Closed-form solutions for estimating a rigid motion from plane correspondences extracted from point clouds

Kourosh Khoshelham
Department of Infrastructure Engineering, University of Melbourne, Australia

ABSTRACT

Registration is often a prerequisite step in processing point clouds. While planar surfaces are suitable features for registration, most of the existing plane-based registration methods rely on iterative solutions for the estimation of transformation parameters from plane correspondences. This paper presents a new closed-form solution for the estimation of a rigid motion from a set of point–plane correspondences. The role of normalization is investigated and its importance for accurate plane fitting and plane-based registration is shown. The paper also presents a thorough evaluation of the closed-form solutions and compares their performance with the iterative solution in terms of accuracy, robustness, stability and efficiency. The results suggest that the closed-form solution based on point–plane correspondences should be the method of choice in point cloud registration as it is significantly faster than the iterative solution, and performs as well as or better than the iterative solution in most situations. The normalization of the point coordinates is also recommended as an essential preprocessing step for point cloud registration. An implementation of the closed-form solutions in MATLAB is available at: http://people.eng.unimelb.edu.au/kkhoshelham/research.html#directmotion.

1. Introduction

Mapping natural and built environments usually involves acquiring dense and detailed spatial data in the form of point clouds. To capture all facets of the objects of interest often multiple point clouds are needed. Hence, a prerequisite step in processing point clouds is to align two or more recordings and transform them into one common coordinate system. This process is called registration. While many approaches to registration exist, a typical approach in practice is to select (manually or automatically) a number of corresponding points from the two point clouds, and estimate a transformation between the point clouds by minimizing the Euclidean distance between the corresponding points.

Using point correspondences usually works best for roughly-aligned dense point clouds or when the correspondences are measured by markers. When the point density is low and the points are noisy, using point correspondences leads to an inaccurate registration. In such cases, using plane correspondences will result in a more accurate registration (Rusinkiewicz and Levoy, 2001). Plane fitting methods are less influenced by noise and low density of the points, and can be made robust to possible outliers. Plane-based registration is particularly suitable for point clouds of man-made objects, since such objects typically have polyhedral shapes and planar surfaces.

Plane-based registration of point clouds has been a topic of growing interest in recent years (Grant et al., 2012; Habib et al., 2010; Khoshelham, 2010; Lichti and Chow, 2013; Pathak et al., 2010; Van der Sande et al., 2010; Taguchi et al., 2012). It consists of three main steps: extraction of planes, establishing correspondence, and estimating a transformation using the corresponding plane pairs. The focus of this paper is the estimation step. Most of the existing approaches use iterative solutions for the estimation of transformation parameters from plane correspondences. Iterative solutions require an initial approximate estimate of the transformation parameters, and the iterations might converge slowly, converge to a local optimum or not converge at all. Also, since the estimation is often a necessary step for finding or updating the correspondences, an iterative solution will increase the computational cost of the whole registration process. The efficiency is particularly important when registering data acquired by range cameras and depth sensors, such as the Kinect, which
can capture point clouds at video frame rates (Khoshelham et al., 2013).

In contrast, closed-form solutions are independent of the initial approximation, and are more efficient as they don’t involve iteration. While several closed-form solutions based on point correspondences exist and have been evaluated (Eggert et al., 1997), a thorough study of closed-form solutions based on plane correspondences is not available. This paper has three main contributions:

- a new closed-form solution based on point–plane correspondences is proposed;
- the role of normalization in accurate plane fitting and estimation from planes is investigated;
- a thorough evaluation of the closed-form solutions is presented, including comparisons with the iterative solution in terms of accuracy, robustness, stability and efficiency.

The paper proceeds with a review of related research on point cloud registration in Section 2. The iterative and closed-form methods for transformation estimation based on plane correspondences are described in Section 3. Experimental evaluation of the methods is presented in Section 4. The paper concludes with a discussion in Section 5.

2. Related work

Early work on estimating a transformation from point correspondences goes back to studies of the absolute orientation problem in photogrammetry by Thompson (1958), and later by Schut (1960), Sansò (1973) and Horn (1987), who used different quaternion representations to derive linear equations for the estimation of rotation parameters. Estimating a rotation matrix from point observations has also been referred to as Procrustes problem (Hurley and Cattell, 1962) and Wahba’s problem (Wahba, 1965), for which solutions based on Singular Value Decomposition (SVD) have been proposed (Markley, 1988; Schönemann, 1966). Arun et al. (1987) developed a closed-form method using SVD to estimate a rigid motion from point correspondences. Horn et al. (1988) proposed a closed-form solution for the special case where point correspondences are co-planar. Eggert et al. (1997) provide a comparative evaluation of four closed-form solutions that use point correspondences. The popular Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992) makes use of Horn’s solution that is based on unit quaternions (Horn, 1987).

Chen and Medioni (1992) developed a different ICP algorithm, in which the transformation is estimated by minimizing the sum of squared orthogonal distances from the points in the source surface to the corresponding tangent planes in the destination surface. This point–plane metric has been shown to perform significantly better than the point–point metric in terms of accuracy and convergence rate (Rusinkiewicz and Levoy, 2001). However, unlike the point–point metric, the estimation based on the point–plane metric has been done in an iterative fashion, making the algorithm computationally more expensive. In the past decade, many works on point cloud registration have shown a preference for the point–plane metric (Grant et al., 2012; Gruen and Akca, 2005; Habib et al., 2010; Lichti and Chow, 2013; Mitra et al., 2004; Rabbani et al., 2007), though all these methods use iterative solutions. Olsson et al. (2006) developed a method for minimizing the point–plane metric, which gives a direct estimate of translation, but is iterative in estimating rotation. Van der Sande et al. (2010) proposed a closed-form solution for minimizing point–plane distances between overlapping point clouds acquired by an airborne Lidar mapping system. Their solution gives a 3D affine transformation rather than a rigid motion, with the undesired consequence that it can deform the point cloud. A more recent development is the set of minimal solutions of Ramalingam and Taguchi (2013) for estimating a rigid motion from several configurations of point–plane correspondences. These minimal solutions handle specific configurations of point–plane correspondences differently, and as such are not applicable to arbitrary numbers of points and planes.

Another approach to plane-based registration is by using plane–plane correspondences. In this approach, the error metric for minimization is the difference between the corresponding parameters of the corresponding planes. There are closed-form solutions for minimizing the plane–plane metric, which take advantage of the closed-form solutions for point correspondences. In these solutions, the normal vectors of the corresponding planes are treated as 3D points, enabling the application of point-based closed-form solutions. Gregor and Whitaker (2001) use the SVD approach of Arun et al. (1987) to estimate the rotation matrix from corresponding plane normals, whereas Brenner et al. (2008) and Pathak et al. (2010) use the quaternion method of Horn (1987). Although the closed-form plane–plane solutions have been shown to result in a satisfactory registration, they have not been thoroughly evaluated in terms of robustness to noise and improper plane configurations.

In summary, the literature suggests that a thorough study of closed-form solutions for estimating a rigid motion from plane correspondences is not available. In this paper, we propose a new closed-form solution based on singular value decomposition for estimating a rigid motion from point–plane correspondences, and evaluate the performance of the closed-form solutions in comparison to the iterative solution.

3. Estimation of a rigid motion from plane correspondences

3.1. Preliminaries

This section provides mathematical preliminaries on the transformation of points and planes in 3D space, the Kronecker product and the singular value decomposition. In the following mathematical expressions, matrices are denoted by bold uppercase letters, column vectors are denoted by bold lowercase letters and scalars are written in italics. Points and planes in 3D space are represented as column vectors.

The transformation between two point clouds typically consists of a 3D rotation $\mathbf{R}$ and a 3D translation $\mathbf{t}$. It is convenient to combine these two in a transformation matrix $\mathbf{H}$ of homogeneous coordinates:

$$
\mathbf{p}' = \mathbf{H} \mathbf{p} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{p}
$$

(1)

where the superscripts $s$ and $d$ denote respectively the source and destination coordinate system, and $\mathbf{p} = (x, y, z)^T$ is the homogeneous representation of a point in 3D space. The transformation $\mathbf{H}$ is usually referred to as a rigid motion. For a set of points that lie on a plane, the plane equation can be written as:

$$
\mathbf{n}^T \mathbf{p} = 0
$$

(2)

where $\mathbf{n} = (n_x, n_y, n_z)^T$ is the homogeneous representation of the plane defined by a normal vector $\mathbf{n} = (n_x, n_y, n_z)^T$ of unit length and its perpendicular distance $\rho$ from the origin. From (1) and (2), it can be seen that if $\mathbf{H}$ transforms a set of co-planar points from a source coordinate system $s$ to a destination coordinate system $d$, the corresponding transformation for the planes is (Khoshelham and Gorte, 2009):

$$
\mathbf{p}' = \mathbf{H}^T \mathbf{p}
$$

(3)

where $\mathbf{H}^T$ denotes the inverse transpose of $\mathbf{H}$. 

The estimation of the transformation $H$ from a set of plane correspondences can be based on minimizing the plane–plane metric (difference between the corresponding parameters of corresponding planes) or the point–plane metric (distance between the points and their corresponding planes). To derive the observation equation for minimizing the point–plane metric (Section 3.4) we make use of the properties of the Kronecker product. The Kronecker product of a $m \times n$ matrix $B$ and a $p \times q$ matrix $C$ is defined as (Neudecker, 1969):

$$B \otimes C = \begin{bmatrix} b_{11}C & \cdots & b_{1n}C \\ \vdots & & \vdots \\ b_{m1}C & \cdots & b_{mn}C \end{bmatrix}$$  \hspace{1cm} (4)

which is a $mp \times nq$ block matrix. A useful property of the Kronecker product, which we will use to simplify matrix equations, is the following:

$$\text{vec}(BCD) = (D^T \otimes B) \text{vec}(C)$$  \hspace{1cm} (5)

where $\text{vec}$ is the vectorization operator that converts a matrix to a column vector by stacking the columns on top of one another. The proof for (5) can be found in Neudecker (1969).

The singular value decomposition of a $m \times n$ ($m \geq n$) matrix $A$ with real elements is defined as: $A = UDV^T$, where $U$ is a $m \times m$ matrix with orthonormal columns, $V$ is a $n \times n$ orthogonal matrix and $D$ is a $m \times n$ diagonal matrix containing the singular values on its diagonal. A property of the singular value decomposition is that the columns of $V$ are the eigenvectors of $A^TA$ and the squared singular values in $D$ are the eigenvalues of $A^TA$. This property is useful in plane fitting, where the plane normal is typically estimated as the eigenvector corresponding to the smallest eigenvalue of the covariance of the points (Rabbani, Shah, 2006). Let $P$ ($n \times 3$) be a matrix containing a set of $n$ 3D points. The singular value decomposition of $(P - p)$, where $p$ is the mean of the points and $-$ here denotes subtracting the mean from each point in $P$, is:

$$P - p = UDV^T.$$  \hspace{1cm} (6)

The singular value decomposition is also useful for decomposing an affine transformation into rotation and scale matrices. Let $H$ $(3 \times 3)$ be a 3D affine transformation. The singular value decomposition $H = UDV^T$ yields two orthogonal $(3 \times 3)$ matrices $U$ and $V$, which can be seen as two rotation matrices, and a diagonal matrix $D$ $(3 \times 3)$, which can be interpreted as a scale matrix.

3.2. The iterative solution

The common approach to estimating a rigid motion from plane correspondences is to estimate the transformation parameters (three Euler angles and three translations) by minimizing the sum of squared distances between the points in the source coordinate system $s$ and their corresponding planes in the destination coordinate system $d$ (Chen and Medioni, 1992; Rusinkiewicz and Levoy, 2001). For a point $i$ and its corresponding plane $j$ the perpendicular distance can be written as:

$$\nu_i = \pi_j^T H \hat{p}_i^j$$  \hspace{1cm} (7)

where $\hat{p}$ and $\pi$ denote the homogeneous point and the plane respectively, and $H$ is the transformation. Using the Euclidean representation of the point and the components of $\pi$ and $H$ as defined in Section 3.1 we can write:

$$\begin{align*}
\pi_j^T (R \hat{p}_i^j + t) &= \rho_j^D + \nu_i \\
\text{Note that here } \hat{p}_i^j \text{ contains the Cartesian coordinates of the point. Eq. (7) basically expresses that by applying } R \text{ and } t \text{ to a point in the source coordinate system, it will be transformed to lie on its corresponding plane in the destination coordinate system, such that there is only a residual distance } \nu_i \text{ between the transformed point and the plane due to noise. The left-hand side of (7) is a non-linear function of the unknown transformation parameters. It can be approximated by the Taylor expansion around an initial estimate of the unknowns: }
\end{align*}$$

$$\begin{align*}
n_i^D (R \hat{p}_i^j + t) &= f(u) = f(u^0) + \sum_{k=1}^{6} \frac{\partial f}{\partial u_k} |_{u_k = u_0^k} (u_k - u_0^k) \\
\text{where } f(u^0) \text{ is obtained by evaluating the left-hand side of the equation with the initial values of the six unknown transformation parameters } u_0^k, \text{ and } (u_k - u_0^k) \text{ are corrections to the value of the unknowns. With a minimum of six point–plane correspondences in a general configuration a system of equations } A \Delta u = b + v \text{ is formed, where } A \text{ is the coefficient matrix containing the partial derivatives in (8), } b \text{ is the vector of constants containing } p_j^D - f(u^0) \text{ for all point–plane correspondences, } \Delta u \text{ is the vector of unknown corrections and } v \text{ is the vector of residual point–plane distances. The least-squares estimate for the unknown corrections is then given by: } \Delta u = (A^T W A)^{-1} A^T W b, \text{ where } W \text{ is a weight matrix that can be used to assign weights to point–plane correspondences. The least–squares estimate given above minimizes the squared norm } v^T W v \text{ of weighted residual point–plane distances. The initial estimate of the transformation parameters is then iteratively modified by calculating the corrections until some convergence criteria are satisfied. Typically, the iterations are terminated when the norm of residuals and/or the corrections to the unknowns become smaller than a predefined threshold.}
\end{align*}$$

3.3. Closed-form solution based on plane–plane correspondences

The transformation between two point clouds can also be estimated by minimizing the differences between the corresponding parameters of corresponding planes. For a plane $j$, (3) can be rewritten as:

$$\begin{align*}
n_i^j H &= n_i^j \\
\text{which can be further decomposed to:}
\end{align*}$$

$$\begin{align*}
\begin{bmatrix} n_i^j - \rho_j^i \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} &= \begin{bmatrix} n_i^j - \rho_j^i \end{bmatrix}
\end{align*}$$  \hspace{1cm} (10)

This provides two separate sets of equations per plane correspondence, one for $R$ and one for $t$:

$$\begin{align*}
n_i^j R &= n_i^j \\
n_i^j t &= \rho_j^i - \rho_j^i \\
\text{Note that both equations hold for exact planes, and when the plane parameters are noisy each equation should be augmented with a variable for residual error. With exactly three corresponding plane pairs in a general configuration the twelve equations derived from those correspondences can be directly solved for } R \text{ and } t \text{ respectively. For } m > 3 \text{ pairs of corresponding planes, the equations in (11) and (12) can be extended in the following forms:}
\end{align*}$$

$$\begin{align*}
N^T R &= N \\
N^T t &= d
\end{align*}$$  \hspace{1cm} (13)
(14)
where \( \mathbf{N}^i / (m \times 3) \), \( \mathbf{N}^j / (m \times 3) \) and \( \mathbf{d} / (m \times 1) \) are obtained by stacking \( \mathbf{n}_i^\mathbf{e} \), \( \mathbf{n}_j^\mathbf{e} \) and \( \mathbf{p}_i^\mathbf{e} - \mathbf{p}_j^\mathbf{e} \) respectively for all \( m \) plane pairs. The least-squares solution for \( t \) is then obtained as: 

\[
\mathbf{t} = (\mathbf{N}^i \mathbf{W}^j)\mathbf{W}^j / \mathbf{d}^\mathbf{e}
\]

where \( \mathbf{W} \) is a \((m \times m)\) weight matrix that can be used to assign weights to the plane pairs. Similarly, the solution for \( \mathbf{R} \) is:

\[
\mathbf{R} = (\mathbf{N}^i \mathbf{W}^j)\mathbf{W}^j / \mathbf{d}^\mathbf{e}
\]

where \( \mathbf{W} \) is a \((m \times m)\) weight matrix that can be used to assign weights to the plane pairs. Similarly, the solution for \( \mathbf{R} \) is:

\[
\mathbf{R} = (\mathbf{N}^i \mathbf{W}^j)\mathbf{W}^j / \mathbf{d}^\mathbf{e}
\]

However, when the observations are noisy this estimated rotation will not necessarily be an orthogonal matrix. An orthogonal estimate of the rotation matrix can be obtained by singular value decomposition of \( \mathbf{R} \) as:

\[
\mathbf{R} = \mathbf{U} \mathbf{D} \mathbf{V}^T
\]

where \( \mathbf{U} \), \( \mathbf{D} \) and \( \mathbf{V} \) are the orthogonal rotation matrix derived as: \( \mathbf{R} = \mathbf{U} \mathbf{V}^T \).

The last three elements of \( \mathbf{x} \) are the estimated translation parameters. This estimate is dependent on the estimation of \( \mathbf{R} \); thus, after enforcing the orthogonality, the translation vector should be re-estimated. From (7) we have:

\[
\mathbf{n}_i^j / \mathbf{t} = \mathbf{p}_i^\mathbf{e} - \mathbf{p}_j^\mathbf{e} / \mathbf{R}_i^j \mathbf{v} + \mathbf{v}_j
\]

By substituting the orthogonal \( \mathbf{R} \) in (17) and writing the equation for all point–plane correspondences, we again obtain a system of equations \( \mathbf{A} \mathbf{t} = \mathbf{b} + \mathbf{v} \), which we can solve for a new estimate of \( \mathbf{t} \) as:

\[
\mathbf{t} = (\mathbf{A}^\mathbf{W}^\mathbf{A})^{-1} \mathbf{A}^\mathbf{W}^\mathbf{b}
\]

In the experiments we verify whether re-estimation of \( \mathbf{t} \) improves the accuracy of the estimated translation parameters.

Although for registering point clouds the transformation is typically a rigid motion, which does not include scale, for other applications it will be useful to estimate scale parameters. In Appendix A, an extension of the closed-form solution based on point–plane correspondences for estimating a scale matrix is presented.

### 3.5. Normalizing point coordinates

An important issue in solving a system of linear equations is the condition of the coefficient matrix \( \mathbf{A} \), or more precisely of \( \mathbf{A}^\mathbf{A}^\mathbf{T} \). In the solutions that minimize the weighted sum of squared point–plane distances, some of the elements of the coefficient matrix are components of the normal vectors, which are bound between \(-1 \) and \(+1 \), while some are point coordinates, which can be very large. If the point coordinates are very large, e.g. when the point clouds are defined in a map projection coordinate system as in airborne laser scanning, the coefficient matrix will become poorly conditioned resulting in an inaccurate estimation of the transformation parameters. A measure of the condition of a system of linear equations is the condition number of the coefficient matrix, defined as:

\[
k(\mathbf{A}) = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}
\]

where \( \lambda_{\text{max}} \), \( \lambda_{\text{min}} \) are respectively the largest and smallest singular value of \( \mathbf{A}^\mathbf{A}^\mathbf{T} \). The condition number of a perfectly conditioned coefficient matrix is 1, whereas for a poorly conditioned coefficient matrix the condition number will be very large.

To avoid a poorly conditioned coefficient matrix when estimating a transformation, it is generally recommended that the point coordinates are normalized before the estimation (Hartley, 1997). Estimation based on normalized coordinates consists of three steps: (i) transforming the points in the source and destination coordinate system into normalized coordinates; (ii) estimating the rigid motion between the planes defined by the normalized points; (iii) retrieving the rigid motion between the original points from the rigid motion estimated from the normalized points. Let \( \mathbf{p}_i^s = \mathbf{H} \mathbf{p}_i \) denote a rigid motion from the source points to the destination points. If we normalize the points by applying a transformation \( \mathbf{T} \) to the source points and a transformation \( \mathbf{T}^d \) to the destination points, and then estimate a rigid motion \( \mathbf{H} \) using planes defined by the normalized points, we can write:

\[
\mathbf{T}^d / \mathbf{p}_i^s = \mathbf{H} \mathbf{H}^\mathbf{T} / \mathbf{p}_i^s
\]

Substituting \( \mathbf{p}_i^s \) with \( \mathbf{H} \mathbf{p}_i \) from the previous equation, we can retrieve \( \mathbf{H} \) from \( \mathbf{H} \) as:

\[
\mathbf{H} = \mathbf{T}^d / \mathbf{H} \mathbf{T}
\]

Typically, \( \mathbf{T} \) and \( \mathbf{T}^d \) contain a translation of the respective point cloud such that its geometric center is at the origin of the coordinate system, and a scaling such that the average point is a distance \( \sqrt{3} \) from the origin (Hartley and Saxena, 1997). However, scaling has two problems: (i) introducing different scales in \( \mathbf{T} \) and \( \mathbf{T}^d \) will make the rotation matrix in \( \mathbf{H} \) non-orthogonal, and (ii) intuitively
we expect that planes in a larger point cloud give a more accurate estimate of rotation, because a small change in the rotation results in larger point–plane distances when the planes are further away. Thus, the idea that shrinking a point cloud to have smaller coordinates will improve the estimation accuracy sounds counter-intuitive. Let us hypothesize that applying the same isotropic scaling (i.e. equally scaling each coordinate) in the source and destination coordinate system will indeed lead to improved estimation of the transformation parameters. Applying the same isotropic scaling is convenient for two reasons. First, in practice point clouds acquired by different sensors often have the same isotropic scale; second, applying the same scale in the source and destination coordinate systems will not affect the orthogonality of the rotation matrix in H. Therefore, we define the normalization transformations as:

\[
\begin{align*}
T' &= s \cdot \begin{bmatrix} 1 & -P_x' \\ 0 & 1 \end{bmatrix}, \\
T'' &= s \cdot \begin{bmatrix} 1 & -P_y'' \\ 0 & 1 \end{bmatrix}
\end{align*}
\]

(20)

where \( P_x' \) and \( P_y'' \) are the mean of the source and destination points respectively, \( s \) is the scale factor and \( I_3 \) is a 3 \( \times \) 3 identity matrix. By substituting the transformations of (20) in (19) we can cancel \( s \) and obtain:

\[
\begin{bmatrix} I_3 & -P' \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R \quad t \\ 0 \quad 1 \end{bmatrix} = \begin{bmatrix} R' \quad t' \\ 0 \quad 1 \end{bmatrix} \begin{bmatrix} I_3 & -P'' \\ 0 & 1 \end{bmatrix}
\]

(21)

Multiplying the matrices and simplifying the equations yields:

\[
R = R' \\
t = t' + P'' - R' P'
\]

The first equation is an obvious result since neither mean-centering the points nor isotropic scaling changes the rotation between the two point clouds. The second equation describes how \( t \) can be retrieved from \( t' \) and \( R' \). In the experiments, we will verify whether normalization, and in particular scaling, improves the estimation of the transformation parameters.

### 3.6. Degenerate plane configurations

A prerequisite for the estimation of a rigid motion from plane correspondences is that the planes are in a general configuration. A general configuration of planes is one in which the normal vectors span the three dimensional Euclidean space. This requires that at least three normal vectors are linearly independent and form a basis for 3D space. The system of linear equations will then have a unique least-squares solution. A degenerate configuration arises when the plane normals are coplanar or collinear. An example degenerate configuration is a set of vertical walls, where the normal vectors all lie in the horizontal plane. This leads to a system of linear equations where the coefficient matrix is rank-deficient. Therefore, transformation parameters cannot be estimated from a set of corresponding planes in a degenerate configuration. Degenerate configurations can also be caused by points, e.g. when points are collinear. However, in practice point clouds contain large numbers of points, so degenerate point configurations are very rare.

The adequacy of the planes configuration can be measured as well by the condition number of the coefficient matrix. A set of planes in an ideal configuration (e.g. two perpendicular vertical planes and a horizontal plane) will lead to a coefficient matrix with a condition number close to 1, whereas for a set of planes in a degenerate configuration the condition number will be infinite.

### 4. Experiments

Experiments were carried out to evaluate the accuracy of the three estimation methods, and to analyze the effect of noise, scene scale, configuration of planes and redundancy on the accuracy of each solution. To verify the role of normalization of point coordinates, each experiment was carried out with and without normalized coordinates. In the following, the solutions are referred to as “iterative”, “SVD plane–plane” (closed-form based on plane–plane correspondences) and “SVD point–plane” (closed-form based on point–plane correspondences).

#### 4.1. Evaluation method

Evaluating the accuracy of the estimation methods requires knowledge of the true transformation parameters. Therefore, we design a simulation in which we randomly generate a set of true transformation parameters (three rotation angles and three translation parameters), apply these to transform a set of points and planes, and then use the source and transformed points and planes to estimate the transformation parameters. The difference between the true and the estimated transformation parameters is used to calculate measures of accuracy for the estimation methods. Fig. 1 illustrates this procedure. To analyze the effect of noise, scene scale (i.e. dimension of the point cloud), configuration of planes and redundancy (i.e. number of points and planes), we simulate a point cloud representing a unit cube of 1 m in dimensions, in which the points are initially noise-free and the true plane parameters are known. In each analysis, the relevant parameter of the data is changed and the remaining ones are kept fixed. Table 1 summarizes the parameters used for each analysis.

To obtain a statistically reliable measure of accuracy the estimation errors are calculated over 100 tests. For each test, three rotation angles are generated by random sampling from a uniform distribution within the range \([-90, +90]^\circ\), and three translation parameters by random sampling from a uniform distribution within the range \([-10, +10] \text{ m}\). This set of \(6 \times 100\) parameters is generated once and then used as the true transformation parameters in all experiments. To evaluate the accuracy of each solution three error measures are defined as follows:

- **RMS residual error** defined as the root mean square of the residual distances between the points and their corresponding planes after applying the estimated transformation, i.e.

\[
\text{RMS residual error} = \left[ \frac{1}{6} \sum_{i=1}^{6} \left( \hat{p}_i^T H \hat{p}_i \right)^{2} \right]^{\frac{1}{2}}.
\]

- **Rotation error** defined as the absolute difference between the true and estimated rotation. For convenience, we convert the three rotation angles to one Euler angle \( \theta = \cos^{-1}(r_{13} + r_{22} + r_{33} - 1)/2 \), where \( r_{11}, r_{22} \) and \( r_{33} \) are the diagonal elements of \( R \) (Eberly, 2008).

![Fig. 1. The evaluation method.](image)
Translation error defined as the absolute difference between the length of the true translation and that of the estimated translation.

The evaluation of the three solutions in all experiments is based on the mean of the above error measures over 100 tests.

In the experiments, the following settings are used in the iterative solution: the iterations are terminated when the residual point–plane distances or the corrections to the unknown parameters become smaller than $10^{-6}$. If this termination criterion is not met, the iterations are set to terminate after a maximum of 20 iterations. Furthermore, in all experiments the initial approximate value for all unknown parameters, i.e. the three rotation angles and the three translation parameters, is set to zero.

In the iterative solution and the SVD point–plane solution the source planes and transformed points are used for the estimation of the transformation parameters. In the SVD plane–plane solution, planes in the source and transformed coordinate system are used for the estimation of the transformation parameters. While the source planes have known parameters, the planes in the transformed cube are obtained by fitting planes to the transformed points using the method described in Section 3.1.

4.2. Accuracy with exact points and planes

Fig. 2 shows box plots of the base-10 logarithm of the error measures calculated for the three solutions with and without normalized coordinates. Each box summarizes the results of 100 tests with noise-free points and planes. It is clear that the error measures for the two closed-form solutions are close to the machine round-off error, whereas the error measures for the iterative solution are mostly close to the value set for the termination of the computation.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Parameter</th>
<th>Number of planes</th>
<th>Number of points per plane</th>
<th>Noise standard deviation (m)</th>
<th>Scale (dimension of point cloud) (m)</th>
<th>Condition number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td></td>
<td>6</td>
<td>100</td>
<td>0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Robustness to noise</td>
<td></td>
<td>6</td>
<td>100</td>
<td>0.001, 0.01, ..., 1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Effect of scene scale</td>
<td></td>
<td>6</td>
<td>100</td>
<td>0.01</td>
<td>1.0, 10, ..., 1.0e5</td>
<td>1.0</td>
</tr>
<tr>
<td>Effect of improper plane configurations</td>
<td></td>
<td>5</td>
<td>100</td>
<td>0.01</td>
<td>1.0–2.0</td>
<td>1.0, ..., 1.0e4</td>
</tr>
<tr>
<td>Effect of number of points</td>
<td></td>
<td>6</td>
<td>4, 10, ..., 5000</td>
<td>0.01</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Computation time</td>
<td></td>
<td>6, 10, ..., 50</td>
<td>100</td>
<td>0.01</td>
<td>1.0</td>
<td>1.0–3.9</td>
</tr>
</tbody>
</table>

Table 1: Parameters of the data used in the experiments.

Fig. 2. Comparison of the accuracy of the three solutions in terms of RMS residual error (a), rotation error (b) and translation error (c) with and without normalization.
the iterations. Normalization results in a significant improvement only in the translation error of the iterative method; however, the overall accuracy of the iterative method, with and without normalization, is bounded by the threshold set for terminating the iterations.

Another interesting observation is that in all tests the iterative solution seems to have converged from the initial value of zero to a correct estimate independent of the true value of the transformation parameters. To verify that the iterations have converged in all tests the histogram of the number of iterations in 100 tests is shown in Fig. 3. While in all tests the solution converges within no more than 9 iterations, in majority of the tests the convergence is reached after 4 or 5 iterations. The normalization did not have any impact on the number of iterations.

4.3. Robustness to noise

To analyze the effect of noise on the accuracy of the estimation methods, different levels of noise were added to the points in the source unit cube, and after transforming the unit cube with randomly generated transformations, the transformation parameters were estimated from the noisy points and planes. Before comparing the three estimation methods, recall that in the SVD point–plane solution we suggested that when the points are noisy one has to enforce orthogonality on \( R \) and subsequently re-estimate \( t \). To test whether this improves the estimate of \( t \), we first compare the estimation error of the SVD point–plane method against different noise levels with and without the re-estimation of \( t \). The log–log plots in Fig. 4 show the result. The horizontal axes show the standard deviation of Gaussian noise added to coordinates of all transformed points. All estimates were obtained without normalization. From the translation error it can be seen that the re-estimated \( t \) is about one order of magnitude more accurate that the initially estimated \( t \). The lower error of the re-estimated \( t \) also contributes to lower RMS residual error for all noise levels. These results confirm that in presence of noise the re-estimation of \( t \) in the SVD point–plane method is indeed necessary. In the following experiments, all results of SVD point–plane solution include the re-estimation of \( t \).

Fig. 5 compares the effect of increasing noise levels on the accuracy of the three solutions without normalization. While the rotation error measure shows a similar linear increase for all three solutions, the translation errors are significantly larger for the SVD plane–plane solution across all noise levels. The larger translation errors result also in slightly larger RMS residual errors for the SVD plane–plane solution.

To understand why the SVD plane–plane solution gives less accurate estimates of the translation parameters in presence of noise, recall that the estimation of \( t \) in the plane–plane solution is dependent on the \( \rho \) parameter of the noisy planes (see Eq. (12)). Since \( \rho \) is calculated as \( \rho = n^T p \) in plane fitting (Section 3.1), its variance is dependent on the variance of the plane normal \( n \) and the magnitude of \( p \), that is the mean of the points. This means that the random error in \( n \) caused by the noise in the point coordinates can be projected to a large error in \( \rho \) depending on the magnitude of \( p \). Thus, the precision of plane fitting is not only dependent on the noise level of the points, but also on the choice of the coordinate system. This is an interesting observation with important implications in plane-based registration. In practice, the error of plane parameters due to the choice of the coordinate system deteriorates the result of the point–plane solutions as well, although to a lesser degree, since in the point–plane solutions the planes are defined only in one point cloud (compared to two in the plane–plane solution). A similar observation on the effect of the coordinate system on line fitting in images has been made in (Vosselman and Haralick, 1996). Appendix B provides the derivation of the variance of \( \rho \) in plane fitting.

The dependence of plane fitting precision on the choice of coordinate system suggests that normalizing the point coordinates should improve the estimate of \( \rho \) in plane fitting (by making \( p \) smaller), and consequently improve the estimate of the translation parameters in the SVD plane–plane solution. To verify this, the accuracy of the three solutions against increasing noise levels

---

**Fig. 3.** Histogram of number of iterations for the iterative method showing convergence in all 100 tests.

**Fig. 4.** RMS residual error (a) and translation error (b) of the SVD point–plane method against different noise levels with and without re-estimation of \( t \).
was evaluated this time with normalized coordinates. The result is shown in Fig. 6. Note that normalization involves only mean-centering as the cube dimension is unity. With normalized points and planes the three solutions have a similar performance in terms of all three error measures. In particular, the translation error of the SVD plane–plane solution is improved as a result of normalization. The SVD point–plane solution performs slightly better in estimating the rotation at large noise levels, but that does not result in better RMS residual error after the transformation. It is also interesting to note that the RMS residual errors are quite consistent with the standard deviation of the noise.

4.4. Effect of scene scale

In practice, point clouds represent scenes that are larger than the simulated unit cube. Planes defined in large point clouds are expected to give more accurate estimates of the rotation parameters. On the other hand, smaller point clouds, which may as well be obtained by normalization scaling, lead to solutions with better condition numbers and better estimation accuracies. To study the effect of scene scale on the three solutions, enlarged versions of the simulated point cloud were generated by applying a range of isotropic scale factors to the point coordinates. To each point cloud Gaussian noise of 0.01 m standard deviation was added, considering that in practice noise is usually independent of the spatial extent of the point cloud. The three solutions were then evaluated using the source and transformed enlarged point clouds with and without normalization. Fig. 7 shows the results obtained without normalization. The rotation error for the three solutions decreases linearly with increasing scale. This indicates that the accuracy of the rotation estimates is dominated by the spatial extent of the point cloud rather than by the condition of the solution. Unlike the rotation, the translation error and the RMS residual error for the iterative and SVD point–plane solution are not affected by scale, but for the SVD plane–plane solution the translation error, and consequently the RMS residual error, are larger at small scales, and improve with increasing scale. This might seem surprising, because from the previous experiment we learned that the translation error depends on the error of the \( q \) parameter of the planes, which is expected to be larger at larger scales (due to larger \( C_{22} \)). However, the error of \( q \) depends also on the error of plane normal (see Appendix B), which we expect to be smaller for larger planes.

Fig. 5. Effect of noise on the accuracy of the three solutions without normalization evaluated by RMS residual error (a), rotation error (b) and translation error (c).

Fig. 6. Effect of noise on the accuracy of the three solutions with normalized coordinates evaluated by RMS residual error (a), rotation error (b) and translation error (c).
To understand the effect of plane fitting errors on the accuracy of the SVD plane–plane solution, we compared at each scale the plane parameters obtained by plane fitting to noisy transformed points with those obtained by applying the transformation to noise-free source planes. The discrepancies were used to calculate two error measures as shown in Fig. 8, where $e_q$ is the standard deviation (over 100 tests) of discrepancies between the two estimates of $q$, and $e_n$ is the mean norm of the difference vector between the two estimates of the plane normal $n$. Note how the error of $n$ decreases linearly with increasing scale. The error of $q$ shows the combined effect of the error of $n$ and magnitude of $\rho$, $C^2_p$ (mean of the points). It decreases up to a scale of 10 and then remains constant. This explains the translation error for the SVD plane–plane solution in Fig. 7(c) and its effect on the RMS residual error in Fig. 7(a).

Fig. 7. Effect of scale on the accuracy of the three solutions without normalization evaluated by RMS residual error (a), rotation error (b) and translation error (c).

Fig. 8. Effect of scale on the precision of plane fitting. While the error of plane normal, $e_n$, decreases with increasing scale, the error of $\rho$, $e_{\rho}$, shows the combined effect of $e_n$ and the magnitude of $p$ (mean of the points).

To understand the effect of improper plane configurations, opened versions of the simulated unit cube were generated by tilting the vertical planes to various degrees as shown in Fig. 10. The adequacy of the configuration of planes in each opened cube was measured by calculating the condition number of the matrix of plane normals. Table 2 shows the condition numbers corresponding to different tilt angles of the opened cubes, where a tilt of 89° represents an almost degenerate configuration of planes. The point cloud of each opened cube was transformed using the known randomly generated transformations, and to each transformed point cloud Gaussian noise of 0.01 m standard deviation was added. The three solutions were then evaluated using the source and transformed point clouds with and without normalization.

Fig. 9 shows the effect of scale on the performance of the three solutions this time with normalized coordinates. The translation errors are now quite similar for the three solutions, and are slightly lower than those obtained without normalization. The rotation errors, however, are curious. Surprisingly, for the closed-form solutions the rotation errors still decrease with increasing scale. This is due to the fact that normalization suppresses the noise level while shrinking the point cloud. For the iterative solution, the rotation error decreases up to a scale factor of 200, but then increases and remains constant at about 0.001°. Examining the iterations revealed that before this turning point the iteration termination is ruled by the magnitude of the corrections to the estimated rotation, but after that the iterations terminate by the mean point–plane residual distance condition. This is because at large scales the noise level is more suppressed by normalization. As a result, the point–plane residuals become smaller than the $1e^{-6}$ threshold after a few iterations, and the iteration stops even though the rotation estimates are not very precise. In practice, this situation can be mitigated by modifying the criteria for stopping the iterations, though at the price of increasing the number of iterations. In Fig. 9(a), it is also interesting to note that since the RMS residual error for the iterative solution is calculated based on the retrieved transformation parameters, it grows quite large at larger scales. In contrast, for the closed-form solutions the RMS residual error is not affected by scale, and is consistent with the standard deviation of the added noise.

4.5. Stability against different plane configurations

To analyze the effect of improper plane configurations, opened versions of the simulated unit cube were generated by tilting the vertical planes to various degrees as shown in Fig. 10. The adequacy of the configuration of planes in each opened cube was measured by calculating the condition number of the matrix of plane normals. Table 2 shows the condition numbers corresponding to different tilt angles of the opened cubes, where a tilt of 89° represents an almost degenerate configuration of planes. The point cloud of each opened cube was transformed using the known randomly generated transformations, and to each transformed point cloud Gaussian noise of 0.01 m standard deviation was added. The three solutions were then evaluated using the source and transformed point clouds with and without normalization.

Fig. 11 shows the error measures plotted against the condition number for the three solutions without normalization. The translation error of the SVD plane–plane solution shows again the effect of noise on parameter $\rho$ of the planes, but overall the SVD point–plane solution is clearly more affected by improper plane configurations. Fig. 12 shows the effect of different plane configurations on the performance of the three solutions this time with normalized coordinates. The iterative and SVD plane–plane solution are not affected by improper plane configurations, and yield RMS residual errors consistent with the added noise even at the largest condition number corresponding to 89° tilt, which is almost a flattened cube. In contrast, the SVD point–plane solution performs poorly in terms of all error measures at condition numbers larger than 23.6 (i.e. tilt angles larger than 70°).
In practice, for point cloud registration usually large numbers of points and planes are available. To study the effect of the number of points and planes on the accuracy of the estimates three solutions were evaluated with different numbers of points and planes. Fig. 13 compares the accuracy of the three solutions without normalization against increasing numbers of points per plane. While the performance of the SVD plane–plane solution is again influenced by the low precision of parameter \( \rho \) of planes, the iterative and the SVD point–plane solution have similar accuracies, which improve with increasing number of points in terms of rotation error and translation error. Fig. 14 shows the error measures for the three solutions against increasing numbers of points this
Fig. 12. Effect of planes configuration on the accuracy of the three solutions with normalized coordinates evaluated by RMS residual error (a), rotation error (b) and translation error (c).

Fig. 13. Effect of number of points on the accuracy of the three solutions without normalization evaluated by RMS residual error (a), rotation error (b) and translation error (c).

Fig. 14. Effect of number of points on the accuracy of the three solutions with normalized coordinates evaluated by RMS residual error (a), rotation error (b) and translation error (c).
time with normalized coordinates. Normalization again proves necessary for the SVD plane–plane solution and improves the translation error and consequently the RMS residual error. With normalized coordinates the three solutions perform quite similarly in terms of all three error measures. The rotation errors and translation errors decrease with increasing numbers of points, but the RMS residual errors are smaller than the noise level when fewer points are used and become larger until about 100 points are used. For point numbers larger than 100 the RMS residual errors remain constant at 0.01 m corresponding to the standard deviation of the added noise.

Fig. 15 compares the accuracy of the three solutions without normalization against increasing numbers of planes (each containing 100 points). Except for the 6 planes of the unit cube, all the additional planes were generated randomly (by applying a random rotation to one of the planes of the unit cube) to have arbitrary configurations. Again, the iterative and the SVD point–plane solution outperform the SVD plane–plane solution, when point coordinates are not normalized. Fig. 16 compares the solutions with normalized coordinates. The SVD plane–plane solution still yields slightly larger errors, whereas the iterative and SVD point–plane solution perform equally better in terms of translation error, rotation error and the overall RMS residual error. Increasing the number of planes reduces the translation and rotation errors, but does not have a clear impact on the RMS residual error.

4.7. Computation times

The computation time required by the solutions is largely determined by the number of points and planes used to estimate the transformation parameters. For the iterative and SVD point–plane solution the computation time depends on the number of points, whereas for the SVD-plane–plane solution it depends on the number of planes. Fig. 17 compares the computation time (average over 100 runs) of the three solutions (without normalization) for different numbers of planes. Each plane corresponds to 100 points, so by increasing the number of planes the number of points also increases proportionally. The SVD-plane–plane solution is the fastest with computation times ranging from 0.3 ms to 1.0 ms. The iterative and SVD point–plane solution show linear complexity, while the SVD point–plane solution is about four times faster than the iterative solution. Normalization did not show a significant impact on the computational efficiency of the solutions. Note that the computation times exclude plane fitting, and were obtained by running MATLAB implementations of the solutions on an Intel

Fig. 15. Effect of number of planes on the accuracy of the three solutions without normalization evaluated by RMS residual error (a), rotation error (b) and translation error (c).

Fig. 16. Effect of number of planes on the accuracy of the three solutions with normalized coordinates evaluated by RMS residual error (a), rotation error (b) and translation error (c).
the level of noise. Fig. 18 shows the normalized implementations can further reduce the computation times. Core i7 CPU with 2.10 GHz speed and 8 GB memory. More optimized implementations can further reduce the computation times. Besides the number of points, the efficiency of the iterative solution is also affected by the level of noise. Fig. 18 shows the number of iterations (averaged over 100 runs) for the iterative solution with and without normalization against increasing levels of noise. At lower noise levels the number of iterations stays below 7, but at noise levels larger than 0.1 m (that is 10% of the dimension of the unit cube) the number of iterations increases to reach the maximum 20 iterations. Normalization does not have a significant effect on the number of iterations.

5. Conclusions

In terms of accuracy, the closed-form SVD point–plane solution performs as well as or better than both the iterative and the SVD plane–plane solution, except when the planes are in a very poor configuration. In practice, planes configuration is not a big issue, especially in built environments where vertical and horizontal planar surfaces are often available. The performance of the SVD point–plane solution is very much dependent on normalization. Without normalization the SVD plane–plane solution yields estimates that are consistently less accurate than the other two solutions. With normalization, the SVD plane–plane solution with 6 planes performs as well as the other solutions; however, with increasing number of planes its accuracy becomes lower than that of the iterative and SVD point–plane solution. The accuracy of the iterative solution is bounded by the threshold for terminating the iterations, but in presence of noise it performs as well as the SVD point–plane solution.

In terms of computational efficiency, the SVD plane–plane is the fastest solution followed by the SVD point–plane solution, which runs four times faster than the iterative solution. The computation time of the SVD point–plane solution for large numbers of planes is only a fraction of a second, so it is not a disadvantage of the SVD point–plane solution.

Normalization of the coordinates should be considered an essential preprocessing step for all solutions and particularly for the SVD plane–plane solution, which is more influenced by errors in the plane parameters. Normalization improves the precision of plane fitting and results in a more accurate estimation of the transformation parameters. For the iterative solution, however, it is recommended that normalization involves only mean-centering. Scaling proved to have an adverse effect on the accuracy of the iterative solution.

Acknowledgment

I wish to thank Prof. Wolfgang Förstner (University of Bonn) for the insightful discussions and for introducing me to the interesting properties of singular value decomposition.

Appendix A. Closed-form estimation of scale

In applications where a non-rigid motion is involved the closed-form solution based on point–plane correspondences can be easily extended to estimate the scale parameters. One example application is fitting a polyhedral model to a set of points, for which the closed-form solution based on point–plane correspondences can be used, but the transformation should generally include a non-uniform scale matrix, i.e. a different scale factor along each axis. The equation for point–plane correspondences can then be written as:

$$n_i^T(SR_i^T+t) = \rho_i t + v_{ij}$$  \hspace{1cm} (A1)

where $S$ is a $3 \times 3$ diagonal scale matrix. Note that in practice $S$, $R$, and $t$ are applied to the model (e.g. vertices of a mesh or patch model) rather than the points. However, once the transformation parameters are estimated based on (A1), it is trivial to apply the inverse transformation to the model.

To estimate the scale parameters, we form the system of equations and solve it for $S$ as described in Section 3.4. However, we note that $R$ is now the product of a scale matrix and a rotation matrix: $R = SR$. From the singular value decomposition of $R$ we have: $R = SR = UDV^T$. Since we have an estimate of the rotation matrix as $R = UV^T$, and because both $R$ and $V$ are orthogonal matrices, we can obtain an estimate of $S$ as: $S = UDV^T R = UDV^T UV^T = UDU^T$. This estimate of the scale matrix as $S = UDU^T$ may not necessarily be a diagonal matrix with scale factors on its diagonal, but, together with $R$ and $t$, it guarantees the best fit of the model to the points by minimizing the sum of squared point-model distances.

Appendix B. Precision of plane parameter $\rho$

For a set of points on a plane the normal to the plane can be estimated by singular value decomposition of the covariance of the points. The $\rho$ parameter of the plane is then estimated as: $\rho = n^T \hat{p}$, or:

$$\rho = n_x x + n_y y + n_z z$$  \hspace{1cm} (B1)
with \( n_x, n_y, n_z \) components of the normal vector \( \mathbf{n} \), and \( x, y, z \) coordinates of the mean of the points. Since the estimator for the mean and the estimator for the covariance are uncorrelated (Ash, 2011), we can assume that there is no correlation between the variables on the right hand side of the equation, so the variance of \( \rho \) can be approximated as:

\[
\sigma^2_\rho = x^2 \sigma^2_{n_x} + y^2 \sigma^2_{n_y} + z^2 \sigma^2_{n_z} + n_x^2 \sigma^2_y + n_y^2 \sigma^2_x + n_z^2 \sigma^2_z
\]  

(B2)

If we further assume equal variances for components of \( \mathbf{n} \) denoted by \( \sigma^2_n \), and equal variances for point coordinates denoted by \( \sigma^2_p \) we can write:

\[
\sigma^2_\rho = \sigma^2_n (x^2 + y^2 + z^2) + \sigma^2_p \left( n_x^2 + n_y^2 + n_z^2 \right)
\]  

(B3)

which simplifies to:

\[
\sigma^2_\rho = \sigma^2_n \| \mathbf{p} \|^2 + \sigma^2_p \left( n_x^2 + n_y^2 + n_z^2 \right)
\]  

(B4)

Thus, the precision of plane parameter \( \rho \) is a function of the precision of the plane normal, as well as the mean, noise and number of the points.

References