

ON REGISTRATION OF VECTOR MAPS WITH KNOWN CORRESPONDENCES

Ales JELINEK, Adam LIGOCKI

Central European Institute of Technology, Brno University of Technology,
Purkynova 123, 612 00 Brno, Czech Republic

ales.jelinek@ceitec.vutbr.cz, adam.ligocki@ceitec.vutbr.cz

DOI: 10.15598/aeec.v17i3.3451

Abstract. *Data association and registration is an important and actively researched topic in robotics. This paper deals with registration of two sets of line segments, which is especially useful in mapping applications. Our method is non-iterative, finding an optimal transformation in a single step, in a time proportional only to a number of the corresponding line segments. The procedure also provides diagnostic measures of reliability of the computation and of similarity of the data sets being registered. At this point, the method presumes known correspondences, which is limiting, but the discussion, in the end, reveals some possibilities to overcome this issue. Practical properties are demonstrated on a typical task of localization of a robot with a known map.*

Keywords

Correspondence search, least squares, point cloud, registration, vectorization.

1. Introduction

In robotics, a perception system of a robot is essential for its operation in the environment. Basic behaviour, such as collision avoidance, can work without any higher interpretations of the sensed data, because the geometrical information needed is either directly measured (laser scanning) [1] or derived exclusively from the incoming data (stereo-vision) [2]. Complex behaviour of the robot, possibly leading to Artificial Intelligence (AI), requires much more. One of the key operations is association of the recent measurements with previous knowledge of the robot, which should open a way for learning (accumulation of knowledge) and reasoning (usage of the learned facts) [3].

Data association problem is tightly coupled with the concept of similarity. Geometrical similarity is not sufficient in this case because of its binary nature - objects in traditional geometry can be similar or not, but nothing in between. In our everyday human experience, similarity of objects is a smooth metric, describing intuitively evident, but hardly evaluable "distance" between two objects. Modern robotics in search of AI aims at complex object recognition, which also leads to extremely complex similarity evaluation. This is the reason why data association problem is so hard and why it is still an open research topic.

Similarity can be examined in various ways for a pair of objects because there are usually many distinct features possible to identify on them. Overall similarity is then a function of similarities between corresponding features. Of course, if a feature is too complex, it can be deconstructed again and again, possibly until its mathematical and physical foundations are revealed. Human introspection is not able to dive into the unconscious mind deep enough, but according to some scientists [4], this is how our sense of similarity works as well.

A frequently examined feature is correlation of two data sets. In the case of robotics, this task appears every time a new measurement of the robot's surroundings is acquired and needs to be aggregated with its internal representation of the world. The process of data fitting is usually referred to as registration, and an optimum is reached when the correlation (aka similarity in this case) is maximal. Our method registers two sets of line segments (vector maps) in a single step and provides diagnostic information on reliability and actual similarity of both sets. A downside is the necessity of data association before computation, but in the final discussion about further research, we will show that the method is usable for correspondence search as well.

2. State of the Art

As can be seen in a wide range of literature, robotic mapping was based on point-like features for a long time [5], [6] and [7]. On the other hand, this method seems to face its limits, and the recent research changes its direction towards more complex objects [3].

2.1. Iterative Closest Point Method

Iterative Closest Point (ICP) [8] is by far the most popular method for registration of geometrical data today [9]. It is able to work both in 2D and 3D space and can be applied on point sets as well as many other geometrical primitives as well. Authors also provide a proof of convergence to the local minimum. If used with a fast space partitioning data structure, the method is computationally efficient and easily parallelizable.

Generalized ICP [10] extends the original method with a probabilistic approach and promises better convergence and accuracy. Many other variants exist [11] and [12], especially those, that address the problem of convergence to the local minimum. Various pre-optimization techniques are used for this reason, for example, geometric features [13], or genetic algorithms [14].

2.2. Alternatives to ICP

Many alternatives to the ICP method exist, but their usage is much less frequent because there is usually some significant drawback reducing their applicability.

Correlation already mentioned in the introduction is a good example. Although it is theoretically capable to find a globally optimal registration transformation, the exhaustive search algorithm cannot do it with reasonable computational costs, which disqualifies it from most of the practical applications.

Random sample consensus method [15] is based on a random sampling of possible correspondences. Although it is a stochastic method, which does not guarantee convergence, it is frequently used as a part of more complex registration systems.

Principal component analysis stems from statistical properties of the processed point cloud. It is useful for point clouds with a simple shape [12], but more complex geometrical primitives or dissected shapes of the data sets are out of its possibilities.

Computer vision provides methods based on feature extraction, which are easier to associate and much sparser, reducing computational costs. Scale-Invariant Feature Transform (SIFT) [16] and Speeded Up Robust Features (SURF) [17] are well-known examples,

but their applicability in geometrical measurements is limited.

2.3. Geometrical Primitives in the Registration Process

Methods specifically designed to work with line segments, curves etc. are very rare. More possible reasons for this situation might be found, but most probably it was the generality and simplicity of the ICP method, which prevented researchers from further investigation. In addition, paper [18], published shortly after the original ICP proposal, came with criticism of the primitive based approach and suggested points as a more robust alternative.

On the other hand, the problem of registration and association is not yet satisfyingly solved, and isolated attempts to achieve this goal using geometrical primitives can be found. Iterative closest line algorithm [19] directly generalizes ICP to line segments, while the other approaches [20] and [21] present completely novel approaches. Some modifications to the original ICP method make use of geometrical primitives as well [10] and [22].

All of these examples prove that geometrical primitives have their benefits. In [23], we have shown, that proper approximation using a more complex geometric primitive can reliably reduce the noise inherently present in the point cloud, which disproves the criticism in [18]. In combination with the general tendency towards object-oriented mapping in robotics [3], the complex object registration seems to be a viable research direction.

3. One-Step Registration Method

As stated in the Introduction, the method is derived for corresponding pairs of line segments in two-dimensional space, where one line segment in the pair belongs to the static set (in robotics usually a map) and the second comes from the dynamic one (new measurement). The output of the method is a transformation applicable to the dynamic set, which leads to optimal registration with the static line segments. Figure 1 depicts one corresponding pair with labelling, as used in the following computations. Note that the line segments are oriented by begin and end points with appropriate directional vector. If the points were swapped, they would denote different line segment.

We can find an infinite number of transformations of the dynamic line segment ℓ_D in each pair, which will move it on the same line with its static counterpart ℓ_S .

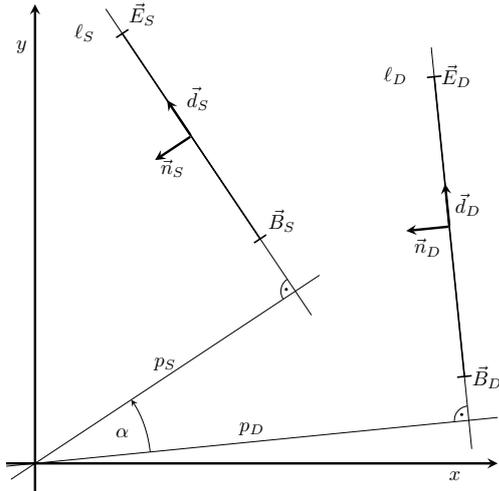


Fig. 1: Static (ℓ_S) and dynamic (ℓ_D) line segments before the fitting process. All points, vectors and distances important for further computations are marked out.

To allow registration of incomplete data, all of them are treated as equally suitable. An object in a 2D space has three degrees of freedom, therefore each transformation is described by three independent variables and corresponds to a single point in a three dimensional space of all possible transformations \mathbb{T} . The set of all ideal transformations bringing ℓ_S and ℓ_D in line can be described as a one dimensional subspace t_i of \mathbb{T} . Using notation from Fig. 1, we get an equation:

$$t_i : \begin{bmatrix} (p_{Di} - p_{Si})\vec{n}_{Si} \\ \alpha_i \end{bmatrix} + \tau_i \begin{bmatrix} \vec{d}_{Si} \\ 0 \end{bmatrix} = \vec{P}_i + \tau_i \vec{v}_i, \quad (1)$$

where $p_{Di} = \vec{B}_{Di} \cdot \vec{n}_{Di}$, $p_{Si} = \vec{B}_{Si} \cdot \vec{n}_{Si}$ and arbitrary coefficient $\tau_i \in \mathbb{R}$. \vec{P}_i and \vec{v}_i are denominations for further computations, all written for an i -th corresponding pair from the input sets of line segments. An example of those transformation lines in \mathbb{T} is shown in Fig. 2.

The optimal transformation for the whole data set is then found as the closest point to all of the transformation lines $t_1 \dots t_N$, where N is the number of corresponding pairs. Using standard vector formula for point-to-line distance, we get:

$$l_i = \frac{\|(\vec{T} - \vec{P}_i) \times \vec{v}_i\|}{\|\vec{v}_i\|}, \quad (2)$$

where \vec{T} is the optimal transformation, and \vec{P}_i and \vec{v}_i come from Eq. (1). \vec{T} is found using the total least squares method, and for additional flexibility, an arbitrary weight w_i for each pair was added. The problem formulated by the equality:

$$\sum_{i=1}^N w_i \frac{\partial l_i^2(\vec{T})}{\partial \vec{T}} = \frac{\partial (\|\vec{T} - \vec{P}_i\|^2 - (\vec{T} - \vec{P}_i) \cdot \vec{v}_i)^2}{\partial \vec{T}} = \vec{0}, \quad (3)$$

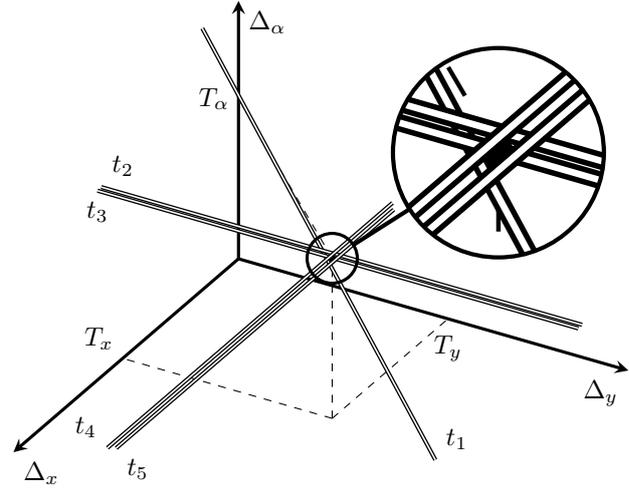


Fig. 2: An example of the search for an optimal transformation in the space of all possible transformations \mathbb{T} . The double lines stand for the perfect transformations of each corresponding pair and the optimal transformation is shown as a black dot in the magnified area.

leads to an equation:

$$\sum_{i=1}^N w_i \vec{T} - \sum_{i=1}^N w_i \vec{v}_i (\vec{T} \cdot \vec{v}_i) = \sum_{i=1}^N w_i \vec{P}_i, \quad (4)$$

which in matrix notation looks as follows:

$$\begin{bmatrix} \sum w_i v_{yi}^2 & -\sum w_i v_{xi} v_{yi} & 0 \\ -\sum w_i v_{xi} v_{yi} & \sum w_i v_{xi}^2 & 0 \\ 0 & 0 & \sum w_i \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_\alpha \end{bmatrix} = \begin{bmatrix} \sum w_i P_{xi} \\ \sum w_i P_{yi} \\ \sum w_i P_{\alpha i} \end{bmatrix}. \quad (5)$$

For convenience, let us define a simplified notation for terms of the matrices:

$$\begin{bmatrix} S_{y2} & -S_{xy} & 0 \\ -S_{xy} & S_{x2} & 0 \\ 0 & 0 & S_w \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_\alpha \end{bmatrix} = \begin{bmatrix} S_{Px} \\ S_{Py} \\ S_{P\alpha} \end{bmatrix}. \quad (6)$$

Equation (5) clearly shows, that translational and rotational parts of the optimal transformation can be found independently, which is a consequence of single optimal rotation for each corresponding pair. Solutions for the translational part is straightforward:

$$T_x = \frac{D_1}{D}, \quad T_y = \frac{D_2}{D}, \quad (7)$$

where (using the simplified notation):

$$\begin{aligned} D &= S_{x2}S_{y2} - S_{xy}^2, \\ D_1 &= S_{Px}S_{x2} - S_{xy}S_{Py}, \\ D_2 &= S_{Py}S_{y2} - S_{xy}S_{Px}. \end{aligned} \quad (8)$$

Situation of the rotational part of the transformation is more complicated because an angle is a circular quantity, so the classical average computation:

$$T_\alpha = \frac{S_{P\alpha}}{S_w}, \quad (9)$$

might lead to an unexpected result. When averaging two angles, there are always two possible outcomes due to circularity, and there is no guarantee, that the right one (expected one) will be obtained. To get the correct results, the Mitsuta's averaging method [24] has been used. An algorithmic implementation of the method is based on [25] and follows the pseudocode of Alg. 1. The output variables $S_{w\alpha}$ and $S_{w\alpha^2}$ should be used in place of $\sum w_i\alpha_i$ and $\sum w_i\alpha_i^2$ in all equations in this paper.

Algorithm 1 Mitsuta's averaging of angles.

Input: $[\alpha_1 \dots \alpha_N]$, such that $-\pi < \alpha_i < \pi$,
 $[w_1 \dots w_N]$ such that $w_i \in \mathbb{R}$.

Output: Correct weighted sums of α_i ($S_{w\alpha}$) and α_i^2 ($S_{w\alpha^2}$).

```

1:  $S_{w\alpha} = S_{w\alpha^2} = \alpha_{old} = 0$ 
2: for all  $\alpha_i, w_i$  do
3:   if  $i = 1$  then
4:      $\alpha_{old} = \alpha_i$ 
5:      $S_{w\alpha} = w_i\alpha_i$ 
6:      $S_{w\alpha^2} = w_i\alpha_i^2$ 
7:   else
8:      $\delta = \alpha_i - \alpha_{old}$ 
9:     if  $\delta < -\pi$  then
10:       $\delta += 2\pi$ 
11:     else if  $\delta > \pi$  then
12:       $\delta -= 2\pi$ 
13:     end if
14:      $\alpha_{old} += \delta$ 
15:      $S_{w\alpha} += w_i\delta$ 
16:      $S_{w\alpha^2} += w_i\delta^2$ 
17:   end if
18: end for

```

The core of the method is finished at this point because we are able to compute the optimal transformation of the dynamic line segment set in a single step. On the other hand, there are several further topics worth discussion, which will be addressed in the following sections.

3.1. Alternative Computation of Optimal Rotation Compatible with 3D Pose Estimation

The previously explained procedure of computation of T_α is perfectly suitable for 2D problems, but in practice, we often need to take into account the third di-

mension as well. Pose estimation and especially orientation estimation in 3D is more tricky compared to 2D case because direct computation with angles leads to non-linear equations. To keep linearity, an alternative formulation of the error metric is being used. The first proposal of this approach was the Wahba's problem [26] stated as follows:

Given two sets of N points $\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N\}$ and $\{\vec{v}'_1, \vec{v}'_2, \dots, \vec{v}'_N\}$, where $N \geq 2$, find the rotation matrix \mathbf{R} (i.e. the orthogonal matrix with determinant +1) which brings the first set into the best least-squares coincidence with the second. That is, find \mathbf{R} which minimizes:

$$\sum_{i=1}^N \|\vec{v}'_i - \mathbf{R}\vec{v}_i\|^2. \quad (10)$$

We see from Eq. (10) that the minimized quantity is not the angle between corresponding vectors from both sets, but rather the Euclidean distance between their tips. Different criterion function gives slightly different optimal rotation; therefore, strictly speaking, makes both methods incompatible. If the perfect compatibility of computation of T_α with the Wahba's approach is required, we provide the following procedure.

State of the art contains plenty of methods for dealing with the Wahba's problem in 3D [27], but for 2D vectors, a great simplification is possible. Let us restate the Eq. (10) using the notation from Fig.1:

$$E_\alpha = \sum_{i=1}^N w_i \|\vec{d}_{S_i} - \mathbf{R}\vec{d}_{D_i}\|^2, \quad (11)$$

where E_α is the total error and the rotation matrix \mathbf{R} is as follows:

$$\mathbf{R} = \begin{bmatrix} \cos T_\alpha & -\sin T_\alpha \\ \sin T_\alpha & \cos T_\alpha \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}. \quad (12)$$

For optimal T_α , the Eq. (11) differentiated with respect to T_α should be equal to zero. First, we expand and simplify the relation to the form:

$$E_\alpha = 2 \sum_{i=1}^N w_i - 2s \sum w_i n_{S_i} \cdot \vec{d}_{D_i} - 2c \sum w_i \vec{d}_{S_i} \cdot \vec{d}_{D_i}, \quad (13)$$

which can be trivially differentiated:

$$\frac{\partial E_\alpha(T_\alpha)}{\partial T_\alpha} = 2s \sum w_i \vec{d}_{S_i} \cdot \vec{d}_{D_i} - 2c \sum w_i n_{S_i} \cdot \vec{d}_{D_i} = 0, \quad (14)$$

and rearranged giving tangent of the optimal T_α :

$$\tan T_\alpha = \frac{\sin T_\alpha}{\cos T_\alpha} = \frac{\sum w_i n_{S_i} \cdot \vec{d}_{D_i}}{\sum w_i \vec{d}_{S_i} \cdot \vec{d}_{D_i}}. \quad (15)$$

T_α can be easily retrieved by an atan2(y, x) function provided by every decent mathematical library.

3.2. Metric for the Transformation Space and an Isomorphism with a Spring Net Equilibrium

A more careful examination of the Eq. (2) reveals that the computation requires existence of a metric on the transformation space \mathbb{T} , but we have not defined it. Derivation of the optimal transformation silently presumes usual Euclidean metric on \mathbb{E}^3 , and at the point, where this expectation would fail, the computation naturally splits into two independent parts for the translational and the rotational components of the transformation (see the zeros in the left matrix in the equation Eq. (5)) and the correct results are obtained separately. This dirty trick means that we have used distance between transformations in the computation, but we are not able to actually evaluate it. Such discrepancy definitely needs clarification.

Of course, the problem lies in the fact that linear and angular quantities would be carelessly summed up during Euclidean metric evaluation, which is physically inadmissible. Fortunately, inspiration from a completely unrelated physical problem offers an elegant solution. The optimal transformation was found using the total least squares method, which essentially minimizes the sum of the squared distances from the transformation (point in \mathbb{T}) to the transformation lines Eq. (1). Similar situation arises in case of a net of springs connected together, for which we want to find an equilibrium with the smallest sum of potential energies $\sum E_i$. Since the potential energy of a spring is proportional to a square of its displacement from the quiescent state, multiplied by a stiffness coefficient according to a formula $E_l = \frac{1}{2}k_l l^2$, the computation results in minimization of squared distances as well, making both tasks mathematically isomorphic.

Potential energy for torsion springs is defined in a similar way ($E_a = \frac{1}{2}k_a \alpha^2$), and potential energies of both kinds of springs can be obviously summed up. Introduction of linear and angular coefficients k_l and k_a allows physically sound summation of the translational and rotational portions of the total squared distance between two transformations. It is also important to point out that introduction of the coefficients does not affect position of the optimal transformation because linear scaling of a function does not move its minima.

Mitsuta's averaging could be also incorporated into the metric of \mathbb{T} and derivation of the method could be rebuilt on those foundations, but we do not think it is necessary. Considerations above fully legitimise formulas for the optimal transformation and will be further used in the following section.

3.3. The Method in Context of General Similarity

The computation above provides an optimal transformation for any given set of line segment pairs (except situation, when discriminant D from Eq. (8) is zero), but there is no measure describing, how compromise the solution actually is. In principle, this means exposure of the internal criterion function, but with proper scaling using the linear and angular coefficients described previously. Sum of the squared distances between transformation lines and the optimal transformation \vec{T} , derived from Eq. (2), is:

$$\sum l_i^2(\vec{T}) = \sum w_i((T_\alpha - P_{\alpha i})^2 + (T_x - P_{xi})^2 v_{yi}^2 + (T_y - P_{yi})^2 v_{xi}^2 - 2(T_x - P_{xi})(T_y - P_{yi})v_{xi}v_{yi}), \quad (16)$$

where T_x , T_y and T_α are parameters of the optimal transformation as derived above, and the rest of the variables comes from the definition Eq. (1). Total contribution of rotational part of the distance between transformations can be expressed as:

$$A_\alpha = k_\alpha \left(\sum w_i P_{\alpha i}^2 - T_\alpha \sum w_i P_{\alpha i} \right), \quad (17)$$

where k_α is a coefficient for the rotational movement, and $\sum w_i P_{\alpha i}^2$ is a new sum, which needs to be precomputed. Methodology for averaging circular quantities in [25] and pseudocode in Alg. 1 cover this matter as well.

In case we work with T_α computation described in Sec. 3.1. we use the sum of squared errors given by Eq. (13) and compute A_α as a product:

$$A_\alpha = k_\alpha E_\alpha. \quad (18)$$

Total contribution of the translational part of the distance is somewhat more complicated to derive, but can be simplified down to the form:

$$A_t = k_t \left(\sum w_i (P_{xi}v_{yi} - P_{yi}v_{xi})^2 - T_x^2 \sum w_i v_{yi}^2 + 2T_x T_y \sum w_i v_{xi}v_{yi} - T_y^2 \sum w_i v_{xi}^2 \right). \quad (19)$$

where k_t is a coefficient for the translational portion of the distance. Similar to the equation Eq. (17), there is only one additional sum to be precomputed. Ambiguity A of the solution is then expressed by a simple sum of the angular and linear components:

$$A = A_\alpha + A_t. \quad (20)$$

Ambiguity evaluation provides a control mechanism expressing ambivalence of the data sets. If the fitting is perfect, the lines of optimal transformations intersect at one point, and the ambiguity is zero. Noised measurements from practical experiments exhibit some ambiguity, but it stays limited. If the limit is exceeded,

a strong suspicion of badly set correspondences is in place. Equation (17), Eq. (19) and Eq. (20) work for any transformation in place of \vec{T} . For example, if an ambiguity before registration is needed, T_x , T_y and T_α would be zero.

The notion of similarity appears regularly in literature, but there is no widely accepted definition. The only common property corresponds to an intuitive expectation that the more similar the objects are, the higher is the number representing it. Reciprocal of the ambiguity as defined above seems to be a good candidate for similarity evaluation of the sets of line segments.

3.4. Reliability Evaluation of the Optimal Transformation

Every computation that may fail needs some way to detect hazardous results and report mistakes, if it is meant to be used in practice. Our registration method for line segment sets fails, when all line segments are collinear, which is correct behaviour and can be easily detected because in such case the determinant D from Eq. (8) is zero. In real-world situations, exact collinearity rarely appears, but nearly collinear lines can cause wrong results as well, and this can happen more frequently. Such situation produces perfectly solvable equations, but the computation is extremely sensitive to noise as depicted in Fig. 3. Continuous reliability metric covering the situation from degenerate input to perfect data from synthetic tests is therefore mandatory.

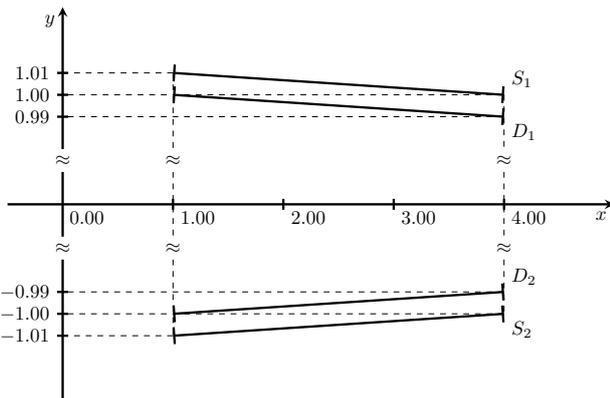


Fig. 3: Nearly collinear line segments make the registration process extremely sensitive to noise. Tiny deviation of the line segments D_i will cause translation by three units in the x direction after registration. Reliability evaluation is designed to detect such hazardous situations.

To evaluate the reliability, we have decided to examine direction vectors of the registered line segments because if most of them point in the same direction, the result will be less reliable, than in a case, when

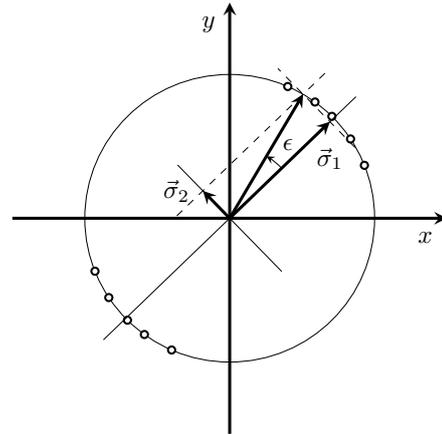


Fig. 4: Small circles represent direction vectors of the static line segments and their opposite counterparts. Vectors $\vec{\sigma}_1$ and $\vec{\sigma}_2$ correspond to eigenvectors of \mathbf{E} with lengths $\sqrt{\lambda_1}$ and $\sqrt{\lambda_2}$ respectively. The angle ϵ defines reliability.

there are many orthogonal pairs. The main idea of the computation is shown in Fig. 4. For each direction vector, its opposite counterpart is added as well, which ensures, that a mean of their coordinates becomes zero and collinear line segments with opposite direction vectors will appear the same for the reliability evaluation purpose. For this purpose, the precomputed sums change as follows:

$$\begin{aligned} \sum_{i=1}^N w_i v_{xi} &\Rightarrow 0, & \sum_{i=1}^N w_i v_{xi}^2 &\Rightarrow 2 \sum_{i=1}^N w_i v_{xi}^2, \\ \sum_{i=1}^N w_i v_{yi} &\Rightarrow 0, & \sum_{i=1}^N w_i v_{yi}^2 &\Rightarrow 2 \sum_{i=1}^N w_i v_{yi}^2, \\ \sum_{i=1}^N w_i &\Rightarrow 2 \sum_{i=1}^N w_i, & \sum_{i=1}^N w_i v_{xi} v_{yi} &\Rightarrow 2 \sum_{i=1}^N w_i v_{xi} v_{yi}. \end{aligned} \quad (21)$$

Principal component analysis is used to inspect dispersion of the direction vectors. The vectors demarcate points on a unit circle, and for those points, a correlation matrix \mathbf{E} can be found:

$$\mathbf{E} = \frac{1}{\sum w_i} \begin{bmatrix} \sum_{i=1}^N w_i v_{xi}^2 & \sum_{i=1}^N w_i v_{xi} v_{yi} \\ \sum_{i=1}^N w_i v_{xi} v_{yi} & \sum_{i=1}^N w_i v_{yi}^2 \end{bmatrix}. \quad (22)$$

Eigenvalues are computed using the standard formula $\text{Det}(\mathbf{E} - \lambda \mathbf{I}) = 0$, which leads to a quadratic equation:

$$\begin{aligned} \lambda^2 - \lambda \frac{(\sum w_i v_{xi}^2 + \sum w_i v_{yi}^2)}{\sum w_i} + \\ + \frac{(\sum w_i v_{xi}^2 \sum w_i v_{yi}^2 - (\sum w_i v_{xi} v_{yi})^2)}{(\sum w_i)^2} = 0. \end{aligned} \quad (23)$$

Since:

$$\sum w_i v_{x_i}^2 + \sum w_i v_{y_i}^2 = \sum w_i (v_{x_i}^2 + v_{y_i}^2) = \sum w_i, \quad (24)$$

and

$$\frac{\left(\sum w_i v_{x_i}^2 \sum w_i v_{y_i}^2 - \left(\sum w_i v_{x_i} v_{y_i}\right)^2\right)}{\left(\sum w_i\right)^2} = \text{Det}(\mathbf{E}), \quad (25)$$

the Eq. (23) can be simplified in the following manner:

$$\lambda^2 - \lambda + \text{Det}(\mathbf{E}) = 0. \quad (26)$$

The relationship of the eigenvalues λ_1 and λ_2 is:

$$\begin{aligned} \lambda_{1,2} &= \frac{1}{2} \pm \sqrt{\frac{1}{4} - \text{Det}(\mathbf{E})}, \\ \lambda_{1,2} &= \frac{1}{2} \pm \sqrt{\frac{1}{4} - \lambda_1 \lambda_2}, \\ \lambda_{1,2}^2 - \lambda_{1,2} + \frac{1}{4} &= \frac{1}{4} - \lambda_1 \lambda_2, \\ \lambda_1^2 - \lambda_1 &= \lambda_2^2 - \lambda_2, \\ \lambda_1 + \lambda_2 &= 1. \end{aligned} \quad (27)$$

Usage of the formulas $\sin^2 \epsilon + \cos^2 \epsilon = 1$ and $\sin(2\epsilon) = 2 \sin \epsilon \cos \epsilon$ leads to the criterion value:

$$\mathbf{R} = \sin(2\epsilon) = 2\sqrt{\lambda_1 \lambda_2} = 2\sqrt{\text{Det}(\mathbf{E})}. \quad (28)$$

The definition of reliability directly shows its properties. Because the positive result of the square root is taken, the value of \mathbf{R} can change in the interval $[0; 1]$, where zero demarcates an unsolvable situation and one means the perfect reliability. There is a smooth transition between these two states, so the unreliable results caused by nearly collinear line segments can be easily detected.

3.5. Implementation Considerations

Besides subtleties of averaging of angles, which were already discussed above, there are two more implementation details worth mentioning. The first is precomputation of sums. Throughout the computation, there are many sums, which take linear time to compute, proportional to the total number of corresponding pairs N . In many practical situations, we need to modify the sets slightly and recompute the registration again. Keeping the sums and adding or subtracting the values corresponding to the pair, which is currently being added or removed, can save a lot of processing time. The same can be applied to a whole set of correspondences, for which the sums are known. This way, the modification of the precomputed sums take constant time, independent of N , greatly improving the efficiency of algorithms using our method.

Second important note regards numerical stability. When adding a lot of floating point numbers with certain precision and storing them in a variable of the same byte length, inevitable loss of accuracy occurs, which may negatively affect the computation. Our implementation in C++ uses float type for line segment parameters and double precision for sums. This precaution ensures enough precision for stable computation.

4. Experimental Verification

This section presents several localization experiments with a priori known maps and artificially generated data. This approach allows us to explore a large amount of combinations of data parameters, environments, special cases and registration algorithm set-ups, which would be hard or even impossible in practice. We have chosen three simple scenarios with special characteristics (see Fig. 5), which should illustrate all important aspects of the registration process. We start with a virtual laser scanner positioned in several places in the map, where simulated data are acquired. Number of points per scan and Gaussian noise can be set individually for every measurement. Point-like data are processed using the total least squares vectorization algorithm [23], providing a set of line segments to be registered with the map. Exactly given poses of the scanner serve for the verification of quality of the registration process. Virtual environment is also useful because the correspondences between scan and map are known for sure, which simplifies experimentation and removes a potential source of human errors.

4.1. Influence of Line Segment Accuracy on the Registration Process

Since both vectorization of the points as well as registration of the line segments are performed in the least-squares sense, we would expect that with growing number of input data and decreasing noise, the accuracy will improve. For this reason, we have prepared a set of experiments, whose results are summed up in Fig. 6. The expectation is perfectly satisfied, and we observe a great impact of both parameters. Both linear and angular error decrease linearly with the growing number of input points and quadratically with the diminishing noise. We have also compared the outputs of Eq. (19) and Eq. (17) plotted in green in Fig. 6 with the dispersion of estimated poses around the true pose (plotted in gray). Estimated error is (up to scale) equivalent to the ground truth, which justifies its usage in practice.

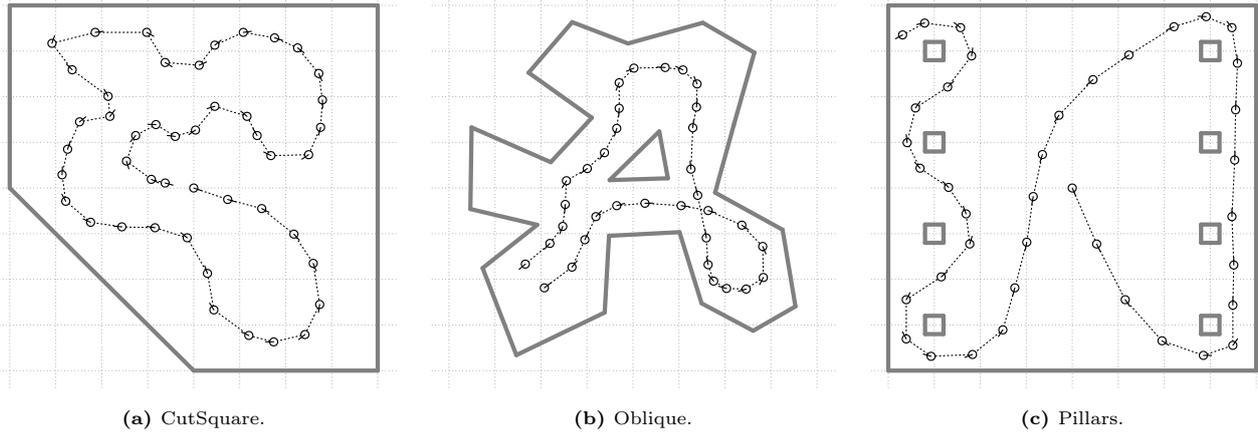


Fig. 5: The three testing environments for registration and localization experiments. Thick gray line segments demarcate the obstacle edges and the black dashed path corresponds to the trajectory of the virtual robot with marked out poses, where the surround scanning was performed.

For better visualization, in Fig. 7 we have plotted the positional and rotational dispersion under varying scanner parameters. The statistics are based on 100 virtual scans taken from every given point. The ellipses correspond to dispersion of estimated positions, and blue pies represent uncertainty in orientation. We can visually confirm findings from Fig. 6 regarding point density and noise level influence. Also, the central points of the ellipses and axes of the pies are closer to the true pose, which proves another frequently used technique: If we have a poor sensor with low point density and high noise, taking several scans in one place can help to increase the accuracy.

There is one more interesting phenomenon to be observed in experiments in Fig. 7a, Fig. 7b, Fig. 7c, Fig. 7d and Fig. 7e. The dispersion ellipses at a given point are changing their size with changing conditions, but their shape and tilt are nearly constant despite the number of points or the noise level. This is given by the structure of the surrounding environment and shows the importance of reliability and ambiguity evaluation. Even such a simple scenario as the CutSquare environment has varying informational gain, when scanned from different positions.

Even more volatile in this regard is the Pillars environment, where the view is obstructed by narrow obstacles. Significance of this behavior can be spotted in Fig. 7f, where mediocre point number and noise amplitude result in highly differing accuracy of results in various points of the trajectory. Also note, that some dispersion ellipses lap over the map edges into the mass of the obstacle. The registration method has no mechanism to detect such erroneous results, but additional verification after the registration process is easy to append.

4.2. Correspondence Weighting

The equations for optimal transformation derived in the theoretical part of this paper are all designed to incorporate weight allowing prioritization of certain corresponding pairs over the others. This feature was not utilized in the previous section, but its potential to

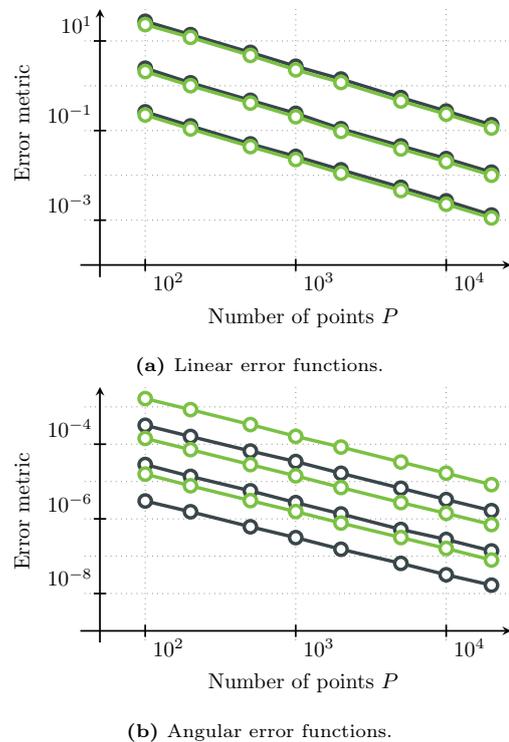


Fig. 6: Error functions (green for A_t from Eq. (19) and gray for comparison with ground truth) of the registration process. The plots belong to the CutSquare environment scanned with varying point density and (horizontal axis) and noise level ($\sigma = 1, 3, 10$ units in the bottom-up order). Weighting was not used during the experiments, all corresponding edges were assigned a weight of one.

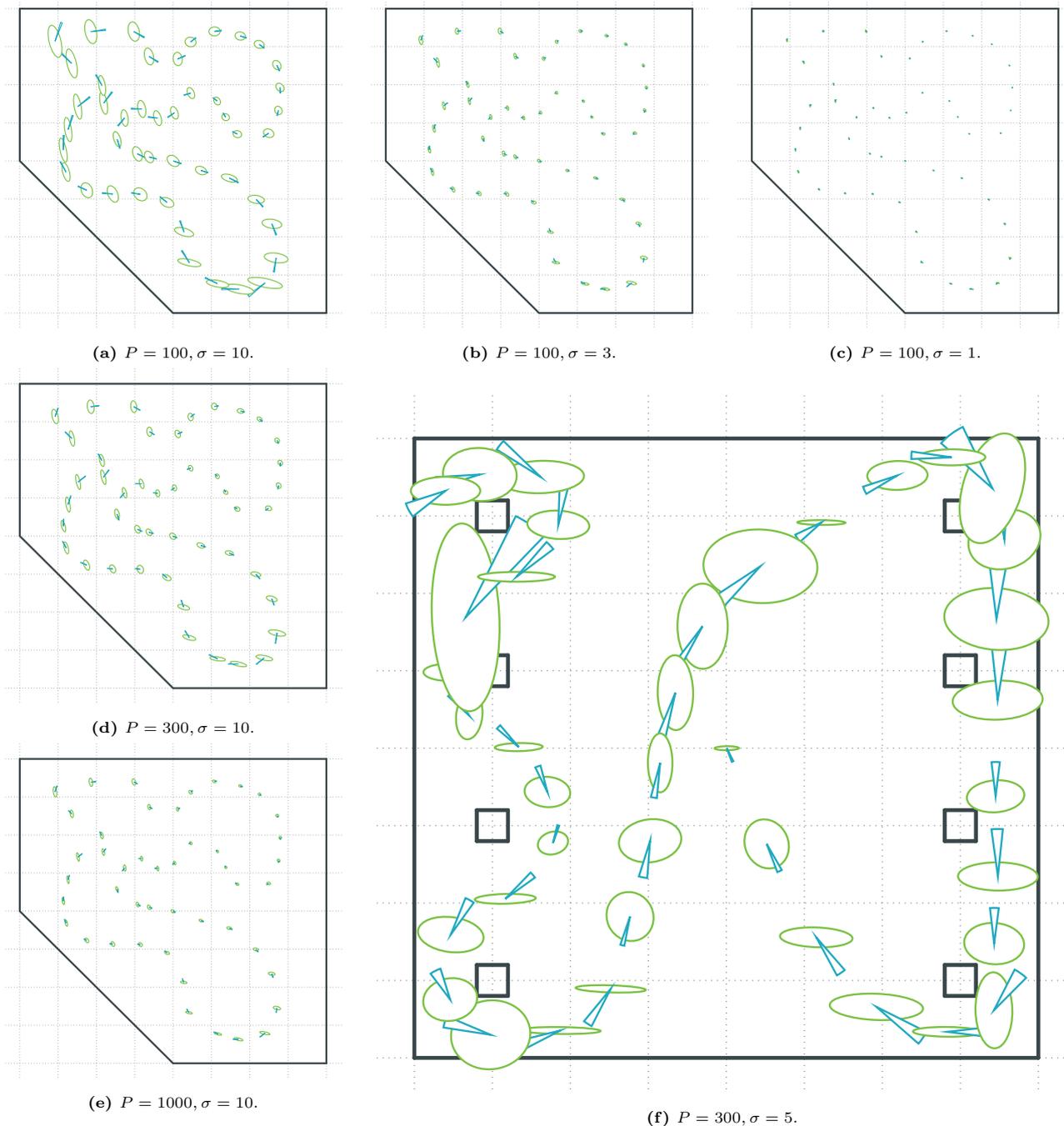


Fig. 7: An influence of sampling density (number of points per scan P) and measurement noise (with standard deviation σ) on line segment accuracy and subsequent effect on the registration process. Green ellipses demarcate positional dispersion and blue pies stand for range of orientations of the reconstructed poses. Dependence of quality of the registration on quality of the input data is obvious. Grid size is 50×50 units, the parameter σ is measured in these units as well. Weighting was not used during the experiments, all corresponding edges were assigned a weight of one.

improve the accuracy of the registration results is substantial.

Shorter line segments tend to be less accurate, because in practical measurements, usually, less data is available to define them. Giving higher weight to the corresponding pairs, where both line segments are long is therefore sensible. For our experiments, we have

used the length of the shorter line segment to define the weight of each pair and significant accuracy gain have appeared. Confronting Fig. 8 obtained through weighted computation with Fig. 7f, where all pairs were treated equally, shows clear improvement on the same data. Common practice is to discard the lowest accuracy line segments right after vectorization, but this way either permits excessive error if the threshold is too

low, or rejects useful information. Weighting provides more fluent transition between high and low accuracy pairs and leads to better results.

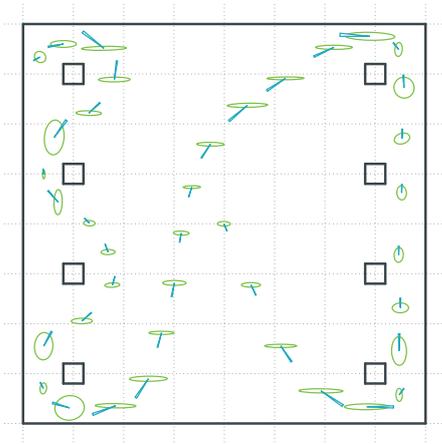


Fig. 8: The Pillars environment scanned with the same parameters ($P = 300, \sigma = 5$) as in Fig. 7f, but with weighting proportional to the length of the shorter of the corresponding edges. Improvement of the registration accuracy over Fig. 7f is significant.

Tab. 1: Ratios of the weighted over the non-weighted error metrics in various environments. Weighting is proportional to the length of the shorter of the corresponding edges and significantly increases accuracy of the registration procedure. (Lower is better.)

Error metric	Environment		
	CutSquare	Oblique	Pillars
A_t	0.68	0.46	0.27
A_α	0.65	0.43	0.25
D_t	0.58	0.25	0.07
D_α	0.66	0.27	0.07

To explore the impact of weighting under various conditions, we have performed the same set of experiments as in the previous section, but interestingly, there was no statistically significant impact of point density or noise level on the improvement caused by weighting. In other words, the accuracy gain from weighting is mostly given by the environment. The three scenarios depicted in Fig. 5 exhibit growing variability in edge lengths, which nicely corresponds to results summarized in Tab. 1. The error reduction in the simple CutSquare environment is roughly 30–40 %. In the more structured Oblique scenario, the reduction is higher than 50 %, and the Pillars environment with its narrow beams exhibits even bigger improvement. The energy-based and dispersion-based error metrics are influenced in different magnitude, but the behavior of the error functions presented in the previous section is still perfectly valid. Positive effect of sensibly set weights is clear and can significantly improve registration results, especially in a complicated environment, where we need it the most.

Tab. 2: Correlation of the linear and angular ambiguity with distance from the true pose. The w index stands for experiments, where weighting as described in Sec. 4.2. was applied, while uniformly weighted experiments have no special marking.

Correlation coefficient	Environment		
	CutSquare	Oblique	Pillars
$Corr_t$	0.90	0.91	0.77
$Corr_\alpha$	0.59	0.82	0.58
$Corr_{tw}$	0.91	0.80	0.74
$Corr_{\alpha w}$	0.53	0.56	0.61

4.3. Ambiguity Correlation with Distance from True Pose

Ambiguity is directly connected to the accuracy of the registration process, although we cannot take it as an exact measure related to some true value, which is obviously unknown. Instead, the ambiguity reflects how well the transformations aligning the corresponding line segments coincide with each other. We have examined correlation between the linear and angular ambiguities and a square of linear and angular error computed using the ground truth.

The results of the correlation measurement are summarized in Tab. 2. No significant influence of point cloud density or noise level was spotted. Similar to the previous section, all remarkable differences were given by the environment where the experiment was performed. We see a strong correlation for the translational part of the transformation, while the ambiguity of the rotational part is less dependent on the real misalignment. Weighting slightly reduces the correlation, however rough estimation of the registration accuracy is still possible.

4.4. Insufficient Line Segments and Reliability

So far, we have investigated only the accuracy in experiments, where enough data for computation was always available. The reliability evaluation was silently omitted because in all cases, we could say that \mathbf{R} in the sense of Eq. (28) closes to one. As illustrated along with the theoretical derivation using the nearly colinear line segments example (see Fig. 3), hazardous cases can easily occur in practical situations. To display less straightforward problems with reliability, we have employed the Pillars environment again.

Figure 9 shows an experiment, where the point density, noise level and edge rejection threshold are set to generate data, for which the registration is nearly unsolvable. The threshold of 80 units means that from every examined pose it is possible to spot at least two linearly independent edges of the map, but lower point density and high noise result in problematic data.

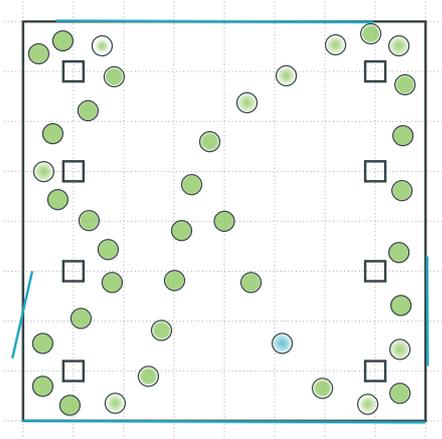


Fig. 9: The Pillars environment scanned with the parameters $P = 300, \sigma = 10$ and with only 80 units or longer line segments allowed in the registration process. Weighting is enabled in the computation. The circles represent the reliability - black border corresponds to $R = 1$, while the filled area corresponds to the actual reliability values. The blur stems from the fusion of results of 100 experiments. A single measurement with lower reliability is depicted in blue.

Weighting also penalizes short line segments further reducing the reliability coefficient (but making it better corresponding to the given situation). Note that the positions, where low reliability was identified have always highly obstructed field of view, although their close neighbors are perfectly fine. At first, some of these examples might seem counter-intuitive, but this is exactly why the reliability needs to be evaluated - even in a common environment, scans from wrong places can cause misleading results.

4.5. Wahba's Rotation Estimation

Let us start the experimentation with Mitsuta's and Wahba's averaging with a simple thought experiment. Having angles $[0^\circ, 0^\circ, 90^\circ]$, an intuitive mean is 30° , which is also what we get using the Mitsuta's algorithm Alg. 1. The operation suits the orientation estimation very well because only the rotation is in question and corresponds to the linearity of angle measurement in a $[-180^\circ, 180^\circ]$ window. Wahba's averaging Eq. (15) gives approximately 26.56° , which is noticeably different and describes a situation, when we move two steps in $[0^\circ]$ direction and one in $[90^\circ]$ direction. The resulting angle than corresponds to the orientation, which will bring us straight to the end of such trajectory. Both methods have meaningful applications, but Mitsuta's averaging cannot be generalized to 3D, so even in 2D case, when consistency with 3D is required, the Wahba's method has to be used.

Since both methods provide different results, an analysis of the influence of the input data on this dif-

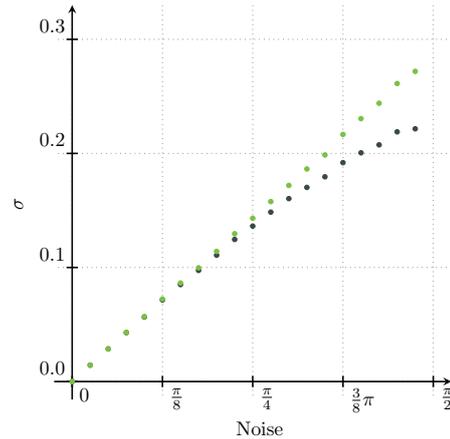


Fig. 10: The plot shows standard deviation of mean estimates from the true value. Mitsuta's method (Alg. 1, green) exhibits linear dependence of the deviation on the noise in the data, while the Wahba's method [Eq. (15), black] is more permissive to large errors in data, which leads to lower deviation on highly noised data.

ference is in place. We have prepared data sets, where one of the corresponding line segments was always rotated by a random angle (generated with uniform distribution in an interval centered at zero). For each averaging procedure, we have recorded the mean value and repeated the computation with different random variables. After a large number of repetitions, the distribution of the means turned out to correspond to the normal distribution centered at zero with certain standard deviation σ . These deviations for both methods and various levels of maximal noise are plotted in Fig. 10. The diagram proves our theoretical expectations because the Mitsuta's algorithm produces a deviation linearly dependent on noise, which is a consequence of direct computation with angular values. The Wahba's method is more permissive to large deviations, so the σ is lower for highly noised data. The sinusoidal shape of the characteristics directly stems from the original Wahba's problem statement Eq. (10) - the distance between vector tips correspond to the half of the angle they define. In practice, such large noise levels, where the difference is significant, are uncommon, so the overall outcome of this experiment is positive: For practical averaging of small angles, both methods are equivalent.

5. Conclusion

This paper deals with the registration of two sets of line segments with known correspondences. It is computationally efficient because its time complexity is only proportional to the number of corresponding pairs being involved. We also present two ways of orientation computation, one linear on angular data and sec-

ond compatible with state of the art 3D pose estimation methods. The criterion for optimal registration is transparent and can be used to define a metric of similarity of given data sets. The method also provides a mechanism for evaluation of reliability of the results, which allows identifying degenerate data sets, which cannot be registered with certainty. All properties of the registration process were systematically tested to prove and illustrate the theoretical results.

A significant drawback is the initial necessity of data association. We are well aware of this issue and our research effort is now focused on a procedure using the described registration method to examine possible correspondences and isolate a set of those, which exhibit similarity after the registration. Because of computational efficiency and the direct measure of similarity, this approach seems to be a perspective way to deal with the data association problem.

Acknowledgment

The research was supported by ECSEL JU under the project H2020 737469 AutoDrive - Advancing fail-aware, fail-safe, and fail-operational electronic components, systems, and architectures for fully automated driving to make future mobility safer, affordable, and end-user acceptable. This research has been financially supported by the Ministry of Education, Youth and Sports of the Czech republic under the project CEITEC 2020 (LQ1601).

References

- [1] KOCMANOVA, P., L. ZALUD and A. CHROMY. 3D proximity laser scanner calibration. In: *18th International Conference on Methods & Models in Automation & Robotics (MMAR)*. Miedzyzdroje: IEEE, 2013, pp. 742–747. ISBN 978-1-4673-5508-7. DOI: 10.1109/MMAR.2013.6670005.
- [2] BURIAN, F., P. KOCMANOVA and L. ZALUD. Robot mapping with range camera, CCD cameras and thermal imagers. In: *19th International Conference on Methods and Models in Automation and Robotics (MMAR)*. Miedzyzdroje: IEEE, 2014, pp. 200–205. ISBN 978-1-4799-5081-2. DOI: 10.1109/MMAR.2014.6957351.
- [3] CADENA, C., L. CARLONE, H. CARRILLO, Y. LATIF, D. SCARAMUZZA, J. NEIRA, I. REID and J. J. LEONARD. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*. 2016, vol. 32, iss. 6, pp. 1309–1332. ISSN 1552-3098. DOI: 10.1109/TRO.2016.2624754.
- [4] HOFSTADTER, D. R. *Godel, Escher, Bach: An Eternal Golden Braid*. 1st ed. New York: Basic Books, 1979. ISBN 978-04650-2685-2.
- [5] THRUN, S. *Robotic Mapping: A Survey*. 1st ed. San Francisco: Morgan Kaufmann Publishers, 2003. ISBN 1-55860-811-7.
- [6] THRUN S. W. BURGARD and D. FOX. *Probabilistic Robotics*. 1st ed. The Cambridge: MIT Press, 2005. ISBN 978-02622-0162-9.
- [7] DISSANAYAKE, G., S. HUANG, Z. WANG and R. RANASINGHE. A review of recent developments in Simultaneous Localization and Mapping. In: *6th International Conference on Industrial and Information Systems*. Kandy: IEEE, 2011, pp. 477–482. ISBN 978-1-4577-0032-3. DOI: 10.1109/ICIINF.2011.6038117.
- [8] BESL, P. J. and N. D. MCKAY. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992, vol. 14, iss. 2, pp. 239–256. ISSN 0162-8828. DOI: 10.1109/34.121791.
- [9] POMERLEAU, F., F. COLAS and R. SIEWART. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics*. 2015, vol. 4, iss. 1, pp. 1–104. ISSN 1935-8261. DOI: 10.1561/23000000035.
- [10] SEGAL, A., D. HAEHNEL and S. THRUN. Generalized-ICP. In: *Robotics: Science and Systems V*. Seattle: The MIT Press, pp. 1–8. 2009. ISBN 978-02625-1463-7. DOI: 10.15607/RSS.2009.V.021.
- [11] TAM, G. K. L., Z. CHENG, Y. LAI, F. C. LANGBEIN, Y. LIU, D. MARSHALL, R. R. MARTIN, X. F. SUN and P. L. ROSIN. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics*. 2012, vol. 19, iss. 7, pp. 1199–1217. ISSN 1941-0506. DOI: 10.1109/TVCG.2012.310.
- [12] BELLEKENS, B., V. SPRUYT, R. BERKVEN, R. PENNE and M. WEYN. A Benchmark Survey of Rigid 3D Point Cloud Registration Algorithms. *International Journal on Advances in Intelligent Systems*. 2015, vol. 8, no. 1, pp. 118–227. ISSN 1942-2679.
- [13] LIU, S. and X. XIE. Research on algorithm of point cloud MapReduce registration. In: *IEEE International Conference on Cloud Computing and Intelligence Systems*. Beijing: IEEE,

- 2011, pp. 338–341. ISBN 978-1-61284-203-5. DOI: 10.1109/CCIS.2011.6045086.
- [14] JI, S., Y. REN, Z. JI, X. LIU and G. HONG. An improved method for registration of point cloud. *Optik*. 2017, vol. 140, iss. 1, pp. 451–458. ISSN 0030-4026. DOI: 10.1016/j.ijleo.2017.01.041.
- [15] FISCHLER, M. A. and R. C. BOLLES. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. 1981, vol. 24, iss. 6, pp. 381–395. ISSN 0001-0782. DOI: 10.1145/358669.358692.
- [16] LOWE, D. G. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Kerkyra: IEEE, 1999, pp. 1150–1157. ISBN 0-7695-0164-8. DOI: 10.1109/ICCV.1999.790410.
- [17] BAY, H., A. ESS, T. TUYTELAARS and L. VAN GOOL. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*. 2008, vol. 110, iss. 3, pp. 346–359. ISSN 1077-3142. DOI: 10.1016/j.cviu.2007.09.014.
- [18] ZHANG, Z. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*. 1994, vol. 13, iss. 2, pp. 119–152. ISSN 0920-5691. DOI: 10.1007/BF01427149.
- [19] ALSHAWA, M. ICL, Iterative Closest Line: A novel point cloud registration algorithm based on linear features. *Ekscentar*. 2007, no. 10, pp. 53–59. ISSN 1331-4939.
- [20] LU, F. *Shape Registration for Using Optimization for Mobile Robot Navigation*. Toronto, 1995. Dissertation. University of Toronto. Supervisor Evangelos Milios.
- [21] KWON, Y. P. *Line Segment-based Aerial Image Registration*. Berkeley, 2014. Master's thesis. Electrical Engineering and Computer Sciences University. Supervisor Sara McMains.
- [22] MITRA, N. J., N. GELFAND, H. POTTMANN and L. GUIBAS. Registration of point cloud data from a geometric optimization perspective. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. Nice: ACM Press, 2004, pp. 22–31. ISBN 3-905673-13-4. DOI: 10.1145/1057432.1057435.
- [23] JELINEK, A., L. ZALUD and T. JILEK. Fast total least squares vectorization. *Journal of Real-Time Image Processing*. 2019, vol. 16, iss. 2, pp. 459–475. ISSN 1861-8200. DOI: 10.1007/s11554-016-0562-6.
- [24] MORI, Y. Evaluation of Several "Single-Pass" Estimators of the Mean and the Standard Deviation of Wind Direction. *Journal of Climate and Applied Meteorology*. 1986, vol. 25, no. 10, pp. 1387–1397. ISSN 0733-3021. DOI: 10.1175/1520-0450(1986)025<1387:EOSPEO>2.0.CO;2.
- [25] *Meteorological Monitoring Guidance for Regulatory Modeling Applications*. Durham: U.S. Environmental Protection Agency. 2000. Available at: <https://www3.epa.gov/scram001/guidance/met/mmgrma.pdf>
- [26] WAHBA, G. A Least Squares Estimate of Satellite Attitude. *SIAM Review*. 1965, vol. 7, iss. 3, pp. 409–409. ISSN 0036-1445. DOI: 10.1137/1007077.
- [27] MARKLEY, F. L. and D. MORTARI. How to estimate attitude from vector observations. *Advances in the Astronautical Sciences*. 2000, vol. 103, iss. 1, pp. 1979–1996. ISSN 0065-3438.

About Authors

Ales JELINEK was born in Hradec Kralove, Czech Republic. He received his Ph.D. from Brno University of Technology in 2017. His research interests include mobile robotics and autonomous vehicles with special attention on surround sensing and representation. His current work is split between high-throughput data processing and vector representation of environment geometry.

Adam LIGOCKI was born in Karvina, Czech Republic. He received his M.Sc. from Brno University of Technology in 2017. His research interests include dynamic 3D mapping techniques for autonomous robot agents, fail-safety of self driving vehicles and application of artificial intelligence in this fields. His work is currently aiming on high bandwidth data acquisition and preprocessing.