Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Robotics and Autonomous Systems 44 (2003) 29–40

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Robot localization based on scan-matching—estimating the covariance matrix for the IDC algorithm

Ola Bengtsson*, Albert-Jan Baerveldt

Halmstad University, Halmstad, Sweden

Abstract

We have previously presented a new scan-matching algorithm based on the IDC (iterative dual correspondence) algorithm, which showed a good localization performance even in environments with severe changes. The problem of the IDC algorithm is that there is no good way to estimate a covariance matrix of the position estimate, which prohibits an effective fusion with other position estimates of other sensors. This paper presents two new ways to estimate the covariance matrix. The first estimates the covariance matrix from the Hessian matrix of the error function minimized by the scan-matching algorithm. The second one, which is an off-line method, estimates the covariance matrix of a specific scan, from a specific position by simulating and matching scans around the position. Simulation results show that the covariance matrix provided by the off-line method fully corresponds with the real one. Some preliminary tests on real data indicate that the off-line method gives a good quality value of a specific scan position, which is of great value in map building.

Keywords: Mobile robots; Self-localization; Scan-matching; Covariance matrix

1. Introduction

Autonomous mobile robots have a high potential for future use, as the costs of labor will increase and the cost of mobile robots will probably decrease. For any task, the mobile robot must know where it is and how it can move to another position in the environment while continuously keeping track of its position. The unbounded error from the use of dead reckoning must be eliminated so that the robot's estimate of its position becomes correct. The areas in which most robots will work will be subject to significant changes, e.g., human beings in motion, other robots and doors that are opening and closing. It is therefore crucial that the robot localization system is robust towards changes in environments. Autonomous robots that will work in an industrial environment must be able to find their location in any kind of environment: changing or non-changing, and rectilinear as well as non-rectilinear. One good way among other methods [1] is to match range scans, made by, e.g. a laser range finder, taken at different locations at different times in the environment and to update the position estimate according to the match result. Such a scan-matching algorithm must handle the following three demands:

1. Exclude scan points from changed areas in the environment or otherwise the result will be incorrect.
2. Match in any kind of environment, i.e. both rectilinear and non-rectilinear environments.
3. Provide some information about the uncertainty of the estimated position, preferably in the form of a covariance matrix.

* Corresponding author.
E-mail addresses: ola.bengtsson@ide.hh.se (O. Bengtsson), albert-jan.baerveldt@ide.hh.se (A.-J. Baerveldt).

0921-8890/03/$ – see front matter
doi:10.1016/S0921-8890(03)00008-3

Most existing scan-matching algorithms work well in environments that are static but poorly in dynamic environments, i.e. changing environments.

Some scan-matching algorithms work well only in specific types of environments, e.g. rectilinear environments, and function poorly in others, e.g. the Cross Correlation Algorithm [2] and the Cox Algorithm [3].

The IDC (iterative dual correspondence) [4,5] algorithm matches two scans based on a point-to-point correspondence, which gives the algorithm a robust performance in both rectilinear and non-rectilinear environments.

In [6,7] we presented a new variant of the IDC algorithm, IDC-S (IDC-Sector), which shows significantly better performance when major changes in the environment occur. The problem with both IDC and IDC-S is that the algorithms sometimes judge their matching results too optimistically, i.e. indicating that the match results are good when they are actually poor, or partially poor. Besides being able to match in any kind of environment, it is therefore crucial that the algorithm can judge its own results, especially when the results are to be fused with other sensor information, e.g. by means of a Kalman filter. If the estimated covariance is too small, many features observed in the environment are rejected as they are farther away than expected [12]. The opposite, i.e. the covariance is too large, will pose an ambiguity problem, as currently extracted features in the environment may match several of the reference features. In the scan-matching case, this means that the scan taken at the actual robot position could match several scans in the reference map. In that case, a more sophisticated method, such as Multi-hypothesis Localization and Tracking [13], may be used to maintain more than one possible robot position.

In [5] it is shown how a covariance matrix for the IDC algorithm can be estimated directly from the corresponding pair of points. This method gives a good value of the quality of the matching process, i.e. how well the scans overlap, but does not say anything about the actual result calculated by the scan-matching algorithm. This means that it is often too *optimistic* and therefore not reliable in, e.g. sensor fusion. We are interested in a covariance matrix that reflects the results of the scan-matching process rather than a covariance matrix that reflects the matching process.

In [8] we presented a new way to estimate the covariance matrix for the IDC algorithm. This covariance is estimated from the Hessian matrix of the error function that is minimized by the scan-matching algorithm. Preliminary tests showed some good results but, in further investigation, the estimated covariance seems to be too *pessimistic*, i.e. classifying good match results as bad. The advantage of the approach, however is that the estimated covariance captures the shape, i.e. the correlation, of the real covariance.

In this paper we present another, off-line, approach to estimate the covariance matrix. The idea is to estimate the covariance for a specific scan by building a geometric map from the scan itself and simulating matches from positions surrounding the scan position. The estimated covariance can then be used either as a quality value of a specific scan position or as the expected covariance for future matches to the scan.

The paper is organized as follows. Section 2 briefly discusses the scan-matching algorithm, IDC. Section 3 presents how the covariance matrix can be estimated from the Hessian matrix [8] and Section 4 the way in which the covariance matrix can be estimated off-line. Section 5 gives some results and Section 6 concludes the paper.

2. Scan-matching

The purpose of scan-matching when used for robot localization is to find the relative distance and rotation between a reference position and the actual position of the robot by comparing one scan taken at the reference position and one scan taken at the robots actual position. It is assumed that the actual position is known approximately from, e.g. dead reckoning, which limits the search space of the scan-matching algorithm. The scan-matching algorithm then translates and rotates ($\mathrm{d}x$, $\mathrm{d}y$, $\mathrm{d}\theta$) the actual scan to make the best overlap of the reference scan. The translation and rotation ($\mathrm{d}x$, $\mathrm{d}y$, $\mathrm{d}\theta$) is, if the match is done correctly, the relative distance and rotation between the reference position and the actual position. The relative distance and rotation are then used to update the position of the robot.

Before the match is done, the scans are filtered with a reduction filter, which smoothes the scan points, and a projection filter, which removes scan points in the

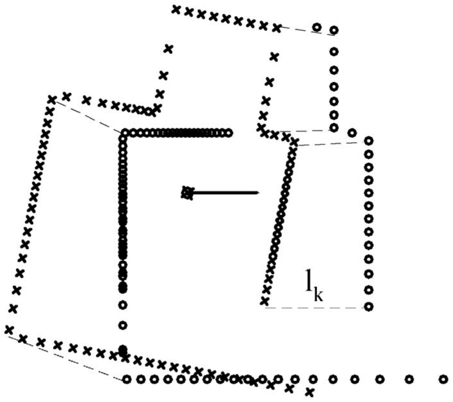

Fig. 1. The reference scan and the actual scan in the same coordinate system. Examples of line segments are shown as dashed lines.

reference scan that are not visible from the actual scan position, and vice versa, see [6,7,9] for more details.

2.1. IDC

IDC (iterative dual correspondence) [4,5] is an iterative scan-matching algorithm consisting of two steps. The first step is to find, for each point in the reference scan, a corresponding point in the actual scan. Corresponding points are points in the reference scan and actual scan that represent the same physical point in the environment. Fig. 1 shows a reference scan (circles) and an actual scan (crosses) taken at two different locations in the same environment. Fig. 1 illustrates with dashed lines some of the corresponding points and the line segments connecting them. The second step minimizes a squared error as a function of translation and rotation. The process of finding corresponding points and minimizing the error is iterative, see [4,5] for more details.

In [6,7] we presented a sector-based variant of the IDC algorithm, referred to as IDC-S (IDC-Sector). The improvement is that IDC-S adjusts well to the amount of changes in the environment and is thus more robust toward dynamic environments. IDC-S performs well even in environments changed as much as 70%, i.e. it can cope with great changes in the environment.

2.2. Drawbacks of IDC

The basic problem in IDC lies in the heuristic way in which it finds the corresponding pairs of points. This will be illustrated by the following example.

Assume in Fig. 2 that circles represent scan points of a reference scan (r_1–r_{12}) taken in a long corridor and that crosses represent scan points of an actual scan (a_1–a_6). Further, the vertical lines represent the corridor walls, the circle with a cross between the walls represents the reference scan position and the square represents the actual position. Also assume that the correct correspondence of points should be (r_4–a_1), (r_5–a_2), (r_6–a_3), (r_7–a_4), (r_8–a_5) and (r_9–a_6). If the actual position is translated along the *x*-axis (Fig. 2, left side), the resulting error will be large because the line segments are becoming longer, which means that IDC can adjust the actual scan and find the true translation between the actual position and the reference position. The same happens when the actual scan is rotated relative to the reference scan (Fig. 2, middle), i.e. the resulting error will be large and IDC can adjust the actual

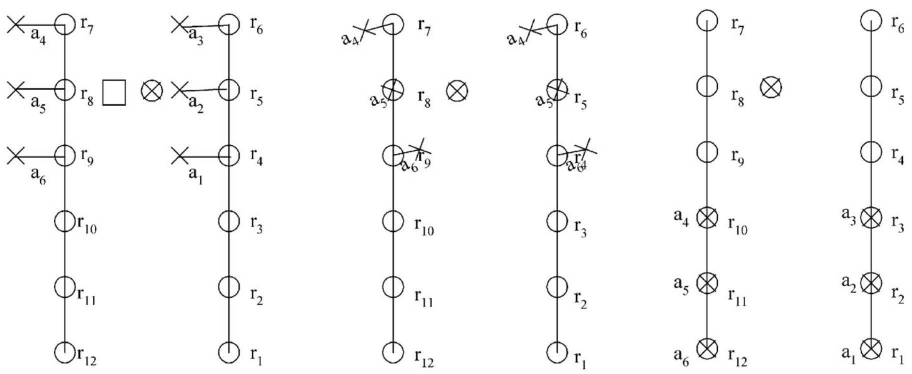

Fig. 2. Line segments between corresponding points in a long corridor.

scan and find the relative rotation between the reference position and the actual position. If the actual scan is translated along the y-axis (Fig. 2, right side), the corresponding pairs of points become (r_1–a_1), (r_2–a_2), (r_3–a_3), (r_{10}–a_4), (r_{11}–a_5) and (r_{12}–a_6), i.e. there is an incorrect correspondence of points. This means the resulting error will be small, i.e. IDC wrongly believes that the actual position is its current position.

In [5] Lu shows how a covariance matrix can be estimated directly from the corresponding pair of points. The estimation is based on the assumption that the algorithm always finds the same physical point in the reference scan and the actual scan. This assumption is violated, e.g. in a long corridor where the corresponding pair of points does not necessarily represent the same physical point in the two scans, because the scans can be taken at any location along the corridor. The result is that the covariance matrix is much too optimistic in the direction along the corridor.

To fuse scan-matching information with information from other sensors, it is important to have a covariance matrix that shows when and in what directions the results can be trusted. This is even more important for IDC-S, especially in the case of severe changes where the matching result is based on only one or two sectors. The rather small sectors often only have a sufficient number of points to determine an accurate position estimate in one direction, i.e. either x, y or θ.

3. Hessian matrix—covariance matrix

An error function is defined as a part of the scan-matching algorithm. This is a non-linear function of the displacement (x, y, θ), which is minimized in a least-squares sense. If the error function can be linearized, it is possible to use the known theory of linear regression [10,11] and derive the covariance matrix analytically. For this reason we first, briefly repeat the theory of linear regression, which is needed in our case, and then apply this to the IDC scan-matching algorithm. In the linear case, the true model function, Y, and the estimated model function, $\hat{Y}$, can be expressed as

$$Y = MX + w \quad \text{and} \quad \hat{Y} = M\hat{X}. \tag{3.1}$$

In Eq. (3.1), Y contains the responses, M is the observation matrix, X contains the model parameters, i.e. in our case (x, y, θ) and w is white noise, i.e. zero mean and known variance. The error to be minimized is

$$E(\hat{X}) = (Y - M\hat{X})^{\mathrm{T}}(Y - M\hat{X}). \tag{3.2}$$

This function has an optimal estimate [10] of the parameter vector, $\hat{X}$, and the covariance matrix, $C(\hat{X})$, as

$$\hat{X} = (M^{\mathrm{T}}M)^{-1}M^{\mathrm{T}}Y, \tag{3.3}$$

$$C(\hat{X}) = (M^{\mathrm{T}}M)^{-1}\sigma^2. \tag{3.4}$$

Eq. (3.5) provides an unbiased estimate of σ^2 where n is the number observations and k the number of parameters to be estimated, see [11] for more details:

$$s^2 = \frac{E_{\mathrm{MIN}}(\hat{X})}{n-k}. \tag{3.5}$$

A double differentiation of Eq. (3.2) gives the Hessian matrix, H, as

$$H = \frac{\mathrm{d}E^2(\hat{X})}{\mathrm{d}\hat{X}^2} = 2M^{\mathrm{T}}M \Rightarrow M^{\mathrm{T}}M = \frac{1}{2}H. \tag{3.6}$$

Combining Eqs. (3.4) and (3.6), $C(\hat{X})$ becomes

$$C(\hat{X}) = (\tfrac{1}{2}H)^{-1}\sigma^2. \tag{3.7}$$

The following estimation is performed in the scan-matching case. By translating ($T = (x, y)^{\mathrm{T}}$) and rotating (θ) one of the scans, IDC minimizes the square error function, $E_{\mathrm{FIT}}(T, \theta)$, which is a non-linear function

$$E_{\mathrm{FIT}}(T, \theta) = \sum_{i=1}^{N}[R(\theta)P_i + T - P_i']^2. \tag{3.8}$$

In Eq. (3.8), $P_i = (x_i, y_i)^{\mathrm{T}}$ and $P_i' = (x_i', y_i')^{\mathrm{T}}$ are the ith corresponding pair of points found by the IDC algorithm, where P_i is the point expressed in reference scan coordinates and P_i' the same point expressed in actual scan coordinates. Eq. (3.8) can be linearized for small θ:

$$\begin{aligned} &\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}\begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_i' \\ y_i' \end{pmatrix} \\ &\approx \begin{pmatrix} 1 & -\theta \\ \theta & 1 \end{pmatrix}\begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &\Rightarrow \begin{pmatrix} x_i' - x_i \\ y_i' - y_i \end{pmatrix} = \begin{pmatrix} 1 & 0 & -y_i \\ 0 & 1 & x_i \end{pmatrix}\begin{pmatrix} x \\ y \\ \theta \end{pmatrix}. \end{aligned} \tag{3.9}$$

The above equation is linear and of the same form as Eq. (3.1), which means that the optimal estimate of the covariance matrix can be calculated according to Eqs. (3.4) and (3.5). Lu and Milios do this in [4,5], but the calculation unfortunately only holds under the assumption that the algorithm always finds the same physical point in the reference scan and the actual scan, which is not always the case. (This is discussed in detail in Section 2.) To overcome this problem, a sound idea might be to examine the actual error function by first estimating the Hessian matrix and then calculating the covariance matrix according to Eq. (3.7).

Once the scan-matching algorithm has found the minimum, the elements in the Hessian matrix, H, of the error function can be estimated. This is done by first translating and rotating the actual scan and then letting the algorithm once again find a corresponding pair of points, which gives the covariance matrix, C, as

$$C(\hat{X}) = s^2(\tfrac{1}{2}H)^{-1}. \tag{3.10}$$

In Eq. (3.10), s^2 is the unbiased estimate of σ^2, which is calculated according to Eq. (3.5), see [8] for further details. Henceforth we refer to this method as the Hessian method.

4. Off-line calculation of the covariance matrix

The covariance matrix for future matches with a specific scan can be calculated, off-line, once and for all from the scan itself. This can be done, e.g. when a reference scan is stored into a map. Once the covariance matrix is calculated, all future matches with the specific scan will have the calculated covariance matrix. Of course this holds only if the environment in which the scan is taken remains unchanged or changes only slightly.

The covariance matrix for the scan is calculated by building a local geometric map of the environment on the basis of the scan. (Fig. 3 shows, with a dashed line, the map built from the scan taken in position P_0.) The geometric map is then used to create simulated scans from randomly chosen positions around the scan (illustrated as a dark area around P_0 in Fig. 3). To summarize, the following is done at position P_0 in environment E:

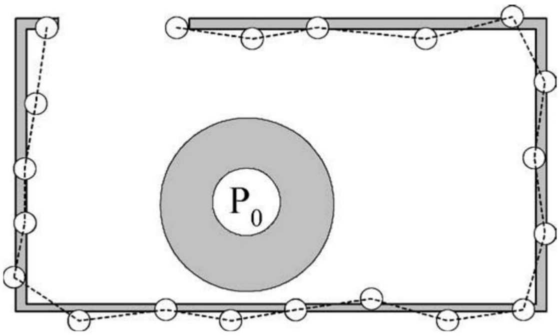

Fig. 3. Environment in which a scan is taken and the local geometric map created from it.

1. Read the first scan, S_0, from the environment, E, which gives a scan consisting of N points.
2. Create a geometric map from S_0 by connecting each consecutive two points with a line, which becomes a part of a wall in the geometric map, G. *Do not* connect two points if the distance between them is longer than some threshold, in our case 1 m. In total, the map will, at most, consist of N lines.
3. Repeat step 4 a 100 times.
4. Randomly choose a position, P_1, within G according to an error distribution. Create a new scan, S_1, from that position, match S_1 to S_0 and store the remaining errors (ex, ey, $e\theta$).
5. Calculate the covariance matrix on the basis of the matching errors.

The size of the estimated covariance matrix in the directions in which matching is not possible varies with the distribution of the simulated position error but does not affect the directions in which matching is possible. Fig. 4 shows the matching error ellipses (X–Y) for matches done in a long corridor with an error distribution of two different sizes, where the dashed line has standard deviations (0.35, 0.35 m, 7.5°) and the solid line has standard deviations (0.15, 0.15 m, 3.5°). The algorithm is unable to match in the y direction. As shown in Fig. 4, only the error in y is affected by the size of the error distribution. Please observe that the scale on the x-axis is 1000 times smaller than that on the y-axis. The rotation is not shown in the figure, but the same reasoning holds here which means that in the corridor case, it is not affected by the error distribution.

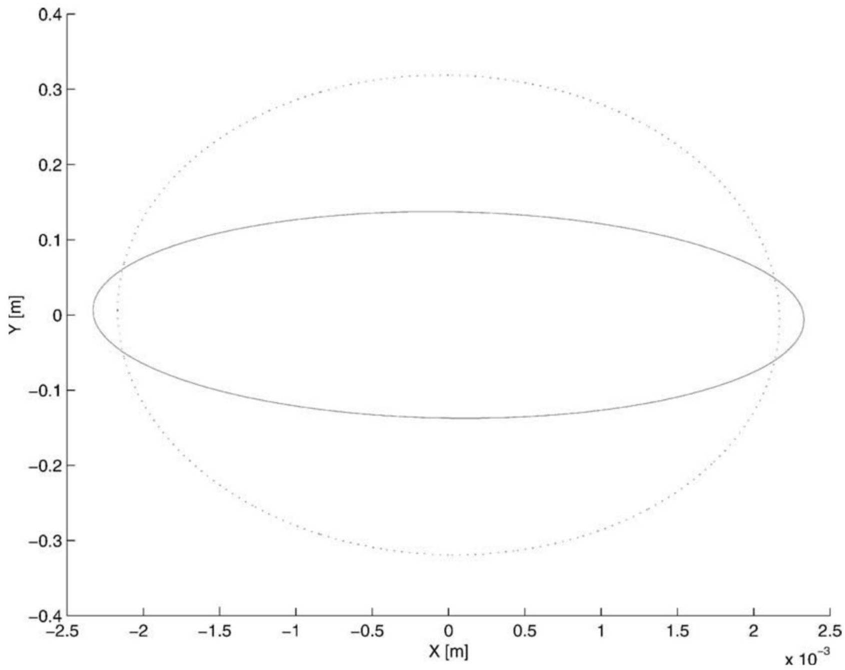

Fig. 4. Constant mahalanobis distance for X and Y of one standard deviation ($1\sigma_x$, $1\sigma_y$).

Henceforth we refer to this method as the off-line method.

5. Tests

The purpose of tests 5.1–5.4 is to verify how well the estimated covariance matrices capture the real one in different environments. The purpose of test 5.5 is to get an indication of whether the off-line method works as it is intended.

The following is true for all simulations. The noise in the sensor readings is normally distributed with a zero mean and a standard deviation of 0.03 m. All position errors (x, y, θ) that are to be estimated by the scan-matching algorithm are normally distributed with a zero mean and standard deviations (0.35, 0.35 m, 7.5°). In Figs. 6, 8, 10, 12 and 13, all illustrating covariance matrices for different cases, the Hessian method is shown as a dashed line, the Lu method as a solid dot or as a ring, and the off-line method and the real as solid lines. All simulations are done in the following way:

1. Take the reference scan, R, in the positions shown in Figs. 5, 7, 9 and 11.
2. Create a local geometric map from the reference scan.
3. Repeat steps 4–6 a 100 times.
4. Choose the actual scan position randomly within the position error distribution and take the actual scan, S, at this position. Match R to S and store the remaining errors.
5. Estimate the covariance matrices according to the Hessian method explained in Section 3 and

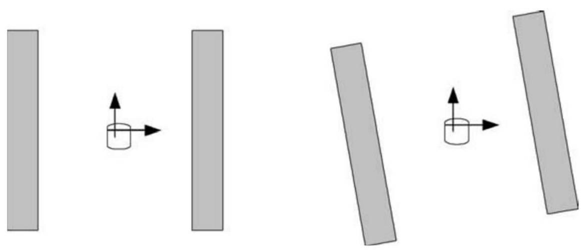

Fig. 5. Simulated corridor environment. The ends are not visible to the sensor.

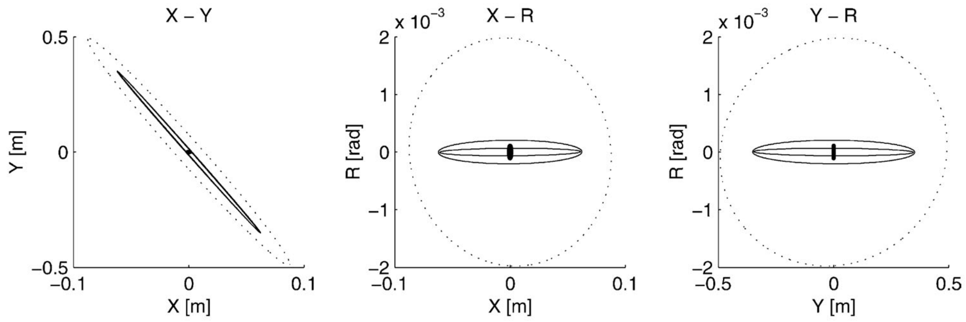

Fig. 6. Matches are done as shown in Fig. 5 (right side).

according to Lu and Milios's method explained in [4,5].

6. Randomly pick a new position error and simulate a scan, R', at this position from the local geometric map. Match R to R' and store the remaining errors.
7. Calculate the average covariance matrix from the results above and compare them with the true covariance matrix, which is based on the actual outcome of all 100 scan matches.
8. Calculate the covariance matrix according to the off-line method, which is based on the outcome of all 100 R to R' matches and compare it to the true covariance.

5.1. Corridor environments

When the robot stands perpendicular to the walls, none of the parameters (x, y, θ) are correlated (Fig. 5, left side). The matching algorithm can only correct the errors in x and θ, and not in y, and for this reason the variance in y should theoretically be infinite. The Hessian method and the off-line method capture this well and show a large uncertainty in y. The Lu method shows, for reasons we have discussed in Section 2.2, the same uncertainty for both x and y. The Hessian method is somewhat too pessimistic. The standard deviations for x, y and θ are summarized below in this section.

When the robot's coordinate system is rotated 10° relative to the walls, x and y are correlated (Fig. 5, right side). There is still no correlation with θ. This case is also captured well by the Hessian and the off-line methods (Fig. 6 shows a graphic illustration of the covariance matrices for this case), although the Hessian method is somewhat too pessimistic. The standard deviations for x, y and θ are summarized below in this section.

Corridor environments	σ_x (mm)	σ_y (mm)	σ_θ (°)
Lu (Fig. 5, left side)	1.1	1.1	0.006
Hessian method	12.9	428.6	0.109
Off-line method	2.4	309.2	0.011
True error	2.3	338.2	0.005
Lu (Fig. 5, right side)	1.2	1.2	0.007
Hessian method	87.8	492.5	0.113
Off-line method	62.3	351.5	0.012
True error	61.2	347.2	0.004

5.2. Circular environments

When the robot stands in the center of a circular environment, none of the parameters (x, y, θ) are correlated (Fig. 7, left side). However, the matching algorithm, can only correct the errors in x and y, and not in θ, as the variance in θ should theoretically be infinite. Both the Hessian and the off-line method capture this well and show a large uncertainty in θ. The Lu method shows almost the same uncertainty in θ in this case as in the corridor case, which occurs for the same

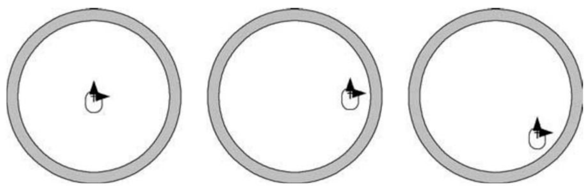

Fig. 7. Simulated circular environment, radius = 5 m.

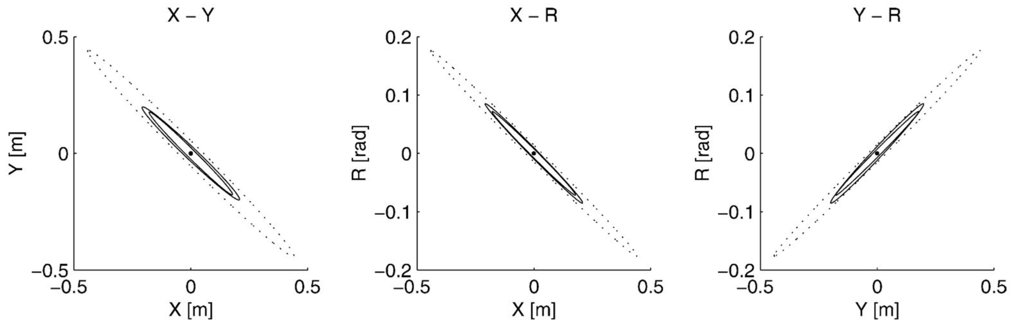

Fig. 8. Matches are done as shown in Fig. 7 (right side).

reasons as in the corridor case. The Hessian method is also somewhat too pessimistic in this environment. The standard deviations for x, y and θ are summarized below in this section.

When the robot moves towards the walls along the x-axis, y and θ get more correlated while x remains uncorrelated to both y and θ (Fig. 7, middle). This case is also captured well by the Hessian and the off-line methods, although somewhat too pessimistically by the Hessian method. The standard deviations for x, y and θ are summarized below in this section.

When the robot moves away from the center, x, y and θ all get more correlated (Fig. 7, right side). In this test, the reference scan is 3.5 m and 45° away from the center. This case is also captured well by the Hessian and the off-line methods, see Fig. 8 for a graphic illustration. The standard deviations for x, y and θ are summarized below in this section.

Circular environments	σ_x (mm)	σ_y (mm)	σ_θ (°)
Lu (Fig. 7, left side)	1.0	1.0	0.012
Hessian method	105.9	79.2	36.872
Off-line method	55.0	59.1	13.284
True error	63.9	59.1	14.577
Lu (Fig. 7, middle)	1.2	1.5	0.014
Hessian method	32.1	576.0	8.091
Off-line method	19.8	288.9	7.191
True error	19.9	276.8	4.067
Lu (Fig. 7, right side)	1.3	1.3	0.014
Hessian method	442.1	441.8	10.103
Off-line method	208.8	199.3	4.884
True error	179.0	177.5	4.115

5.3. Square environments

In the middle of a square environment, there is no correlation between any of the parameters x, y and θ and the scan-matching algorithm can fully determine translation and rotation (Fig. 9, left side). It can also be said that the uncertainty in x and y should be the same, as the robot stands in the middle of a square environment. All of the methods work well in this case as they have approximately the same uncertainty in x and y. The Hessian method is again too pessimistic. The standard deviations for x, y and θ are summarized below in this section.

When the reference position is moved along the y-axis, x and θ gets more correlated, which is exactly the same dependency as was seen in the circular environment except that the correlation is stronger in the circle than in the square environment (Fig. 9, right side). The Hessian and the off-line methods capture this dependency in a correct way, see Fig. 10. Once again, the Hessian method is too pessimistic. The standard deviations for x, y and θ are summarized below in this section.

Square environments	σ_x (mm)	σ_y (mm)	σ_θ (°)
Lu (Fig. 9, left side)	1.1	1.1	0.011
Hessian method	32.0	32.3	0.451
Off-line method	3.7	3.3	0.034
True error	3.9	4.0	0.069
Lu (Fig. 9, right side)	1.7	1.5	0.015
Hessian method	43.1	35.0	0.533
Off-line method	18.4	5.2	0.363
True error	20.3	5.3	0.395

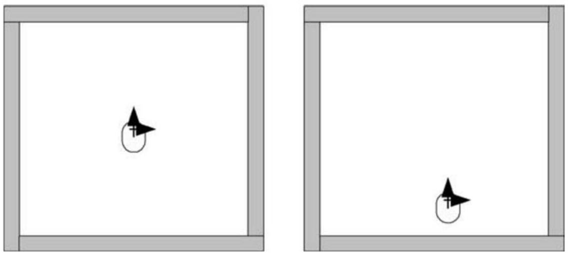

Fig. 9. Simulated square environment, 10 m × 10 m.

5.4. Rectilinear environment/irregular environment

The rectilinear environment (Fig. 11, left side) and the irregular environment (Fig. 11, right side) follow the same reasoning as for environments 5.1–5.3, i.e. depending on the reference position, the parameters x, y and θ become correlated to different extents. The Hessian method is again too pessimistic and the Lu method too optimistic, especially in rotation; see Figs. 12 and 13 for a graphic illustration of the covariance matrices. The standard deviations for the main directions, i.e. x, y and θ, are given below.

	σ_x (mm)	σ_y (mm)	σ_θ (°)
Rectilinear environment			
Lu (Fig. 11, left side)	1.3	1.2	0.016
Hessian method	31.0	30.0	0.503
Off-line method	4.5	4.1	0.177
True error	5.2	4.2	0.206
Irregular environment			
Lu (Fig. 11, right side)	1.3	1.3	0.013
Hessian method	31.6	30.7	0.355
Off-line method	7.1	5.2	0.173
True error	6.3	7.8	0.237

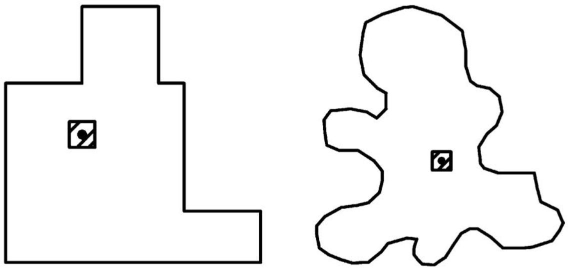

Fig. 11. Simulated rectilinear (10 m × 10 m) and irregular environments (approximately 25 m × 30 m).

5.5. Real data

We have made some preliminary tests with the off-line method described in Section 4 on real sensor data. The data are collected with a LMS200 laser range finder from SICK AG. As we do not know the true position in this test, it is not possible to say whether the estimated covariance matrix is correct, but it does indicate whether the method seems to work.

Fig. 14 shows a map consisting of 18 reference scans. The reference positions are marked as circles and the distance between two reference positions is approximately 1 m. Circles with numbers are nodes whose results are shown in the table below. We estimate a covariance matrix for each reference scan, which directly indicates whether the reference scan is a good reference. The table below gives the standard deviations for the main directions, i.e. x, y and θ, of the selected positions.

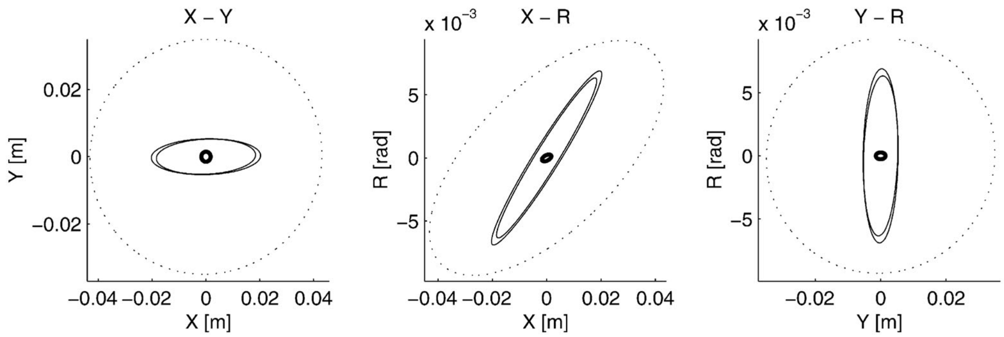

Fig. 10. Matches are done as shown in Fig. 9 (right side).

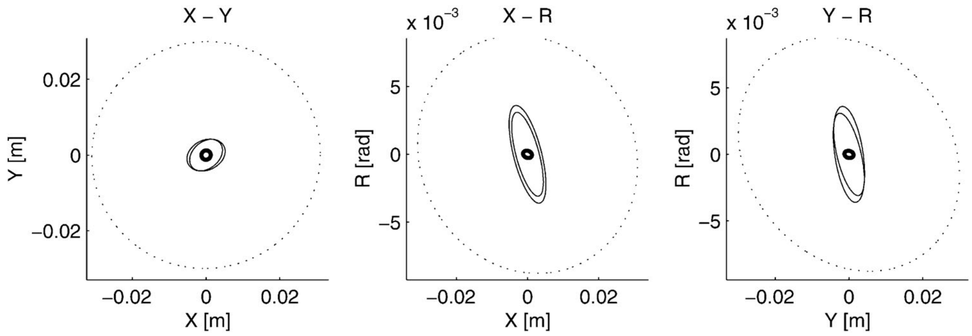

Fig. 12. Matches are done as shown in Fig. 11 (left side).

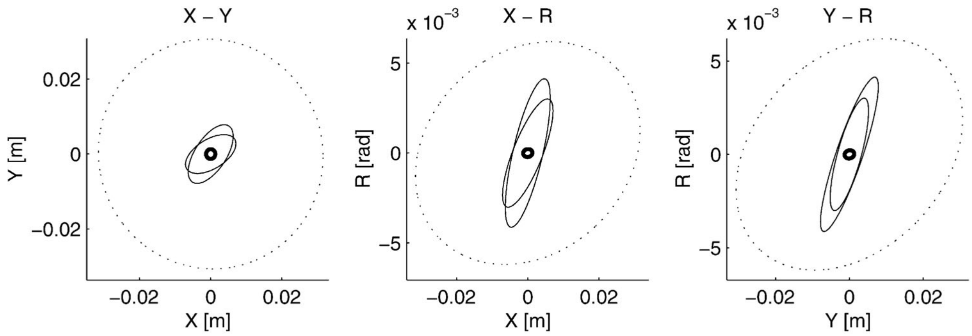

Fig. 13. Matches are done as shown in Fig. 11 (right side).

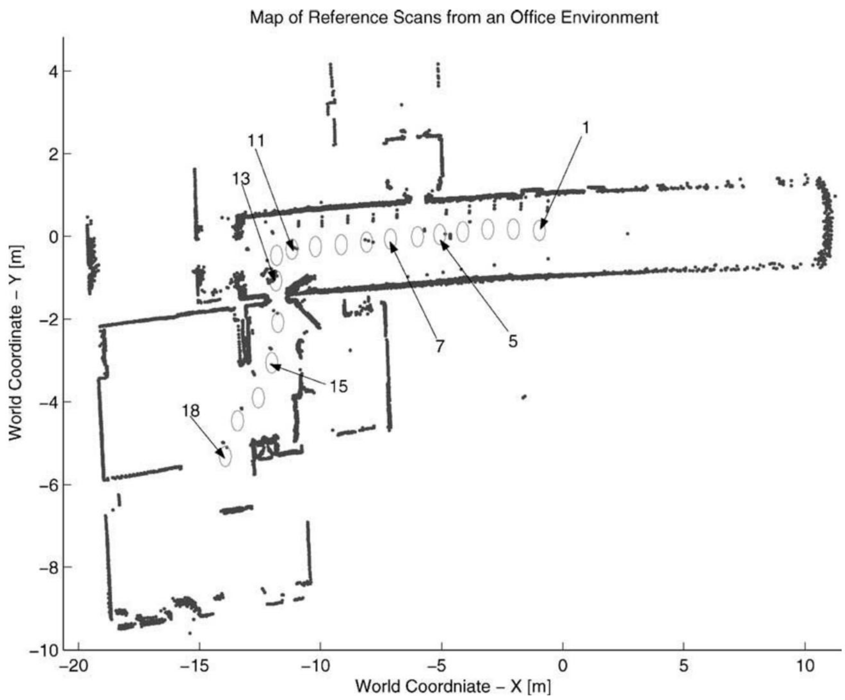

Fig. 14. Real scans from a typical office environment that together build a map.

Real sensor data	σ_x (mm)	σ_y (mm)	σ_θ (°)
Node 1	232.1	6.0	0.055
Node 5	216.3	8.6	0.064
Node 7	135.6	5.9	0.106
Node 11	19.5	3.5	0.166
Node 13	17.2	4.2	0.245
Node 15	5.0	5.5	0.231
Node 18	5.1	5.2	0.227

The results show that the off-line method seems to work in a logical way. The x component of the expected uncertainty is large for node 1 to node 7. This is logical, as the environment offers no, or only little, information to determine the position in the x direction (along the corridor walls). The almost constant uncertainty in y also seems logical as the environment always offers enough information to determine the position in the y direction, which means that any node offers a good match when the position in y will be estimated. This is also true for the rotation, as the environment always offers enough information to determine the rotation.

It is also interesting to note how well the results using real data correlate with the results in the simulated data. The first part (nodes 1–7) of the robots path is a typical corridor environment, and the results should be compared with the results in Section 5.1. The second part (nodes 11–18) is a typical rectilinear environment, or square environment, and the results should thus be compared to those in Sections 5.3 and 5.4.

6. Conclusions and outlook

This paper presented two methods for estimating a covariance matrix that reflects the matching performance of the IDC algorithm. The first method, which was originally presented in [8], is estimated by examining the Hessian matrix of the error function. This method is suitable for on-the-fly estimates of single match results. The second one builds a local geometric map from a specific scan and simulates scan matches to the specific scan within the map. The matching errors are stored, and the covariance matrix is calculated on the basis of these errors. This method is an off-line method and suitable for estimating in advance the expected covariance matrix the next time the scan is used to make a match. This off-line method also gives a good quality value of a specific scan position that can be of great help, e.g. when building a map of reference scans or planning a path along the reference scans.

Both methods are evaluated together with a third method, Lu and Milios [4,5], and compared with the real covariance matrix. The results show that the Hessian method captures the shape of the real covariance well but not the size, that the Lu method is almost constant for all environments, i.e. far too optimistic, and that the off-line method captures both the shape and the size of the real covariance matrix.

In the future we will investigate how the methods work with real sensor data. We will also explore whether it is possible to combine the Hessian and the off-line methods so that the Hessian method also captures the size of the real covariance matrix.

Acknowledgements

We would like to thank NDC Automation AB and Cognex Sweden together with the Swedish foundation “Stiftelsen för Kunskaps-och Kompetensutveckling” (KK-foundation) for funding our research.

References

[1] J.-S. Gutmann, W. Burgard, D. Fox, K. Konolige, An experimental comparison of localization methods, in: Proceedings of the International Conference on Intelligent Robots and Systems (IROS’98), Victoria, Canada, October 1998.

[2] G. Weiss, E.V. Puttkamer, A map based on laser scans without geometric interpretation, in: Proceedings of the Intelligent Autonomous Systems—4 (IAS-4), Karlsruhe, Germany, March 27–30, 1995.

[3] I. Cox, Blanche—an experiment in guidance and navigation of an autonomous robot vehicle, IEEE Transactions on Robotics and Automation 7 (2) (1991) 193–204.

[4] F. Lu, E.E. Milios, Robot pose estimation in unknown environments by matching 2D range scans, Technical Report No. RBCV-TR-94-46, University of Toronto, USA, 1995.

[5] F. Lu, Shape registration using optimization for mobile robot navigation, Ph.D. Thesis, University of Toronto, USA, 1995.

[6] O. Bengtsson, A.-J. Baerveldt, Localization in changing environments by matching laser range scans, in: Proceedings of the Third European Workshop on Advanced Mobile Robots (EURobot’99), Zürich, Switzerland, September 6–8, 1999, pp. 169–176.

[7] O. Bengtsson, A.-J. Baerveldt, Matching of laser range-finder scans in changing environments, in: Proceedings of

the M2VIP-99, Ankara, Turkey, September 1–3, 1999, pp. 103–109.
[8] O. Bengtsson, A.-J. Baerveldt, Localization in changing environments—estimation of covariance matrix for the IDC algorithm, in: Proceedings of the IROS'2001, vol. 4, Outrigger Wailea Resort, Maui, HI, October 29–November 3, pp. 1931–1937.
[9] J.-S. Gutmann, Robuste navigation autonomer mobiler systeme, Ph.D. Thesis, University of Freiburg, DISKI 241, Akademische Verlagsgesellschaft Aka, Berlin, 2000. ISBN 3-89838-241-9.
[10] S.M. Kay, Fundamentals of Statistical Signal Processing—Estimation Theory, Prentice-Hall, Englewood Cliffs, NJ. ISBN 0-13-345711-7, 1993.
[11] G. Barrie Wetherill, Regression Analysis with Applications, Chapman & Hall, London. ISBN 0-412-27490-6, 1986.
[12] S. Scheding, et al., An experiment in autonomous navigation of an underground mining vehicle, IEEE Transactions on Robotics and Automation 15 (1) (February 1999).
[13] K. Arras, et al., Towards feature-based multi-hypothesis localization and tracking, in: Proceedings of the Fourth European Workshop on Advanced Mobile Robots (EURobot'01), Lund, Sweden, September 19–21, 2001, pp. 57–65.

Ola Bengtsson received his M.S. degree in computer systems engineering from the University of Halmstad in 1998. He is currently a Ph.D. student at the School of Information Science, Computer and Electrical Engineering, at Halmstad University in Sweden. His research interests are mechatronics, indoor robotics and autonomous systems based on range measuring sensors especially laser range finders.

Albert-Jan Baerveldt is a Professor of Mechatronic Systems at the School of Information Science, Computer and Electrical Engineering at Halmstad University in Sweden, where he holds the Getinge Chair of Mechatronic Systems. He received his Ph.D. in mechatronics at the Swiss Federal Institute of Technology, Zurich in 1993, where he conducted research mainly in the field of vision-guided robot arms. During this time he also won the first world championship for ping pong playing robots in Hong Kong in 1992. His Ph.D. work on the bin-picking problem was awarded to exhibit the work at the "Research and Development Exhibition" at the Hannover Fair 1992 in Germany, which is one of the largest industrial fairs in the world. In 1994 he held a postdoctoral position at the Institute of Automatic Control at Lund Technical University, Lund, Sweden. He has worked at Halmstad University in Sweden since 1994 and is currently the leader of the Intelligent Systems Lab. His research interests include robotics and computer vision.