{c}_k - \mathbf{R}_k\mathbf{c}_j.$$

C. Uncertainty-based Registration

For some applications it is necessary to have a notion of the uncertainty of the poses calculated by the registration algorithm. The following is the extension of the probabilistic approach first proposed in [19] to 6 DoF. This extention is not straightforward, since the matrix decomposition, i.e., Eq. (9) cannot be derived from first principles. For a more detailed description of these extension refer to [9] and [10]. In addition to the poses $\mathbf{X}_j$, the pose estimates $\bar{\mathbf{X}}_j$ and the pose errors $\Delta\mathbf{X}_j$ are required.

The positional error of two poses $\mathbf{X}_j$ and $\mathbf{X}_k$ is described by:

$$E_{j,k} = \sum_{i=1}^{m} \left\|\mathbf{X}_j \oplus \mathbf{d}_i - \mathbf{X}_k \oplus \mathbf{m}_i\right\|^2 = \sum_{i=1}^{m} \left\|\mathbf{Z}_i(\mathbf{X}_j, \mathbf{X}_k)\right\|^2$$

Here, $\oplus$ is the compounding operation that transforms a point into the global coordinate system. For small pose differences, $E_{j,k}$ can be linearized by use of a Taylor expansion:

$$\begin{aligned}\mathbf{Z}_i(\mathbf{X}_j, \mathbf{X}_k) \approx &\ \bar{\mathbf{X}}_j \oplus \mathbf{d}_i - \bar{\mathbf{X}}_k \oplus \mathbf{m}_i \\ &- \left(\nabla_j \mathbf{Z}_i(\bar{\mathbf{X}}_j, \bar{\mathbf{X}}_k)\Delta\mathbf{X}_j - \nabla_k \mathbf{Z}_i(\bar{\mathbf{X}}_j, \bar{\mathbf{X}}_k)\Delta\mathbf{X}_k\right)\end{aligned}$$

where ∇_j, ∇_k denotes the derivative with respect to $\mathbf{X}_j$ and $\mathbf{X}_k$ respectively. Utilizing the matrix decompositions $\mathbf{M}_i\mathbf{H}_j$ and $\mathbf{D}_i\mathbf{H}_k$ of the respective derivatives that separates the poses from the associated points gives:

$$\begin{aligned}\mathbf{Z}_i(\mathbf{X}_j, \mathbf{X}_k) &= \mathbf{Z}_i(\bar{\mathbf{X}}, \bar{\mathbf{X}}_k) - (\mathbf{M}_i\mathbf{H}_j\Delta\mathbf{X}_j - \mathbf{D}_i\mathbf{H}_k\Delta\mathbf{X}_k) \\ &= \mathbf{Z}_i(\bar{\mathbf{X}}, \bar{\mathbf{X}}_k) - \left(\mathbf{M}_i\mathbf{X}'_j - \mathbf{D}_i\mathbf{X}'_k\right)\end{aligned}$$

Appropriate decompositions are given for both the Euler angles and quaternion representation in the following paragraphs. Because $\mathbf{M}_i$ as well as $\mathbf{D}_i$ are independent of the pose, the positional error $E_{j,k}$ is minimized with respect to the new pose difference $\mathbf{E}'_{j,k}$:

$$\begin{aligned}\mathbf{E}'_{j,k} &= (\mathbf{H}_j\Delta\mathbf{X}_j - \mathbf{H}_k\Delta\mathbf{X}_k)\\ &= (\mathbf{X}'_j - \mathbf{X}'_k).\end{aligned}$$

$\mathbf{E}'_{j,k}$ is linear in the quantities $\mathbf{X}'_j$ that will be estimated so that the minimum of $E_{j,k}$ and the corresponding covariance are given by

$$\begin{aligned}\bar{\mathbf{E}}_{j,k} &= (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{Z}\\ \mathbf{C}_{j,k} &= s^2(\mathbf{M}^T\mathbf{M}).\end{aligned}$$

where s^2 is the unbiased estimate of the covariance of the identically, independently distributed errors of $\mathbf{Z}$:

$$s^2 = (\mathbf{Z} - \mathbf{M}\bar{\mathbf{E}})^T(\mathbf{Z} - \mathbf{M}\bar{\mathbf{E}})/(2m-3).$$

Here $\mathbf{Z}$ is the concatenated vector consisting of all $\mathbf{Z}_i(\bar{\mathbf{X}}_j, \bar{\mathbf{X}}_k)$ and $\mathbf{M}$ the concatenation of all $\mathbf{M}_i$'s.

Up to now all considerations have been on a local scale. With the linearized error metric $\mathbf{E}'_{j,k}$ and the Gaussian distribution $(\bar{\mathbf{E}}_{j,k}, \mathbf{C}_{j,k})$ a Mahalanobis distance that describes the global error of all the poses is constructed:

$$\begin{aligned}\mathbf{W} &= \sum_{j\to k}(\bar{\mathbf{E}}_{j,k} - \mathbf{E}'_{j,k})^T\mathbf{C}^{-1}_{j,k}(\bar{\mathbf{E}}'_{j,k} - \mathbf{E}'_{j,k})\\ &= \sum_{j\to k}(\bar{\mathbf{E}}_{j,k} - (\mathbf{X}'_j - \mathbf{X}'_k))\mathbf{C}^{-1}_{j,k}(\bar{\mathbf{E}}'_{j,k} - (\mathbf{X}'_j - \mathbf{X}'_k)).\end{aligned}$$

In matrix notation, $\mathbf{W}$ becomes:

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

Here $\mathbf{H}$ is the signed incidence matrix of the pose graph, $\bar{\mathbf{E}}$ is the concatenated vector consisting of all $\bar{\mathbf{E}}'_{j,k}$ and $\mathbf{C}$ is a block-diagonal matrix comprised of $\mathbf{C}^{-1}_{j,k}$ as submatrices. Minimizing this function yields new optimal pose estimates. The minimization of $\mathbf{W}$ is accomplished via the following linear equation system:

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

The matrix $\mathbf{B}$ consists of the submatrices

$$\mathbf{B}_{j,k} = \begin{cases}\displaystyle\sum_{k=0}^{n}\mathbf{C}^{-1}_{j,k} & (j = k)\\ \mathbf{C}^{-1}_{j,k} & (j \neq k).\end{cases}$$

The entries of $\mathbf{A}$ are given by:

$$\mathbf{A}_j = \sum_{\substack{k=0\\k\neq j}}^{n}\mathbf{C}^{-1}_{j,k}\bar{\mathbf{E}}_{j,k}.$$

In addition to $\mathbf{X}$, the associated covariance of $\mathbf{C}_X$ is computed as follows:

$$\mathbf{C}_X = \mathbf{B}^{-1}$$

Note that the results have to be transformed in order to obtain the optimal pose estimates.

$$\begin{aligned}\mathbf{X}_j &= \bar{\mathbf{X}}_j - \mathbf{H}_j^{-1}\mathbf{X}'_j,\\ \mathbf{C}_j &= (\mathbf{H}_j^{-1})\mathbf{C}^X_j(\mathbf{H}_j^{-1})^T.\end{aligned}$$

1) Linearization of Euler Angles: The representation of pose $\mathbf{X}$ in Euler angles, as well as its estimate and error is as follows:

$$\mathbf{X} = \begin{pmatrix}t_x\\t_y\\t_z\\\theta_x\\\theta_y\\\theta_z\end{pmatrix}, \bar{\mathbf{X}} = \begin{pmatrix}\bar{t}_x\\\bar{t}_y\\\bar{t}_z\\\bar{\theta}_x\\\bar{\theta}_y\\\bar{\theta}_z\end{pmatrix}, \Delta\mathbf{X} = \begin{pmatrix}\Delta t_x\\\Delta t_y\\\Delta t_z\\\Delta\theta_x\\\Delta\theta_y\\\Delta\theta_z\end{pmatrix}$$

The matrix decomposition $\mathbf{M}_i\mathbf{H} = \nabla\mathbf{Z}_i(\bar{\mathbf{X}})$ is given by (9) and

$$\mathbf{M}_i = \begin{pmatrix}1 & 0 & 0 & 0 & -d_{y,i} & -d_{z,i}\\ 0 & 1 & 0 & d_{z,i} & d_{x,i} & 0\\ 0 & 0 & 1 & -d_{y,i} & 0 & d_{x,i}\end{pmatrix}.$$

As required, $\mathbf{M}_i$ contains all point information while $\mathbf{H}$ expresses the pose information. Thus, this matrix decomposition constitutes a pose linearization similar to those proposed in the preceding sections. Note that, while the matrix decomposition is arbitrary with respect to the column and row ordering of $\mathbf{H}$, this particular description was chosen due to its similarity to the 3D pose solution given in [19].

2) Linearization of Quaternions: The representation of the pose $\mathbf{X}$ as quaternions, as well as its estimate and error are given as follows:

$$\mathbf{X} = \begin{pmatrix}t_x\\t_y\\t_z\\p\\q\\r\\s\end{pmatrix}, \bar{\mathbf{X}} = \begin{pmatrix}\bar{t}_x\\\bar{t}_y\\\bar{t}_z\\\bar{p}\\\bar{q}\\\bar{r}\\\bar{s}\end{pmatrix}, \Delta\mathbf{X} = \begin{pmatrix}\Delta t_x\\\Delta t_y\\\Delta t_z\\\Delta p\\\Delta q\\\Delta r\\\Delta s\end{pmatrix}$$

The matrix decomposition $\mathbf{M}_i\mathbf{H} = \nabla\mathbf{Z}_i(\bar{\mathbf{X}})$ for quaternions is given by:

$$\begin{aligned}\mathbf{M}_i &= \begin{pmatrix}1 & 0 & 0 & d_{x,i} & 0 & -d_{z,i} & d_{y,i}\\ 0 & 1 & 0 & d_{y,i} & d_{z,i} & 0 & -d_{x,i}\\ 0 & 0 & 1 & d_{z,i} & -d_{y,i} & d_{x,i} & 0\end{pmatrix}\\ \mathbf{H} &= \begin{pmatrix}\mathbf{I}_{3\times3} & -2\cdot\mathbf{T}\\ 0 & 2\cdot\mathbf{U}\end{pmatrix}\\ \mathbf{T}^T &= \begin{pmatrix}\bar{p}\bar{t}_x + \bar{s}\bar{t}_y - \bar{r}\bar{t}_z & -\bar{s}\bar{t}_x + \bar{p}\bar{t}_y + \bar{q}\bar{t}_z & \bar{r}\bar{t}_x - \bar{q}\bar{t}_y + \bar{p}\bar{t}_z\\ \bar{q}\bar{t}_x + \bar{r}\bar{t}_y + \bar{s}\bar{t}_z & -\bar{r}\bar{t}_x + \bar{q}\bar{t}_y - \bar{p}\bar{t}_z & -\bar{s}\bar{t}_x + \bar{p}\bar{t}_y + \bar{q}\bar{t}_z\\ \bar{r}\bar{t}_x - \bar{q}\bar{t}_y + \bar{p}\bar{t}_z & \bar{q}\bar{t}_x + \bar{r}\bar{t}_y + \bar{s}\bar{t}_z & -\bar{p}\bar{t}_x - \bar{s}\bar{t}_y + \bar{r}\bar{t}_z\\ \bar{s}\bar{t}_x - \bar{p}\bar{t}_y - \bar{q}\bar{t}_z & \bar{p}\bar{t}_x + \bar{s}\bar{t}_y - \bar{r}\bar{t}_z & \bar{q}\bar{t}_x + \bar{r}\bar{t}_y - \bar{s}\bar{t}_z\end{pmatrix}\\ \mathbf{U} &= \begin{pmatrix}\bar{p} & \bar{q} & \bar{r} & \bar{s}\\ \bar{q} & -\bar{p} & \bar{s} & -\bar{r}\\ \bar{r} & -\bar{s} & -\bar{p} & \bar{q}\\ \bar{s} & \bar{r} & -\bar{q} & -\bar{p}\end{pmatrix}\end{aligned}$$

Again, all point information is restricted to $\mathbf{M}_i$ and all pose information to $\mathbf{H}$, so that the matrices describe a linearization of the quaternion transformation. Since this specific linearization was derived without regarding the normalized nature of the unit quaternions, this approach may yield non-unit quaternions. As non-unit quaternions do not constitute

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & \bar{t}_z \cos(\bar{\theta}_x) + \bar{t}_y \sin(\bar{\theta}_x) & \bar{t}_y \cos(\bar{\theta}_x)\cos(\bar{\theta}_y) - \bar{t}_z \cos(\bar{\theta}_y)\sin(\bar{\theta}_x) \\ 0 & 1 & 0 & -\bar{t}_z & -\bar{t}_x \sin(\bar{\theta}_x) & -\bar{t}_x \cos(\bar{\theta}_x)\cos(\bar{\theta}_y) - \bar{t}_z \sin(\bar{\theta}_y) \\ 0 & 0 & 1 & \bar{t}_y & -\bar{t}_x \cos(\bar{\theta}_x) & \bar{t}_x \cos(\bar{\theta}_y)\sin(\bar{\theta}_x) + \bar{t}_y \sin(\bar{\theta}_y) \\ 0 & 0 & 0 & 1 & 0 & \sin(\bar{\theta}_y) \\ 0 & 0 & 0 & 0 & \sin(\bar{\theta}_x) & \cos(\bar{\theta}_x)\cos(\bar{\theta}_y) \\ 0 & 0 & 0 & 0 & \cos(\bar{\theta}_x) & -\cos(\bar{\theta}_y)\sin(\bar{\theta}_x) \end{pmatrix} \tag{9}$$

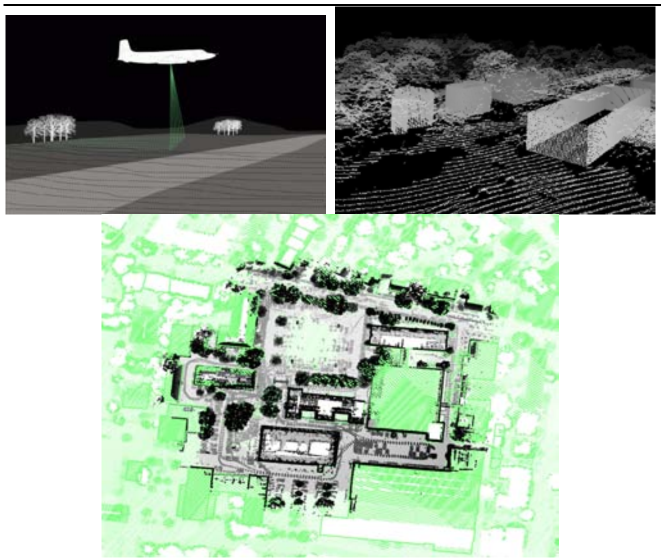

Fig. 5. Top left: Schema of the airborne based acquisition of reference data. Top right: 3D map consisting of aerial laser data and extrapolated 2D reference data. Bottom: Airborne and 3D map (green) with superimposed 3D scans (grey).

valid poses, the resulting quaternion must additionally be normalized before use.

IV. Experiments and Results

The experiments have been made using data of the Robotic 3D Scan Repository [3]. Implementations of the four methods can be found in [4]. The data set Hannover2 (cf. Fig. 1) has been acquired in an urban area and contains 922 3D scans each containing up to 35000 3D data points. It was acquired by a robot carrying a continuously rotating 3D scanner [26]. In this manner, hundreds of 3D scans can be acquired and globally consistent scan matching can be studied. To the data set Hannover2 ground truth data in form of a 2D map provided by the German land registry office (Katasteramt) is available. This map contains the buildings with a precision of 1 cm. In addition, we obtained airborne based 3D data. Based on this data so-called reference data is generated as follows (see Fig. 5): The 2D map is extrapolated to 3D by vertical 3D points and fused with the 3D data from the airplane. The result is a precise 3D reference map. Using this 3D reference map, we generate reference poses for all 922 3D laser scans by matching the scans with the reference map. To these poses we will refer to as “ground truth”.

We used the incremental ICP algorithm to match a sequence of 3D scans. Here we matched every 3D scan against its predecessor and take into account that errors sum up. After the robot returned with the scanner approximately to a known position a loop is closed. Then we applied the global relaxation and analyzed this error function minimization. Fig. 6 presents the results for two 3D scans (No. 80 and No. 150). Two loop closings are recorded, i.e., the first relaxation was up to iteration 150, the second one up to iteration 300. We see that all methods converge to stable different minima and except the quaternion algorithm, the final minimum is close to the ground truth.

Fig. 7 shows the precision of two 3D scans that reside in the second loop closing. Two methods, namely the small angle approximation and the uncertainty based quaternion solution show convergence to incorrect minima for scan No. 200, while for scan No. 384 only the quaternion solutions seems to be incorrect. Note that the error of the small angle approximation is only due to misalignment errors on the y-coordinate, i.e., the height of the scans.

The best performance in terms of accuracy is achieved by the helix transform and the uncertainty based optimization using Euler angles. The experiments show that global registration of many systems yield a complex and fragile optimization system. Small variances in the calculated matrices yield different closest point pairs in the following iteration, which in turn result in different matrices. Most stable results with respect to the final pose estimates of the 3D scans can be reported for the approximation of the error function using the helix transform and the uncertainty based optimization using Euler angles. The quaternion based approach needs a renormalization step to compute orthonormal rotation matrices. Thus, it is more likely to fail.

V. Conclusions and Future Work

This paper addressed the n-scan registration problem. Four methods that approximate a closed-form solution for a global optimal iterative closest point algorithm have been implemented and compared. Using the helix transform or the uncertainty based optimization using Euler angles yields more precise registration results than the other two methods. The principal problem that was addressed here is how to do optimization with quantities that should stay in the manifolds of rotations, i.e., in $SO(3)$. The helix transform gives a solution that lies in this space, i.e., there is a surjection between each helix transform and the rigid body transformation. For small angles this is not true since one can calculate new angles and the rigid body transformation founds does not correspond with the infinitesimal displacement calculated by the gradient descent step. However as a linearization method this method seems to work as well, since only small displacements are computed from the point pairs. We showed that the helix transform works the best. The uncertainty-based with Euler angles seems working similarly good, while not being identical to the helix transform. Since the

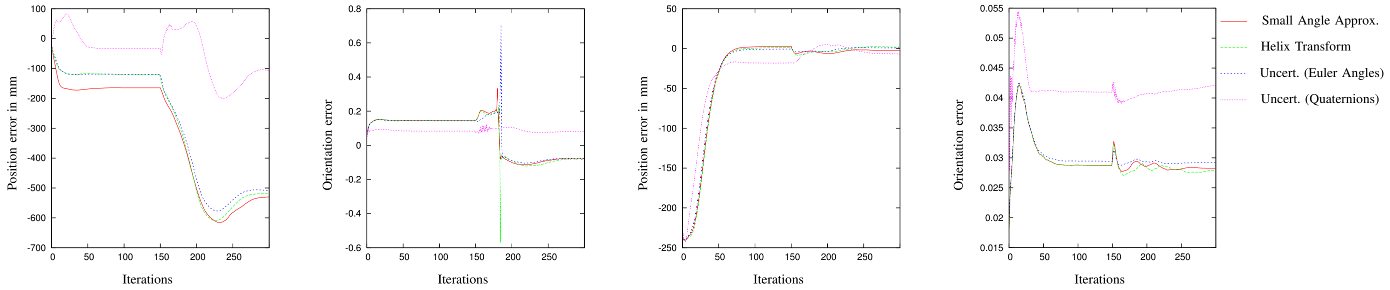

Fig. 6. Convergence of the registration of scan 80 (left) and 150 (right) using different minimization algorithms. The first 150 iterations correspond to the first relaxation (sequence A-B-C-D-A-B) while the second 150 iterations represent the result after registration of 384 3D scans (sequence: A-B-C-D-A-B-E-F-A).

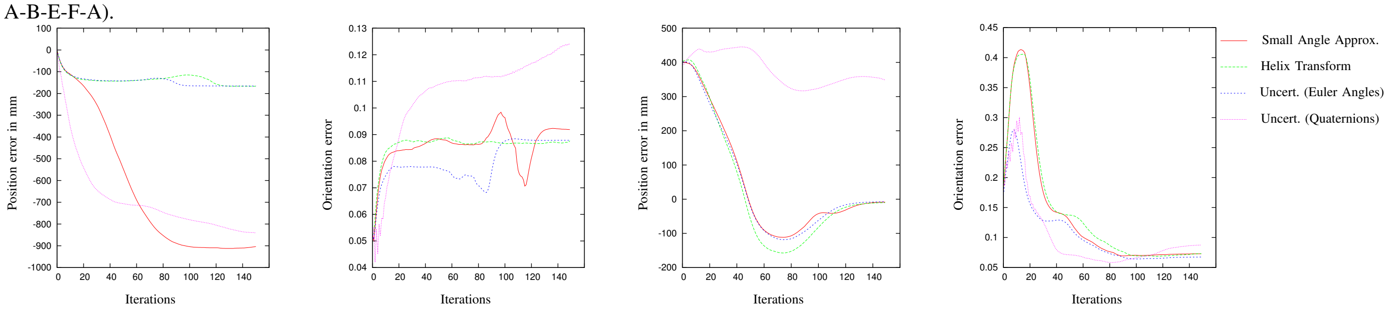

Fig. 7. Convergence of the registration of scan 200 (left) and 384 (right) using different minimization algorithms. The 150 iterations correspond to the second relaxation (sequence: A-B-C-D-A-B-E-F-A).

quaternion-based method leaves $SO(3)$ it shows the poorest performance and we do not recommand using it.

Currently we are working on an uncertainty-based registration using the Helix transform. In future work we are aiming at finding a closed form solution to get rid of the linearization and the problems due to it. Furthermore, as future work we plan to apply the proposed algorithms to large scale experiments, i.e., to 3D mapping of cities. Then, the back-end, i.e., solving the system of linear equations becomes important.

Acknowledgements

We would like to thank Dorit Borrman (Jacobs University Bremen), Kai Lingemann and Joachim Hertzberg (both University of Osnabrück) for supporting our work. Furthermore, the authors thank Oliver Wulf and Bernardo Wagner (both Leibniz University of Hannover, Germany) for making the 3D data set HANNOVER2 publicly available and Claus Brenner (Leibniz University of Hannover) for letting us use the airborne 3D data.

References

[1] MESA Imaging, `http://www.mesa-imaging.ch/`, 2009.

[2] PMDTechnologies GmbH, `http://www.pmdtec.com/`, 2009.

[3] Robotic 3D Scan Repository. `http://kos.informatik.uni-osnabrueck.de/3Dscans/`, 2009.

[4] 6DSLAM. `http://slam6d.sourceforge.net/`, 2009.

[5] R. Benjemaa and F. Schmitt. Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique. In *Proc. IEEE 3DIM*, Ottawa, Canada, May 1997.

[6] R. Benjemaas and F. Schmitt. A Solution For The Registration Of Multiple 3D Point Sets Using Unit Quaternions. *Computer Vision – ECCV '98*, 2:34 – 50, 1998.

[7] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Transactions on PAMI*, 18(5):540 – 547, May 1996.

[8] P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on PAMI*, 14(2):239 – 256, February 1992.

[9] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally consistent 3d mapping with scan matching. *J. of Robotics and Autonomous Systems*, 56(2):130–142, February 2008.

[10] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. The Efficient Extension of Globally Consistent Scan Matching to 6 DOF. In *Proc. 3DPVT*, Atlanta, GA, USA, June 2008.

[11] Y. Chen and G. Medioni. Object Modelling by Registration of Multiple Range Images. In *Proc. IEEE ICRA*, Sacramento, USA, 1991.

[12] Y. Chen and G. Medioni. Object Modelling by Registration of Multiple Range Images. *Image Vision Comput.*, 10(3):145 – 155, 1992.

[13] S. Cunnington and A. Stoddart. N-View Point Set Registration: A Comparison. In *Proc. BMVC*, Nottingham, UK., 1999.

[14] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient Estimation of Accurate Maximum Likelihood Maps in 3D. In *Proc. IEEE/RSJ IROS*, San Diego, CA, USA, 2007.

[15] M. Hofer. Personal communication, 7. July 2008.

[16] M. Hofer and H. Pottmann. Orientierung von Laserscanner-Punktwolken. *Vermessung & Geoinformation*, 91:297 – 306, 2003.

[17] 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 Symp. on Geometry Processing*, 2000.

[18] A. Lorusso, D. Eggert, and R. Fisher. A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations. In *Proc. BMVC*, Birmingham, UK., September 1995.

[19] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. In *Autonomous Robots*, vol. 4, 1997.

[20] H. Pottmann, S. Leopoldseder, and M. Hofer. Simultaneous registration of multiple views of a 3D object. *ISPRS Archives*, 34(3A):265 – 270, 2002.

[21] K. Pulli. Multiview Registration for Large Data Sets. In *Proc. IEEE 3DIM*, Ottawa, Canada, 1999.

[22] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. IEEE 3DIM*, Quebec City, Canada, May 2001.

[23] A. Stoddart and A. Hilton. Registration of multiple point sets. In *Proceedings of the 13th IAPR International Conference on Pattern Recognition*, pages 40–44, Vienna, Austria, August 1996.

[24] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, vol. 1883 of *Lecture Notes in Computer Science*, Springer-Verlag, 2000.

[25] J. Williams and M. Bennamoun. Multiple View 3D Registration using Statistical Error Models. In *Vision Modeling and Visualization*, 1999.

[26] O. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner. Benchmarking urban six-degree-of-freedom simultaneous localization and mapping. *Journal of Field Robotics (JFR)*, 25(3):148–163, March 2008.

[27] Z. Zhang. Iterative point matching for registration of free–form curves. Technical Report RR-1658, INRIA, France, 1992.