Robotics and Autonomous Systems 59 (2011) 310–318

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Amortized constant time state estimation in Pose SLAM and hierarchical SLAM using a mixed Kalman-information filter

Viorela Ila *, Josep M. Porta, Juan Andrade-Cetto

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain

ARTICLE INFO

Article history:
Available online 11 March 2011

Keywords:
State recovery
Kalman filter
Information filter
Pose SLAM
Hierarchical SLAM

ABSTRACT

The computational bottleneck in all information-based algorithms for simultaneous localization and mapping (SLAM) is the recovery of the state mean and covariance. The mean is needed to evaluate model Jacobians and the covariance is needed to generate data association hypotheses. In general, recovering the state mean and covariance requires the inversion of a matrix with the size of the state, which is computationally too expensive in time and memory for large problems. Exactly sparse state representations, such as that of Pose SLAM, alleviate the cost of state recovery either in time or in memory, but not in both. In this paper, we present an approach to state estimation that is linear both in execution time and in memory footprint at loop closure, and constant otherwise. The method relies on a state representation that combines the Kalman and the information-based approaches. The strategy is valid for any SLAM system that maintains constraints between marginal states at different time slices. This includes both Pose SLAM, the variant of SLAM where only the robot trajectory is estimated, and hierarchical techniques in which submaps are registered with a network of relative geometric constraints.

1. Introduction

Seminal solutions to the simultaneous localization and mapping (SLAM) problem relied on the extended Kalman filter (EKF) to estimate the mean absolute position of landmarks and the robot pose and their associated covariance matrix [1,2]. This has quadratic memory and computational cost, limiting its use to small areas.

Instead of using the mean and the covariance, Gaussian distributions can be parametrized in canonical form using the information vector and the information matrix. In SLAM, the information matrix turns out to be approximately sparse, i.e., the matrix entries for distant landmarks are very small and the matrix can be sparsified with a minimal information loss, trading optimality for efficiency [3]. Efficiency without information loss is possible when estimating the entire robot path along with the map, an approach typically referred to as full SLAM [4–6]. Exact sparsification is also possible if only a set of variables is maintained; either by keeping a small set of active landmarks [7], by decoupling the estimation problem maintaining the map only [8], or as it is done in Pose SLAM, by maintaining only the pose history [9,10]. In Pose SLAM, landmarks are only used to obtain relative measurements linking pairs of poses. When working with sensors that are able to identify many landmarks per pose, Pose SLAM produces more compact maps than the other exactly sparse approaches.

Due to their small memory footprint, sparse representations enable SLAM solutions that scale nicely to very large maps. Off-line information-based SLAM approaches [5,11,12] obtain the maximum likelihood solution from the constraints encoded in the information matrix. The optimization iteratively approximates the mean solving a sequence of linear systems using the previously estimated mean as a linearization point for the constraints. This process assumes data association for granted, somehow limiting its applicability. On-line information-based approaches rely either on variants of the batch methods [6] or, more commonly, on filtering [9,13] using the Extended Information Filter (EIF) as the estimation tool of choice. These on-line systems not only have to recover the mean to evaluate the Jacobians, but also need to address the data association problem. Data association might be tackled directly from sensor readings, without relying on the filtered pose priors [14]. The process, however, is prone to perceptual aliasing and it is often convenient to take advantage of the state estimates to limit the search space. False positives can be avoided performing prior-based data association tests that use cross covariances between match candidates. Both, the mean and the cross covariances, are not directly available from the estimates of the information-based representations.

The EKF and the EIF applied to SLAM are different in nature. While in the former the estimate includes all the necessary data for linearization and data association, the latter is advantageous from the point of view of memory footprint. In this paper, we propose a combination of these two filters with the aim of getting the best

* Corresponding author.
E-mail addresses: vila@iri.upc.edu (V. Ila), porta@iri.upc.edu (J.M. Porta), cetto@iri.upc.edu (J. Andrade-Cetto).

0921-8890/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2011.02.010

of the two worlds: reduced memory complexity and easy access to the mean and the relevant blocks of the covariance matrix.

The work presented in this paper improves the formalization of the state estimation technique in [15], where we adopted an extended information filter approach. Here, we abandon this paradigm and propose a novel mixed Kalman-information filter. Moreover, while the approach presented in [15] is limited to Pose SLAM, here we exploit the properties of the new mixed Kalman-information filter to generalize the approach to both the Pose SLAM problem and to hierarchical SLAM. For the sake of clarity, our presentation is sequential in order, first we introduce the new approach in the context of Pose SLAM and we latter extend it to hierarchical SLAM.

The paper is structured as follows. In Section 2, we formalize the Pose SLAM problem and describe its solution via EKF and EIF. In Section 3, we describe a combination of the two filters that allows state estimation in linear time and space complexities. Section 4 describes a refinement of the presented approach that allows updates in constant time during open loop traverse. This is relevant for approaches that carefully select the loops to close in order to avoid inconsistency as much as possible [13] or where previously mapped areas are barely revisited [16]. In these contexts, the linear time complexity of loop closure is amortized over long periods yielding an almost constant time state update. Section 5 extends the approach to hierarchical SLAM and Section 6 presents results both with simulated data and with real data sets that validate the presented approach. Concluding remarks are given in Section 7.

2. Pose SLAM formulation

In the on-line form of Pose SLAM, the objective is to estimate the trajectory of the robot, $\mathbf{x}_n = \{x_0, \ldots, x_n\}$, with x_i the robot pose at time i. The following applies for poses in $SE(2)$ or in $SE(3)$ and in Section 6 we particularize the approach to the planar case. Using a Bayesian recursion, the trajectory, $\mathbf{x}_n$, is updated given a set of observations, $\mathbf{z}_n$, of the relative displacement between the current robot pose and previous poses along the path

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_n) \propto p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{z}_n|\mathbf{x}_n).$$

The observations set $\mathbf{z}_n$ can be split in two independent groups $\mathbf{z}_n = \{\mathbf{u}_n, \mathbf{y}_n\}$ where $\mathbf{u}_n$ gives the displacement between the current robot pose and the immediate previous one, and $\mathbf{y}_n$ links the current pose with any other pose but the previous one. With this, the probabilistic model becomes

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_n) &\propto p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{u}_n, \mathbf{y}_n|\mathbf{x}_n) \\ &\propto p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{u}_n|\mathbf{x}_n)p(\mathbf{y}_n|\mathbf{x}_n) \\ &\propto p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{u}_n)p(\mathbf{y}_n|\mathbf{x}_n). \end{aligned} \tag{1}$$

The estimation problem in Eq. (1) corresponds to the SLAM operations of augmenting the state, $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{u}_n)$, and updating the robot path using relative observations, $p(\mathbf{y}_n|\mathbf{x}_n)$.

Assuming Gaussian distributions, the probabilities in Eq. (1) can be parametrized either in terms of their mean and covariance, $\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, or in terms of the information vector and matrix, $\mathbf{x}_n \sim \mathcal{N}^{-1}(\boldsymbol{\eta}_n, \boldsymbol{\Lambda}_n)$, with $\boldsymbol{\eta}_n = \boldsymbol{\Lambda}_n\, \boldsymbol{\mu}_n$, $\boldsymbol{\Lambda}_n = \boldsymbol{\Sigma}_n^{-1}$, and in which the estimation workhorses are the extended Kalman and information filters, respectively.

Note that simultaneous observations are independent and, thus, observations linking the same pair of poses can be fused before using them to update the filter. In particular, we can assume the set $\mathbf{u}_n$ to include a single element, u_n.

2.1. EKF Pose SLAM state estimation

The observation $u_n \sim \mathcal{N}(\mu_u, \boldsymbol{\Sigma}_u)$ is used to augment the state with a new pose. In Pose SLAM, the state transition model is given by

$$\begin{aligned} x_n &= f(x_{n-1}, u_n) \\ &\approx f(\mu_{n-1}, \mu_u) + \mathbf{F}_n\, (x_{n-1} - \mu_{n-1}) + \mathbf{W}_n(u_n - \mu_u) \end{aligned}$$

with $\mathbf{F}_n$ and $\mathbf{W}_n$ the Jacobians of f with respect to x_{n-1} and u_n, evaluated at μ_{n-1} and μ_u. The EKF augments the state as

$$\boldsymbol{\mu}_n = \begin{bmatrix} \boldsymbol{\mu}_{n-1} \\ x_n \end{bmatrix}, \tag{2}$$

$$\boldsymbol{\Sigma}_n = \begin{bmatrix} \boldsymbol{\Sigma}_{1:n-2,1:n-2} & \boldsymbol{\Sigma}_{1:n-2,n-1}\, \mathbf{F}_n^\top \\ \mathbf{F}_n\, \boldsymbol{\Sigma}_{n-1,1:n-2} & \mathbf{F}_n\, \boldsymbol{\Sigma}_{n-1\;n-1}\, \mathbf{F}_n^\top + \mathbf{Q} \end{bmatrix}, \tag{3}$$

with $\mathbf{Q} = \mathbf{W}_n \boldsymbol{\Sigma}_u \mathbf{W}_n^\top$ and where $\boldsymbol{\Sigma}_{n-1\;n-1}$ is used to denote the block of $\boldsymbol{\Sigma}_{n-1}$ corresponding to the $(n-1)$th pose, and $\boldsymbol{\Sigma}_{1:k,1:k}$ indicate the blocks ranging from the first to the kth pose.

Each set of measures $\mathbf{y}_n = \{y_n^1, \ldots, y_n^k\}$ constrains the relative position of the last pose to some other poses from the robot trajectory forming loops. The measurement model for each of these constraints is

$$\begin{aligned} y_n^i &= h(x_i, x_n) + v_n \\ &\approx h(\mu_i, \mu_n) + \mathbf{H}(\mathbf{x}_n - \boldsymbol{\mu}_n) + v_n, \end{aligned}$$

where h gives the displacement from x_i to x_n in the reference frame of x_i, and $\mathbf{H}$ is

$$\mathbf{H} = [\mathbf{0}\ \ldots \mathbf{0}\ \mathbf{H}_i\ \mathbf{0}\ \ldots\ \mathbf{0}\ \mathbf{H}_n]\,, \tag{4}$$

with $\mathbf{H}_i$ and $\mathbf{H}_n$ the Jacobians of h with respect to x_i and x_n, and $v_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_y)$ the measurement white noise.

The information from observation y_n^i is merged into the filter applying the following increments

$$\Delta\boldsymbol{\mu} = \mathbf{K}\, (y_n^i - h(\mu_i, \mu_n)), \tag{5}$$

$$\Delta\boldsymbol{\Sigma} = -\mathbf{K}\, \mathbf{H}\, \boldsymbol{\Sigma}_n, \tag{6}$$

to $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$, respectively, where $\mathbf{K}$ is the Kalman gain, $\mathbf{K} = \boldsymbol{\Sigma}_n\, \mathbf{H}^\top\, \mathbf{S}^{-1}$, and $\mathbf{S}$ the innovation matrix, $\mathbf{S} = \mathbf{H}\, \boldsymbol{\Sigma}_n\, \mathbf{H}^\top + \boldsymbol{\Sigma}_y$.

Measurements y_n^i result from the data association process. Instead of directly comparing the sensor readings for the current pose with those for all poses along the trajectory, data association is generally tested on a limited region of the trajectory. To identify poses that are close enough to the current one so that the corresponding sensor readings are likely to match (i.e., to produce y_n^i observations), we can estimate the relative displacement, d, from the current robot pose, x_n, to any other previous pose in the trajectory, x_i, as a Gaussian with parameters

$$\mu_d = h(\mu_i, \mu_n), \tag{7}$$

$$\boldsymbol{\Sigma}_d = [\mathbf{H}_i\ \mathbf{H}_n] \begin{bmatrix} \boldsymbol{\Sigma}_{ii} & \boldsymbol{\Sigma}_{in} \\ \boldsymbol{\Sigma}_{in}^\top & \boldsymbol{\Sigma}_{nn} \end{bmatrix} [\mathbf{H}_i\ \mathbf{H}_n]^\top, \tag{8}$$

where $\boldsymbol{\Sigma}_{in}$ is the cross correlation between the ith and the current poses. Only poses whose relative displacement, d, is likely to be inside sensor range need to be considered for sensor registration.

Whereas the EKF estimation maintains all the data necessary for linearization and for data association, its drawback is that storing and updating the whole covariance matrix entails quadratic cost both in memory and in execution time.

2.2. EIF Pose SLAM state estimation

In the EIF form of Pose SLAM [9], the state is augmented as

$$\boldsymbol{\eta}_n = \begin{bmatrix} \boldsymbol{\eta}_{1:n-2} \\ \eta_{n-1} - \mathbf{F}_n{}^\top \mathbf{Q}^{-1}\, (f(\mu_{n-1}, \mu_u) - \mathbf{F}_n\, \mu_{n-1}) \\ \mathbf{Q}^{-1}\, (f(\mu_{n-1}, \mu_u) - \mathbf{F}_n\, \mu_{n-1}) \end{bmatrix},$$

$$\boldsymbol{\Lambda}_n = \begin{bmatrix} \boldsymbol{\Lambda}_{1:n-2,1:n-2} & \boldsymbol{\Lambda}_{1:n-2,n-1} & \mathbf{0} \\ \boldsymbol{\Lambda}_{n-1,1:n-2} & \boldsymbol{\Lambda}_{n-1\;n-1} + \mathbf{F}_n^\top \mathbf{Q}^{-1} \mathbf{F}_n & -\mathbf{F}_n^\top \mathbf{Q}^{-1} \\ \mathbf{0} & -\mathbf{Q}^{-1}\mathbf{F}_n & \mathbf{Q}^{-1} \end{bmatrix}. \tag{9}$$