A Lie Algebraic Approach for Consistent Pose Registration for General Euclidean Motion*

Motilal Agrawal
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025, USA
agrawal@ai.sri.com

***Abstract*— We study the problem of registering local relative pose estimates to produce a global consistent trajectory of a moving robot. Traditionally, this problem has been studied with a flat world assumption wherein the robot motion has only three degrees of freedom. In this paper, we generalize this for the full six-degrees-of-freedom Euclidean motion. Given relative pose estimates and their covariances, our formulation uses the underlying Lie Algebra of the euclidean motion to compute the absolute poses. Ours is an iterative algorithm that minimizes the sum of Mahalanobis distances by linearizing around the current estimate at each iteration. Our algorithm is fast, does not depend on a good initialization, and can be applied to large sequences in complex outdoor terrains. It can also be applied to fuse uncertain pose information from different available sources including GPS, LADAR, wheel encoders and vision sensing to obtain more accurate odometry. Experimental results using both simulated and real data support our claim.**

***Index Terms*— consistent pose registration, Lie algebra, odometry, special euclidean group, Mahalanobis distance**

I. Introduction

The ability of a mobile robot to localize itself is critical to its autonomous operation and navigation. Consequently, there has been considerable effort on the problem of mobile robot localization and mapping. This problem is known as *simultaneous localization and mapping* (SLAM) and there is a vast amount of literature on this topic (see e.g., [1] for a comprehensive survey). SLAM has been especially succesful in indoor structured environments [2], [3]. For indoor mapping, the world is modeled as planar and the pose of the robot has only three degrees of freedom ($2D$ translation and the yaw). This is known as $2D$ SLAM.

On the other hand, mapping for outdoor environments is still an open research problem. Outdoor mapping is harder, primarily because of two factors. First, outdoor environments are unstructured, and simpler features such as planes and lines can rarely be used. Approaches for outdoor navigation mapping systems therefore generally use range sensors that build a $3D$ model of the environment [4], [5], [6], [7]. The second and perhaps the more fundamental problem of applying SLAM for outdoor navigation is the fact that the planar-world assumption is rarely valid in most outdoor terrains. Therefore, the theoretical framework of $2D$ SLAM is not applicable for navigation over rough, undulating outdoor terrains. Succesful navigation over such terrains requires a euclidean motion model that utilizes the full six degrees of freedom.

In this paper, we present a framework for localization of a robot moving in an unconstrained environment with the full six degrees of freedom. We assume that we are given the relative pose constraints of the robot in motion. The uncertainties in these relative poses are also known to us. Typically, these local pose constraints are computed either from odometry readings between the poses or by matching features in the environment. Wheel encoders augmented with inertial navigation system is an example of one odometry-based technique to compute relative poses over short distances. In computing relative poses from features, the $3D$ features that are visible from both locations are used to get an estimate of the relative pose and the uncertainty. The most commonly used sensors in this method are stereo-vision systems [8], [7] and $3D$ laser range scanners [5]. We are indifferent to the process of obtaining these relative pose constraints and their uncertainties as long as we have a reliable and a robust way of computing these.

Given these local relative pose constraints, our goal is to recover the global motion of the robot by linking these constraints. A straight-forward but error-prone method of estimating this trajectory is to localize based on adjacent poses. In such a formulation, the previous position and the relative motion are used to estimate the current pose. This recursive estimate is bootstrapped by assuming that the initial pose is known in the world frame. This technique, however, has two fundamental limitations

1) Error-prone: Error tends to accumulate over the distance traveled. Small errors early in the motion can have large effects on later position estimates. This results in an overall drift of the odometry and poor localization. Accommodating such systematic correlated errors is key to building maps successfully, and it is also a key complicating factor in robotic mapping.
2) Suboptimal: This method is suboptimal as it does not utilize all the available pose constraints optimally. This is due to the fact that the local pose constraints can also be estimated reliably between all adjacent

*This work was partially supported by U.S. Army project #DAAD19-01-2-0012 under subcontract #9613P with General Dynamics Robotic Systems

robot positions, not just the consecutive positions. As long as there are common visible features between the two locations, we can get an estimate of the relative motion between those locations. Therefore, the technique must take into account all the available local constraints simultaneously and produce the trajectory that best fits these constraints.

In this paper, we will use the Consistent Pose Registration (CPR) framework of Lu and Milios [9] to efficiently use all the redundant local relative pose constraints and estimate a global trajectory. CPR is a batch SLAM method that estimates the global poses such that it is consistent with all the available local pose constraints and their uncertainties. Their approach solved a $2D$ SLAM problem, and the $2D$ pose relations were obtained either by odometry or by matching range-finder data in adjacent frames. The global poses were estimated by minimizing the Mahalanobis distance between the actual and derived pose over this whole network of pose relations. Since this involves inversion of a large matrix, this approach is likely to be computationally expensive and was later extended by Gutmann and Konolige [10] to perform online, incremental pose estimation. The computational properties of CPR for very large environments were studied and presented by Konolige [11].

In applying CPR for pose estimation in the general case of euclidean motion, one of the major challenges is the appropriate representation of the motion. In particular, rotations are noncommutative and highly nonlinear. The space of all euclidean motions is a Lie group and can be represented succinctly using the underlying Lie algebra. We propose to solve the CPR problem for the full euclidean case using the underlying Lie algebra. Ours is an iterative algorithm that minimizes the sum of Mahalanobis distances by linearizing around the current estimate of the pose at each iteration represented in the Lie algebra.

The work of Govindu [12] is very similar to the work described here. However, that approach performs averaging in the Lie algebra. Therefore, Govindu treats all the local pose constraints equally and cannot incorporate the uncertainties in these pose constraints. The approach described in this paper minimizes the Mahalanobis distance and therefore is much more general and has wider applicability. In particular, our approach can be used to fuse uncertain pose information from various sources in which some constraints are more reliable than others.

The rest of the paper is organized as follows. Section II provides a brief introduction to Lie groups. In particular, the geometry of the special euclidean group is described along with its underlying algebra. Our problem formulation is also discussed in this section. The CPR framework of Lu and Milios is described in Section III for the linear case. The nonlinear iterative formulation of CPR is discussed in IV. The algorithm for CPR for euclidean motion is also described. Results on both simulated data and real sequences are discussed in Section V. Section VI concludes this presentation and discusses ongoing and future work.

II. Geometry of the Special Euclidean Group

For readers not familiar with Lie algebra, we will provide a brief introduction. For further details, please refer to the texts [13], [14] for a thorough exposition on this topic.

A. Lie Algebra

A group G is a set with a binary operation, usually written as juxtaposition that satisfies the following properties.

1) The group must be closed under its binary operation.
2) The group must be associative.
3) The group must have a unique identity element.
4) Every element of the group must have a unique inverse.

Lie groups satisfy additional axioms.

1) The set of group elements forms a differentiable manifold.
2) The group operation must be a differentiable manifold.
3) The map from a group element to its inverse must also be differentiable.

It can be easiliy seen that the set of nonsingular $n \times n$ square matrices forms a group where the group product is modeled by matrix multiplication. Furthermore, this group satisfies the additional axioms given above and hence is a Lie group, usually denoted by $GL(n)$ for the general linear group of order n. Intuitively, Lie groups are differentiable manifolds on which we can do calculus. Locally, it is topologically equivalent to the vector space $\mathbb{R}^n$, and the local neighborhood of any group element G can be described by its tangent-space. The tangent-space at the identity element forms its Lie algebra $\mathfrak{g}$. The exponential function maps an element of the Lie algebra to its corresponding group element and the logarithm function is the inverse mapping from the Lie group to the algebra. For elements x, y belonging to the Lie algebra, the exponential function satisfies the property $\exp(x)\exp(y) = \exp(BCH(x, y))$, where $BCH(x, y)$ is the Baker-Campbell-Hausdorff formula [14]. The BCH formula is given by the series

$$BCH(x, y) = x + y + \frac{1}{2}[x, y] + \mathcal{O}(|(x, y)|^3) \quad (1)$$

The bracket in this case is the commutator operation $[x, y] = xy - yx$.

B. Euclidean Motion

In particular, the rotation group on $\mathbb{R}^3$ is a subgroup of $GL(3)$, defined as

$$SO(3) = \left\{R | R \in GL(3), RR^T = I, \det(R) = 1\right\} \quad (2)$$

The Special Euclidean Group in three dimensions, SE(3), represents the euclidean transformation of rotation followed by translation. It is a semidirect product of the Special Orthogonal group (SO(3)) and $\mathbb{R}^3$. Using homogenous coordinates, we can represent SE(3) as follows,

$$SE(3) = \left\{ \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \in GL(4) \middle| R \in SO(3), t \in \mathbb{R}^3 \right\} \quad (3)$$

The action of an element $g \in SE(3)$ on a point $p \in \mathbb{R}^3$ is given by

$$g = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \quad (4)$$

$$g.p = Rp + t \quad (5)$$

The Lie algebra of SE(3) is given by

$$\mathfrak{se}(3) = \left\{ \begin{pmatrix} \hat{\omega} & u \\ 0 & 0 \end{pmatrix} \in GL(4) \middle| \hat{\omega} \in \mathfrak{so}(3), u \in \mathbb{R}^3 \right\} \quad (6)$$

Here $\hat{\omega}$ is the skew symmetric matrix form of the rotation vector $\omega = (\omega_x, \omega_y, \omega_z)^T$ and is an element of the Lie algebra for $SO(3)$.

$$\hat{\omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (7)$$

The exponential map, $\exp : \mathfrak{se}(3) \to SE(3)$, is given by

$$\exp \begin{pmatrix} \hat{\omega} & u \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \exp(\hat{\omega}) & Au \\ 0 & 1 \end{pmatrix} \quad (8)$$

where

$$A = I + \frac{1 - \cos(\|\omega\|)}{\|\omega\|^2} \hat{\omega} + \frac{\|\omega\| - \sin \|\omega\|}{\|\omega\|^3} \hat{\omega}^2 \quad (9)$$

and $\exp(\hat{\omega})$ is given by Rodrigue's formula,

$$\exp(\hat{\omega}) = I + \frac{\sin \|\omega\|}{\|\omega\|} \hat{\omega} + \frac{1 - \cos \|\omega\|}{\|\omega\|^2} \hat{\omega}^2 \quad (10)$$

The logarithm map, $\log : SE(3) \to \mathfrak{se}(3)$, is given by

$$\log \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \log(R) & A^{-1}t \\ 0 & 0 \end{pmatrix} \quad (11)$$

where

$$\log(R) = \frac{\phi}{2\sin(\phi)} (R - R^T) \equiv \hat{\omega} \quad (12)$$

and ϕ satisfies

$$Tr(R) = 1 + 2\cos(\phi), \quad |\phi| < \pi \quad (13)$$

and where

$$A^{-1} = I - \frac{1}{2}\hat{\omega} + \frac{2\sin\|\omega\| - \|\omega\|(1 + \cos\|\omega\|)}{2\|\omega\|^2 \sin\|\omega\|} \hat{\omega}^2 \quad (14)$$

Using equations 11 and 8, an element of the Lie group (R, t) can be mapped directly to an element of its Lie algebra $(\hat{\omega}, u)$ and vice versa. We will assume that we are given the relative pose constraints in the Lie algebra. In addition, by computing the Jacobian matrix of the transformation (either analytically or numerically), it is possible to convert the covariance matrix in one representation to the other. If J is the jacobian of the exponential mapping and $\Sigma_{\mathfrak{se}(3)}$ the covariance matrix in the Lie algebra representation, then the covariance matrix in the Lie group representation is given by

$$\Sigma_{SO(3)} = J \Sigma_{\mathfrak{se}(3)} J^T \quad (15)$$

C. Problem Formulation

For a motion of the robot, let n be the number of poses to be estimated. Let X_i be the pose at the position i, $i = 1, \ldots, n$. In the Lie algebra of $SE(3)$, this pose can be expressed as $X_i \equiv (\omega_i, u_i)$. Each local pose constraint, D_{ij} gives us relative motion of the j^{th} pose with reference to the i^{th} pose. In terms of the Lie algebra,

$$\exp(D_{ij}) = \exp(X_j) \exp(-X_i) \quad (16)$$

$$D_{ij} = BCH(X_j, -X_i) \quad (17)$$

We are also given the covariance matrix C_{ij} for each relative pose measurement, D_{ij}. These relative poses may be obtained through various modalities, each having its own characteristic covariance matrix. For example, D_{ij}^{GPS} may be obtained through GPS measurement with the associated uncertainty C_{ij}^{GPS} and this may be quite different from the relative pose obtained by, say, visual odometry. This covariance matrix may be obtained from the measurement equation that relates the measured variables to the pose coordinates. For example, for a laser range scanner, the measured variables are the range estimates of the points used to get an estimate of the relative pose.

The goal then is to estimate the poses X_i, given these redundant relative pose constraint pairs (D_{ij}, C_{ij}) which may be obtained from different modalities (e.g; GPS, range scans, visual odometry, *a priori* information). We will use the Consistent Pose Registration (CPR) framework to solve for these poses optimally, a brief introduction to which is provided next.

III. Linear Consistent Pose Registration(CPR)

The CPR framework of Lu and Milios [9] produces a global map by fusing uncertain local pose constraints. These relative poses enforce geometric consistency, thereby producing a globally consistent map.

In the linear case, the relative pose D_{ij} is given by

$$D_{ij} = X_j - X_i \quad (18)$$

These measurement equations can be expressed in a matrix form as

$$D = HX \quad (19)$$

where X is the concatenation of all the poses $X_1, \ldots, X_n$ and D is the concatenation of all the relative poses D_{ij}. H is the incidence matrix with only zeros, and for each row a one and a minus one at the appropriate column to form the difference (equation 18) for each relative pose constraint.

Assuming a Gaussian, independent generative model for the relative pose errors with mean $\bar{D}_{ij}$ and covariance C_{ij}, the maximum likelihood pose estimates can be obtained by minimizing the following Mahalanobis distance (where the sum is over all the given relative pose measurements):

$$W = \sum_{(i,j)} \left(D_{ij} - \bar{D}_{ij}\right)^T C_{ij}^{-1} \left(D_{ij} - \bar{D}_{ij}\right) \quad (20)$$

$$= \left(\bar{D} - HX\right)^T C^{-1} \left(\bar{D} - HX\right) \quad (21)$$

where $\bar{D}$ is the concatenation of all the observations $\bar{D}_{ij}$ and C is a block diagonal matrix whose entries are the covariance matrices C_{ij}. When C is taken as the identity matrix, this problem reduces to the standard least squares minimization. The solution $X_{\min}$ that minimizes this W is obtained by solving the linear system

$$\begin{aligned} GX_{\min} &= B \quad (22)\\ G &= H^T C^{-1} H \quad (23)\\ B &= H^T C^{-1} \bar{D} \quad (24)\end{aligned}$$

IV. Nonlinear CPR

Unlike the linear case (equation 22), there does not exist a closed-form expression for $X_{\min}$ in the general case where the relative poses are determined by a nonlinear equation. Therefore, the minimum solution is obtained iteratively by linearizing around the current solution at each iteration. Let f be the pose compounding operation in the nonlinear case. Therefore, the pose measurement equation is given by

$$D_{ij} = f(X_j, -X_i) \quad (25)$$

To a first degree this can be approximated by $D_{ij} \approx X_j - X_i$. Corresponding to a relative pose measurement $\bar{D}_{ij}$, the error in this measurement is given by

$$\Upsilon_{ij} = \bar{D}_{ij} - f(X_j, -X_i) \quad (26)$$

Therefore, the Mahalanobis distance becomes

$$W = \sum_{(i,j)} \Upsilon_{ij}^T C_{ij}^{-1} \Upsilon_{ij} \quad (27)$$

At the n^{th} iteration, let X_i^n and X_j^n be the pose estimates at locations i and j, respectively, and Υ_{ij}^n be the error corresponding to the relative pose constraint between these two poses. For the next iteration, we will have to refine these pose estimates. Let δX_i^n be the change in the i^{th} pose between the successive iterations n and $n+1$. We have

$$X_i^{n+1} = X_i^n + \delta X_i^n \quad (28)$$

Also, the error (Υ_{ij}) between the two iterations can be approximated to a first degree as

$$\Upsilon_{ij}^{n+1} = \Upsilon_{ij}^n - \left(\delta X_j^n - \delta X_i^n\right) \quad (29)$$

This is easily derived through a Taylor series expansion of the right side of equation 26. In matrix form, equation 29 can be written as

$$\Upsilon^{n+1} = \Upsilon^n - H\Delta X \quad (30)$$

where Υ^{n+1} and Υ^n are the concatenation of all Υ_{ij}^{n+1} and Υ_{ij}^n, respectively, ΔX is the concatenation of all the pose differences δX_i^n, and H is the incidence matrix (as in equation 19). Substituting for Υ^{n+1} from this equation into equation 27, the Mahalanobis distance at the $(n+1)^{th}$ iteration can be written in matrix form as

$$W = \left(\Upsilon^n - H\Delta X\right)^T C^{-1} \left(\Upsilon^n - H\Delta X\right) \quad (31)$$

Notice that the form of this equation is identical to the linear case (equation 21), and therefore the Mahalanobis distance is minimized for $\Delta X = \Delta X_{\min}$ given by the linear system

$$\begin{aligned} G\Delta X_{\min} &= B \quad (32)\\ G &= H^T C^{-1} H \quad (33)\\ B &= H^T C^{-1} \Upsilon^n \quad (34)\end{aligned}$$

Once we calculate $\Delta X_{\min}$, each of the pose differences δX_i^n becomes known to us and therefore we can use equation 28 to update the pose estimates at the next iteration. This process is repeated until convergence.

Intuitively, each iteration of the algorithm seeks to find changes in the pose that will lead to reduction in the Mahalanobis distance. This is accomplished by solving a linear CPR problem at each step. Therefore, this algorithm is very much like a gradient descent algorithm and is guaranteed to find only a local minima. In practice, however, the algorithm performs very well and finds solutions that are globally optimal for most situations.

Lie-algebraic CPR

The iterative algorithm for nonlinear CPR described in the previous section can be directly applied to the problem of registering euclidean motions in the Lie algebra. The pose compounding function in $\mathfrak{se}(3)$ is given by the BCH formula, that is, $f \equiv BCH$. The BCH formula for $SE(3)$ has a closed-form expression [14].

The algorithm for performing CPR in the Lie algebra of $SE(3)$ is outlined below

1) At the start of the algorithm compute $G = H^T C^{-1} H$ (equation 33) and $K = H^T C^{-1}$ (equation 34).
2) Start with random initial poses X_i^1.
3) Repeat steps 4-7 until convergence.
4) Determine Υ, where for each constraint between poses i and j, $\Upsilon_{ij} = \bar{D}_{ij} - BCH(X_j, -X_i)$ (equation 26).
5) Determine $B = K\Upsilon$ (equation 34).
6) Solve the linear system $G\Delta X_{\min} = B$ for $\Delta X_{\min}$ (equation 32).
7) Update the poses as $X_i^{n+1} = X_i^n + \delta X_i$ (equation 28), where δX_i is obtained from $\Delta X_{\min}$.

The convergence of the algorithm is determined by looking at the norm of the update vector $\Delta X_{\min}$. In other words, the iterations are carried out as long as $\|\Delta X_{\min}\| \geq \epsilon$, where ϵ is the desired accuracy. For computational efficiency, the inverse of the matrix G and the K matrix can be precomputed at the start of the algorithm. Therefore, each iteration of the algorithm involves only two matrix vector products ($\mathcal{O}(n^2)$). In the next section, we present experimental results of applying our algorithm to a simulated sequence and also a real sequence where pose constraints are obtained by visual odometry.

V. Results

Figure 1 shows the $3D$ trajectory consisting of 100 poses $X_1, \ldots, X_{100}$ used in our simulation for experimental validation of our algorithm. Although the orientation of the robot at these poses is not shown, it also varies substantially between two consecutive locations. Thus, the poses X_i have the full six degrees of freedom. The robot starts and ends at the same location and orientation, thereby resulting in a closed loop. The total distance traveled is 22346 mm. Relative pose measurements were obtained

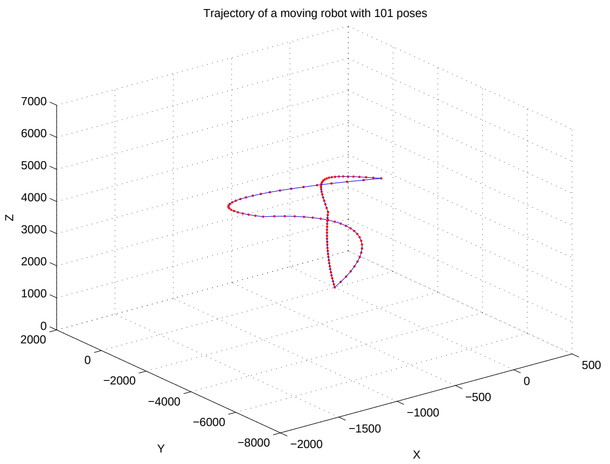

Fig. 1. $3D$ trajectory of a moving robot with 100 poses

between 15 consecutive locations, that is, we have local pose constraints, D_{ij} for all i, j such that $j > i$ and $|j - i| \leq 15$. Each such pose measurement, $\tilde{D}_{ij}$ can be computed as $\tilde{D}_{ij} = BCH(X_j, -X_i)$. For each such constraint, we need to simulate noise during the measurement process. This is accomplished by first generating a 6×6 random symmetric positive definite matrix, C_{ij}, for each $\tilde{D}_{ij}$. Next, we generate a random correlated vector h_{ij} with covariance matrix C_{ij}. h_{ij} is added to $\tilde{D}_{ij}$ to obtain the observed measurement $\bar{D}_{ij}$. In other words

$$\bar{D}_{ij} = BCH(X_j, -X_i) + h_{ij} \tag{35}$$

Thus, the inputs to our algorithm are the measured relative pose pairs $(\bar{D}_{ij}, C_{ij})$. The outputs are the poses $\bar{X}_i$ computed at all the 100 locations on the trajectory. We initialize the algorithm with all the poses set to zero. The error of the results obtained is measured by finding the distance between the computed location $(\bar{t}_i)$ and the ground truth location (t_i); that is, $e_i = \sqrt{|t_i - \bar{t}_i|}$.

Given these pose links, it is straightforward to link the consecutive poses together and thus reconstruct the trajectory from the 99 consecutive local pose links $(\bar{D}_{i(i+1)})$. Obviously, this does not take into account the covariance estimates of these adjacent links. Figure 2, plot 1 shows the error in the locations obtained by linking these adjacent poses together. The X axis is the pose index, and the Y axis is the error. Notice that the error in localization keeps on accumulating as we move further. At the end, the accumulated error is 6052 which is about 27% of the total distance traveled. Plot 2 in the same figure is an error plot obtained by applying our Lie algebraic CPR algorithm to just the consecutive pose relations $(\bar{D}_{i(i+1)})$. Notice that the error is much more well behaved and more or less spread uniformly throughout. The accumulated error in this case is 354 mm, which is only 1.58% of the total distance travelled. Thus, incorporating covariances and minimization of Mahalanobis distances reduces the error dramatically.

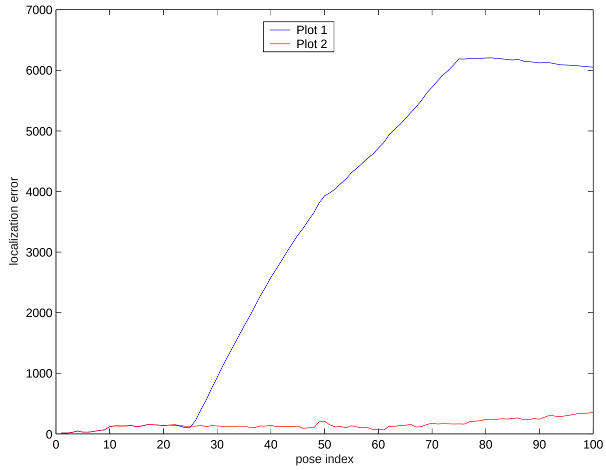

Fig. 2. Error plot obtained from adjacent pose relations

As the number of relative pose links is increased, more constraints are added to the system. Therefore, the error in localization should decrease. Figure 3 shows the effect of adding more constraints on the localization error. Plot 1 is the error plot obtained by linking consecutive poses together (this is same as the error plot 2 of the previous figure). Plot 2 is the error plot obtained where constraints from *two* consecutive pose constraints are incorporated, and plot 3 is the error plot obtained by adding constraints from five adjacent relative pose links. Thus, it is clear that the error dramatically reduces as we add more local pose links. The accumulated error for the trajectory obtained by considering five adjacent pose links is only 0.02%.

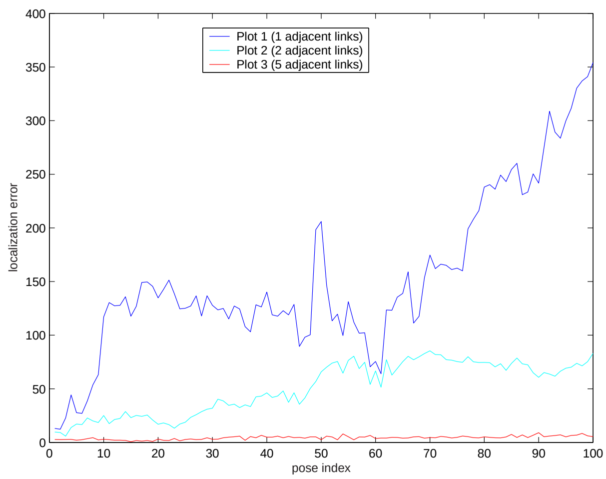

Fig. 3. Effect of increasing the number of local pose constraints

If we do not have information about the covariance matrix, then we may take it to be the identity matrix. In such a case, the Mahalanobis distance becomes the sum of the squared errors and the problem simply becomes the standard least squares minimization. Figure 4 compares the error plot obtained by least squares minimization with that of Mahalanobis distance minimization for the case of five adjacent links. From this error plot, it is clear that incorporating the covariance matrix helps reduce the overall error.

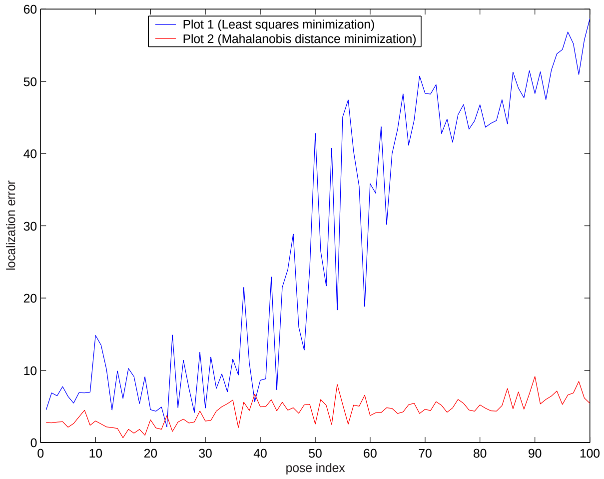

Fig. 4. Comparison of least squares minimization with our approach

Our algorithm typically converges in about 40 to 50 iterations for a precision corresponding to $\epsilon = 1e^{-5}$. In addition, we have observed that the initial starting point for the nonlinear minimization does not affect the computed poses significantly.

We have also applied our approach to a real-world sequence consisting of 50 poses. A pair of stereo cameras was mounted on a cart and moved in an indoor hallway at the intersection of two corridors. The camera was moved so as to 'cut the corner' at the intersections of the two corridors. Relative poses between every 4 adjacent frames were obtained through visual odometry [8]. The covariance matrix for the relative pose was computed from the Jacobian matrix of the nonlinear function which is minimized to estimate the pose from the visual features. Figure 5, plots 2,3,4 shows the estimated trajectory of the vehicle using the approach described in this paper. The different trajectories correspond to different number of adjacent pose constraints used to solve the CPR. For reference, the trajectory obtained by linking consecutive poses (as in Figure 2) is shown in plot 1. Since ground truth is not available for this data, we can not compare the errors, but the trajectories from the plots 1 and 4 are quite different, especially towards the end of the sequence. Linking consecutive poses leads to error accumulation as we proceed and therefore the endpoints are different. As more and more relative pose constraints are added, the CPR algorithm, probably converges to the true pose estimates.

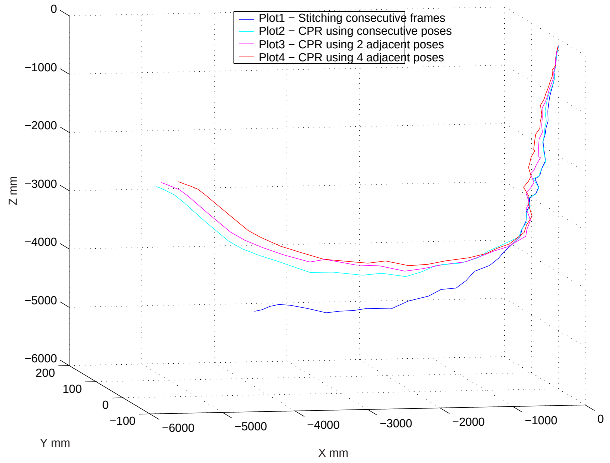

Fig. 5. CPR applied to a real-world sequence

VI. Conclusion

We have presented a Lie algebraic framework for the global localization of a moving robot from uncertain relative pose estimates. Given these pose constraints and their uncertainties, we formulate the problem as minimizing the Mahalanobis distances in the Lie algebra. This nonlinear minimization problem is solved by iterative application of the consistent pose registration framework of Lu and Milios. Experimental results demonstrate that our algorithm converges rapidly and the minimization is not very much affected by the initial starting point. Our approach can be applied to robots moving in outdoor environments where the motion is not necessarily planar. This extends the applicability of robot navigation to rugged, natural terrains.

Although we have demonstrated the proof of concept of our approach by applying it to a simulated sequence and a real experiment with pose relations obtained through visual odometry, we are in the midst of extensive testing and experimental evaluations. In particular, we plan to combine relative pose measurements from different modalities such as visual sensing, vehicle odometry, and laser range finders to obtain the trajectory of the vehicle and compare it with precise ground truth measurements using differential GPS. We are also investigating the effect of the number of available relative pose measurements on the recovered trajectory. As formulated in this paper, our approach is essentially a batch method and requires the inversion of a large matrix. This may become computationally prohibitive for large sequences. We are currently working on developing an incremental version of our algorithm in the lines of [10] and also looking at computational aspects of our algorithm for large-scale map making [11].

Acknowledgment

The author thanks Kurt Konolige at SRI International and Venu Govindu at the University of Maryland for initial discussions and ideas leading to this paper.

References

[1] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[2] H. H. Gonzalez-Banos and J. C. Latombe, "Navigation strategies for exploring indoor environments," *International Journal of Robotics Research*, 2002.

[3] J. D. Tards, J. Neira, P. M. Newman, and J. J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *The International Journal of Robotics Research*, 2002.

[4] C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *Proceedings IEEE International Conference on Robotics and Automation*, May 2003.

[5] C. Brenneke, O. Wulf, and B. Wagner, "Using 3d laser range data for slam in outdoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2003, pp. 27–31.

[6] J. Guivant, E. Nebot, and S. Baiker, "Autonomous navigation and map building using laser range sensors in outdoor applications," *Journal of Robotics Systems*, vol. 17, no. 10, pp. 565–583, October 2000.

[7] M.A.Garcia and A.Solanas, "3d simultaneous localization and modeling from stereo vision," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 847–853.

[8] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2004.

[9] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[10] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Monterey, California, November 1999, pp. 318–325.

[11] K. Konolige, "Large-scale map-making," in *Proc. National Conference on AI (AAAI)*, 2004.

[12] V. M. Govindu, "Lie-algebraic averaging for globally consistent motion estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2004.

[13] J. Selig, *Geometrical Methods in Robotics*. Springer-Verlag, 1996.

[14] V. Varadarajan, *Lie Groups, Lie Algebras and Their Representations*, ser. Graduate Texts in Mathematics. Springer-Verlag, 1984, vol. 102.