A Spectral Learning Approach to Range-Only SLAM

Byron Boots BBOOTS@CS.WASHINGTON.EDU
Department of Computer Science and Engineering, University of Washington, Seattle, WA

Geoffrey J. Gordon GGORDON@CS.CMU.EDU
Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

We present a novel spectral learning algorithm for simultaneous localization and mapping (SLAM) from range data with known correspondences. This algorithm is an instance of a general spectral system identification framework, from which it inherits several desirable properties, including statistical consistency and no local optima. Compared with popular batch optimization or multiple-hypothesis tracking (MHT) methods for range-only SLAM, our spectral approach offers guaranteed low computational requirements and good tracking performance. Compared with MHT and with popular extended Kalman filter (EKF) or extended information filter (EIF) approaches, our approach does not need to linearize a transition or measurement model. We provide a theoretical analysis of our method, including finite-sample error bounds. Finally, we demonstrate on a real-world robotic SLAM problem that our algorithm is not only theoretically justified, but works well in practice: in a comparison of multiple methods, the lowest errors come from a combination of our algorithm with batch optimization, but our method alone produces nearly as good a result at far lower computational cost.

1. Introduction

In range-only SLAM, we are given a sequence of range measurements from a robot to fixed landmarks with known correspondences, and possibly a matching sequence of odometry measurements. We then attempt to simultaneously estimate the robot's trajectory and the locations of the landmarks. Popular approaches to range-only SLAM include EKFs and EIFs (Kantor & Singh, 2002; Kurth et al., 2003; Djugash & Singh, 2008; Djugash, 2010; Thrun et al., 2005), multiple-hypothesis trackers (including particle filters and multiple EKFs/EIFs) (Djugash et al., 2005; Thrun et al., 2005), and batch optimization of a likelihood function (Kehagias et al., 2006).

In all the above approaches, the most popular representation for a hypothesis is a list of landmark locations $(m_{n,x}, m_{n,y})$ and a list of robot poses (x_t, y_t, θ_t). Unfortunately, both the motion and measurement models are highly nonlinear in this representation, leading to computational problems: inaccurate linearizations in EKF/EIF/MHT and local optima in batch optimization approaches. Much work has attempted to remedy this problem, e.g., by changing the hypothesis representation (Djugash, 2010) or by keeping multiple hypotheses (Djugash et al., 2005; Djugash, 2010; Thrun et al., 2005). While considerable progress has been made, none of these methods are ideal; common difficulties include the need for an extensive initialization phase, inability to recover from poor initialization, lack of performance guarantees, or excessive computational requirements.

We take a very different approach: we formulate range-only SLAM as a matrix factorization problem, where features of observations are linearly related to a 4- or 7-dimensional state space. This approach has several desirable properties. First, we need weaker assumptions about the measurement model and motion model than previous approaches to SLAM. Second, our state space yields a *linear* measurement model, so we lose less information during tracking to approximation errors and local optima. Third, our formulation leads to a simple spectral learning algorithm, based on a fast and robust singular value decomposition (SVD)—in fact, our algorithm is an instance of a general spectral system identification framework, from which it inherits

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

desirable guarantees including statistical consistency and no local optima. Fourth, we don't need to worry as much as previous methods about errors such as a consistent bias in odometry, or a receiver mounted at a different height from the transmitters: in general, we can learn to correct such errors automatically by expanding the dimensionality of our state space.

As we will discuss in Section 2, our approach to SLAM has much in common with spectral algorithms for subspace identification (Van Overschee & De Moor, 1996; Boots et al., 2010); unlike these methods, our focus on SLAM makes it easy to *interpret* our state space. Our approach is also related to factorization-based structure from motion (Tomasi & Kanade, 1992; Triggs, 1996; Kanade & Morris, 1998), as well as to recent dimensionality-reduction-based methods for localization and mapping (Biggs et al., 2005; Ferris et al., 2007; Yairi, 2007). Unlike the latter approaches, which are generally nonlinear, we only require linear dimensionality reduction. Finally, the simplest version of our method (Sec. 3.1) is related to multidimensional scaling (MDS). In MDS, there is no distinction between landmark positions and robot positions: we have *all-to-all* range measurements for a set of landmarks, and must recover an embedding of the landmarks. Our smaller set of measurements introduces additional challenges, and forces changes compared to MDS.

2. Background

There are three main pieces of relevant background: first, the well-known solutions to range-only SLAM using variations of the extended Kalman filter and batch optimization; second, recently-discovered spectral approaches to identifying parameters of nonlinear dynamical systems; and third, matrix factorization for finding structure from motion in video. Below, we will discuss the connections between these areas, and show how they can be unified within a spectral learning framework.

2.1. Likelihood-based Range-only SLAM

The standard probabilistic model for range-only localization (Kantor & Singh, 2002; Kurth et al., 2003) represents robot state by a vector $s_t = [x_t, y_t, \theta_t]^\mathsf{T}$; the robot's (nonlinear) motion and observation models are

$$s_{t+1} = \begin{bmatrix} x_t + v_t \cos(\theta_t) \\ y_t + v_t \sin(\theta_t) \\ \theta_t + \omega_t \end{bmatrix} + \epsilon_t \quad (1)$$

$$d_{t,n} = \sqrt{(m_{n,x} - x_t)^2 + (m_{n,y} - y_t)^2} + \eta_t \quad (2)$$

Here v_t is the distance traveled, ω_t is the orientation change, $d_{t,n}$ is the estimate of the range from the nth landmark location $(m_{n,x}, m_{n,y})$ to the current location of the robot (x_t, y_t), and ϵ_t and η_t are noise. (Throughout this paper we assume known correspondences, since range sensing systems such as radio beacons typically associate unique identifiers with each reading.) To handle SLAM rather than just localization, we extend the state to include landmark positions:

$$s_t = [x_t, y_t, \theta_t, m_{1,x}, m_{1,y}, \ldots, m_{N,x}, m_{N,y}]^\mathsf{T} \quad (3)$$

where N is the number of landmarks. The motion and measurement models remain the same. Given this model, we can use any standard optimization algorithm (such as Gauss-Newton) to fit the unknown robot and landmark parameters by maximum likelihood. Or, we can track these parameters online using EKFs, EIFs, or MHT methods like particle filters.

EKFs and EIFs are a popular solution for localization and mapping problems: for each new odometry input $a_t = [v_t, \omega_t]^\mathsf{T}$ and each new measurement d_t, we propagate the estimate of the robot state and error covariance by linearizing the non-linear motion and measurement models. Unfortunately, though, range-only SLAM is notoriously difficult for EKFs/EIFs: since range-only sensors are not informative enough to completely localize a robot or a landmark from a small number of readings, nonlinearities are much worse in range-only SLAM than they are in other applications such as range-and-bearing SLAM. In particular, if we don't have a sharp prior distribution for landmark positions, then after a few steps, the exact posterior becomes highly non-Gaussian and multimodal; so, any Gaussian approximation to the posterior is necessarily inaccurate. Furthermore, an EKF will generally not even produce the best possible Gaussian approximation: a good linearization would tell us a lot about the modes of the posterior, which would be equivalent to solving the original SLAM problem. So, practical applications of the EKF to range-only SLAM attempt to *delay* linearization until enough information is available, e.g., via an extended initialization phase for each landmark. Finally, Djugash et al. proposed a polar parameterization to more accurately represent the annular and multimodal distributions typically encountered in range-only SLAM. The resulting approach is called the ROP-EKF, and is shown to outperform the ordinary (Cartesian) EKF in several real-world problems, especially in combination with multiple-hypothesis tracking (Djugash & Singh, 2008; Djugash, 2010). However, the multi-hypothesis ROP-EKF can be much more expensive than an EKF, and is still a heuristic approximation to the true posterior.

2.2. Spectral System Identification

System identification algorithms attempt to *learn* dynamical system parameters such as a state space, a

dynamics model (motion model), and an observation model (measurement model) directly from samples of observations and actions. In the last few years, spectral system identification algorithms have become popular; these algorithms learn a state space via a spectral decomposition of a carefully designed matrix of observable features, then find transition and observation models by linear regressions involving the learned states. Originally, subspace identification algorithms were almost exclusively used for linear system identification (Van Overschee & De Moor, 1996), but recently, similar spectral algorithms have been used to learn models of partially observable nonlinear dynamical systems such as HMMs (Hsu et al., 2009; Siddiqi et al., 2010) and PSRs (Rosencrantz et al., 2004; Boots et al., 2010; Boots & Gordon, 2010; Boots et al., 2011). This is a powerful and appealing approach: the resulting algorithms are *statistically consistent*, unlike the popular expectation maximization (EM) algorithm, and they are easy to implement with efficient linear algebra operations. As we will see in Section 3, we can view the range-only SLAM problem as an instance of spectral state space discovery. And, the Appendix (Sec. C) discusses how to identify transition and measurement models given the learned states. The same properties that make spectral methods appealing for system identification carry over to our spectral SLAM algorithm: computational efficiency, statistical consistency, and finite-sample error bounds.

2.3. Orthographic Structure From Motion

In some ways the orthographic structure from motion (SfM) problem in vision (Tomasi & Kanade, 1992) is very similar to the SLAM problem: the goal is to recover scene geometry and camera rotations from a sequence of images (compare with landmark geometry and robot poses from a sequence of range observations). And in fact, one popular solution for SfM is very similar to the state space discovery step in spectral state space identification. The key idea in spectral SfM is that is that an image sequence can be represented as a $2F \times P$ measurement matrix W, containing the horizontal and vertical coordinates of P points tracked through F frames. If the images are the result of an orthographic camera projection, then it is possible to show that $\text{rank}(W) = 3$. As a consequence, the measurement matrix can be factored into the product of two matrices U and V, where U contains the 3d positions of the features and V contains the camera axis rotations (Tomasi & Kanade, 1992). With respect to system identification, it is possible to interpret the matrix U as an observation model and V as an estimate of the system state. Inspired by SfM, we reformulate range-only SLAM problem in a similar way in Section 3, and then similarly solve the problem with a spectral learning algorithm.

3. State Space Discovery and Spectral SLAM

We start with SLAM from range data without odometry. For now, we assume no noise, no missing data, and batch processing. We will generalize below: Sec. 3.2 discusses how to recover robot orientation, Sec. 3.3 discusses noise, and Sec. 3.4 discusses missing data and online SLAM. In the Appendix (Section C) we discuss learning motion and measurement models.

3.1. Range-only SLAM as Matrix Factorization

Consider the matrix $Y \in \mathbb{R}^{N\times T}$ of squared ranges, with $N \geq 4$ landmarks and $T \geq 4$ time steps:

$$Y = \frac{1}{2}\begin{bmatrix} d_{11}^2 & d_{12}^2 & \dots & d_{1T}^2 \\ d_{21}^2 & d_{22}^2 & \dots & d_{2T}^2 \\ \vdots & \vdots & \vdots & \vdots \\ d_{N1}^2 & d_{N2}^2 & \dots & d_{NT}^2 \end{bmatrix} \tag{4}$$

where $d_{n,t}$ is the measured distance from the robot to landmark n at time step t. The most basic version of our spectral SLAM method relies on the insight that Y *factors* according to robot position (x_t, y_t) and landmark position $(m_{n,x}, m_{n,y})$. To see why, note

$$d_{n,t}^2 = (m_{n,x}^2 + m_{n,y}^2) - 2m_{n,x}\cdot x_t - 2m_{n,y}\cdot y_t + (x_t^2 + y_t^2) \tag{5}$$

If we write $C_n = [(m_{n,x}^2 + m_{n,y}^2)/2, m_{n,x}, m_{n,y}, 1]^\mathsf{T}$ and $X_t = [1, -x_t, -y_t, (x_t^2 + y_t^2)/2]^\mathsf{T}$, it is easy to see that $d_{n,t}^2 = 2C_n^\mathsf{T} X_t$. So, Y factors as $Y = CX$, where $C \in \mathbb{R}^{N\times 4}$ contains the positions of landmarks, and $X \in \mathbb{R}^{4\times T}$ contains the positions of the robot over time:

$$C = \begin{bmatrix} (m_{1,x}^2 + m_{1,y}^2)/2 & m_{1,x} & m_{1,y} & 1 \\ (m_{2,x}^2 + m_{2,y}^2)/2 & m_{2,x} & m_{2,y} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ (m_{N,x}^2 + m_{N,y}^2)/2 & m_{N,x} & m_{N,y} & 1 \end{bmatrix} \tag{6}$$

$$X = \begin{bmatrix} 1 & \dots & 1 \\ -x_1 & \dots & -x_T \\ -y_1 & \dots & -y_T \\ (x_1^2 + y_1^2)/2 & \dots & (x_T^2 + y_T^2)/2 \end{bmatrix} \tag{7}$$

If we can recover C and X, we can read off the solution to the SLAM problem. The fact that Y's rank is only 4 suggests that we might be able to use a rank-revealing factorization of Y, such as the singular value decomposition, to find C and X. Unfortunately, such a factorization only determines C and X up to a linear transform: given an invertible matrix S, we can write

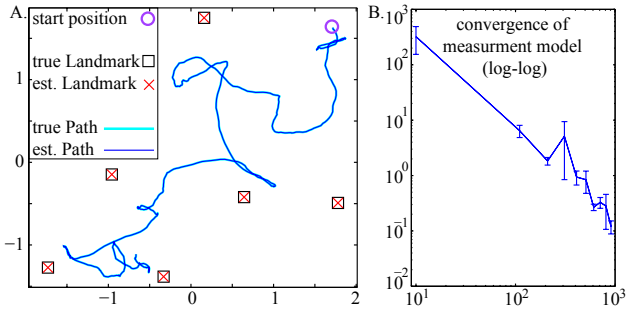

Figure 1. Spectral SLAM on simulated data. See Section D for details. A.) Randomly generated landmarks (6 of them) and robot path through the environment (500 timesteps). A SVD of the squared distance matrix recovers a linear transform of the landmark and robot positions. Given the coordinates of 4 landmarks, we can recover the landmark and robot positions exactly; or, since $500 \geq 8$, we can recover positions up to an orthogonal transform with no additional information. Despite noisy observations, the robot recovers the true path and landmark positions with very high accuracy. B.) The convergence of the observation model $\widehat{C}$: mean Frobenius-norm error vs. number of range readings received, averaged over 1000 randomly generated pairs of robot paths and environments. Error bars indicate 95% confidence intervals.

$Y = CX = CS^{-1}SX$. Therefore, factorization can only hope to recover $U = CS^{-1}$ and $V = SX$.

To upgrade the factors U and V to a full metric map, we have two options. If global position estimates are available for at least four landmarks, we can *learn* the transform S via linear regression, and so recover the original C and X. This method works as long as we know at least four landmark positions. Figure 1A shows a simulated example.

On the other hand, if no global positions are known, the best we can hope to do is recover landmark and robot positions up to an orthogonal transform (translation, rotation, and reflection). It turns out that Eq. (7) provides enough additional geometric constraints to do so: in the Appendix (Sec. A) we show that, if we have ≥ 8 time steps and ≥ 8 landmarks, we can compute the metric upgrade in closed form. The idea is to fit a quadratic surface to the rows of U, then change coordinates so that the surface becomes the function in (7). (By contrast, the usual metric upgrade for orthographic structure from motion (Tomasi & Kanade, 1992), which uses the constraint that camera projection matrices are orthogonal, requires a nonlinear optimization.)

3.2. SLAM with Headings

In addition to location, we often want the robot's global heading θ. We could get headings by post-processing our learned positions; but in practice we can reduce variance by learning positions and headings simultaneously. We do so by adding more features to our measurement matrix: differences between successive pairs of squared distances, scaled by velocity (which we can estimate from odometry). Since we need pairs of time steps, we now have $Y \in \mathbb{R}^{2N \times T-1}$:

$$Y = \frac{1}{2}\begin{bmatrix} d_{11}^2 & d_{12}^2 & \cdots & d_{1T-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^2 & d_{N2}^2 & \cdots & d_{NT-1}^2 \\ \frac{d_{12}^2 - d_{11}^2}{v_1} & \frac{d_{13}^2 - d_{12}^2}{v_2} & \cdots & \frac{d_{1T}^2 - d_{1T-1}^2}{v_{T-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{d_{N2}^2 - d_{N1}^2}{v_1} & \frac{d_{N2}^2 - d_{N3}^2}{v_2} & \cdots & \frac{d_{NT}^2 - d_{NT-1}^2}{v_{T-1}} \end{bmatrix} \tag{8}$$

As before, we can factor Y into a state matrix and a landmark matrix. The key observation is that we can write the new features in terms of $\cos(\theta)$ and $\sin(\theta)$:

$$\begin{aligned} \frac{d_{n,t+1}^2 - d_{n,t}^2}{2v_t} &= -\frac{m_{n,x}(x_{t+1} - x_t)}{v_t} - \frac{m_{n,y}(y_{t+1} - y_t)}{v_t} \\ &\quad + \frac{x_{t+1}^2 - x_t^2 + y_{t+1}^2 - y_t^2}{2v_t} \\ &= -m_{n,x}\cos(\theta_t) - m_{n,y}\sin(\theta_t) \\ &\quad + \frac{x_{t+1}^2 - x_t^2 + y_{t+1}^2 - y_t^2}{2v_t} \end{aligned} \tag{9}$$

From Eq. 5 and Eq. 9 it is easy to see that Y is rank 7: we have $Y = CX$, where $C \in \mathbb{R}^{N \times 7}$ contains functions of landmark positions and $X \in \mathbb{R}^{7 \times T}$ contains functions of robot state,

$$C = \begin{bmatrix} (m_{1,x}^2 + m_{1,y}^2)/2 & m_{1,x} & m_{1,y} & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (m_{N,x}^2 + m_{N,y}^2)/2 & m_{N,x} & m_{N,y} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_{1,x} & m_{1,y} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & m_{N,x} & m_{N,y} & 1 \end{bmatrix} \tag{10}$$

$$X = \begin{bmatrix} 1 & \cdots & 1 \\ -x_1 & \cdots & -x_{T-1} \\ -y_1 & \cdots & -y_{T-1} \\ (x_1^2 + y_1^2)/2 & \cdots & (x_{T-1}^2 + y_{T-1}^2)/2 \\ -\cos(\theta_1) & \cdots & -\cos(\theta_{T-1}) \\ -\sin(\theta_1) & \cdots & -\sin(\theta_{T-1}) \\ \frac{x_2^2 - x_1^2 + y_2^2 - y_1^2}{2v_1} & \cdots & \frac{x_T^2 - x_{T-1}^2 + y_T^2 - y_{T-1}^2}{2v_{T-1}} \end{bmatrix} \tag{11}$$

As with the basic SLAM algorithm in Section 3.1, we can factor Y using SVD, this time keeping 7 singular values. Then we can do a metric upgrade as before to get robot positions (see Appendix, Sec. A for details); finally we can recover headings as angles between successive positions.

3.3. A Spectral SLAM Algorithm

The matrix factorizations of Secs. 3.1 and 3.2 suggest a straightforward SLAM algorithm, Alg. 1: build an

Algorithm 1 Spectral SLAM

In: *i.i.d.* pairs of observations $\{o_t, a_t\}_{t=1}^T$; optional: measurement model for ≥ 4 landmarks $\mathcal{C}_{1:4}$
Out: measurement model (map) $\widehat{C}$, robot locations $\widehat{X}$ (the tth column is location at time t)

1: Collect observations and odometry into a matrix $\widehat{Y}$ (Eq. 8)
2: Find the the top 7 singular values and vectors: $\langle \widehat{U}, \widehat{\Lambda}, \widehat{V}^\top \rangle \leftarrow \text{SVD}(\widehat{Y}, 7)$
3: Find the transformed measurement matrix $\widehat{C}S^{-1} = \widehat{U}$ and robot states $S\widehat{X} = \widehat{\Lambda}\widehat{V}^\top$ via linear regression (from $\widehat{C}_{1:4}$ to $\mathcal{C}_{1:4}$) or metric upgrade (see Appendix).

empirical estimate $\widehat{Y}$ of Y by sampling observations as the robot traverses its environment, then apply a rank $k = 7$ thin SVD, discarding the remaining singular values to suppress noise.

$$\langle \widehat{U}, \widehat{\Lambda}, \widehat{V}^\top \rangle \leftarrow \text{SVD}(\widehat{Y}, 7) \tag{12}$$

Following Section 3.2, the left singular vectors $\widehat{U}$ are an estimate of our transformed measurement matrix CS^{-1}, and the weighted right singular vectors $\widehat{\Lambda}\widehat{V}^\top$ are an estimate of our transformed robot state SX. We can then perform metric upgrade as before.

Statistical Consistency and Sample Complexity Let $M \in \mathbb{R}^{N \times N}$ be the *true* observation covariance for a randomly sampled robot position, and let $\widehat{M} = \frac{1}{T}\widehat{Y}\widehat{Y}^\top$ be the empirical covariance estimated from T observations. Then the true and estimated measurement models are the top singular vectors of M and $\widehat{M}$. Assuming that the noise in $\widehat{M}$ is zero-mean, as we include more data in our averages, we will show below that the law of large numbers guarantees that $\widehat{M}$ converges to the true covariance M. That is, our learning algorithm is *consistent*. (The estimated robot positions will typically not converge, since we typically have a bounded effective number of observations relevant to each robot position. But, as we see each landmark again and again, the robot position errors will average out, and we will recover the true map.)

In more detail, we can give finite-sample bounds on the error in recovering the true factors. For simplicity of presentation we assume that noise is i.i.d., although our algorithm will work for any zero-mean noise process with a finite mixing time. (The error bounds will of course become weaker in proportion to mixing time, since we gain less new information per observation.) The argument (see the Appendix, Sec. B, for details) has two pieces: standard concentration bounds show that each element of our estimated covariance approaches its population value; then the continuity of the SVD shows that the learned subspace also approaches its true value. The final bound is:

$$||\sin \Psi||_2 \leq \frac{Nc\sqrt{\frac{2\log(T)}{T}}}{\gamma} \tag{13}$$

where Ψ is the vector of canonical angles between the learned subspace and the true one, c is a constant depending on our error distribution, and γ is the true smallest nonzero eigenvalue of the covariance. In particular, this bound means that the sample complexity is $\tilde{O}(\zeta^2)$ to achieve error ζ.

3.4. Extensions

Missing data So far we have assumed that we receive range readings to all landmarks at each time step. In practice this assumption is rarely satisfied: we may receive range readings asynchronously, some range readings may be missing entirely, and it is often the case that odometry data is sampled faster than range readings. Here we outline two methods for overcoming this practical difficulty.

First, if a relatively small number of observations are missing, we can use standard approaches for factorization with missing data. For example, probabilistic PCA (Tipping & Bishop, 1999) estimates the missing entries via the EM algorithm, and matrix completion (Candès & Plan, 2009) uses a trace-norm penalty to recover a low-rank factorization with high probability. However, for range-only data, often the fraction of missing data is high and the missing values are structural rather than random.

The second approach is interpolation: we divide the data into overlapping subsets and then use local odometry information to interpolate the range data within each subset. To interpolate the data, we estimate a robot path by dead reckoning. For each point in the dead reckoning path we build the feature representation $[1, -x, -y, (x^2 + y^2)/2]^\top$. We then learn a linear model that predicts a squared range reading from these features (for the data points where range is available), as in Eq. 5. Next we predict the squared range along the entire path. Finally we build the matrix $\widehat{Y}$ by averaging the locally interpolated range readings. This linear approach works much better in practice than the fully probabilistic approaches mentioned above, and was used in our experiments in Section 4.

Online Spectral SLAM The algorithms developed in this section so far have had an important drawback: unlike many SLAM algorithms, they are batch methods not online ones. The extension to online SLAM

is straightforward: instead of first estimating $\widehat{Y}$ and then performing a SVD, we sequentially estimate our factors $\langle \widehat{U}, \widehat{\Lambda}, \widehat{V}^\top \rangle$ via online SVD (Brand, 2006; Boots et al., 2011).

Robot Filtering and System Identification So far, our algorithms have not directly used (or needed) a robot motion model in the learned state space. However, an explicit motion model is required if we want to *predict* future sensor readings or *plan* a course of action. We have two choices: we can derive a motion model from our learned transformation S between latent states and physical locations, or we can learn a motion model directly from data using spectral system identification. More details about both of these approaches can be found in the Appendix, Sec. C.

4. Experimental Results

We perform several SLAM and robot navigation experiments to illustrate and test the ideas proposed in this paper. For synthetic experiments that illustrate convergence and show how our methods work in theory, see Fig. 1. We also demonstrate our algorithm on data collected from a real-world robotic system with substantial amounts of missing data. Experiments were performed in Matlab, on a 2.66 GHz Intel Core i7 computer with 8 GB of RAM. In contrast to batch nonlinear optimization approaches to SLAM, the spectral learning methods described in this paper are *very* fast, usually taking less than a second to run.

We used two freely available range-only SLAM benchmark data sets collected from an autonomous lawn mowing robot (Djugash, 2010), shown in Fig. 2A.[1] These "Plaza" datasets were collected via radio nodes from Multispectral Solutions that use time-of-flight of ultra-wide-band signals to provide inter-node ranging measurements and the unique ID of the radio nodes. (Additional details can be found in (Djugash, 2010).) In "Plaza 1," the robot travelled 1.9km, occupied 9,658 distinct poses, and received 3,529 range measurements. The path taken is a typical lawn mowing pattern that balances left turns with an equal number of right turns; this type of pattern minimizes the effect of heading error. In "Plaza 2," the robot travelled 1.3km, occupied 4,091 poses, and received 1,816 range measurements. The path taken is a loop which amplifies the effect of heading error. The two data sets were both very sparse, with approximately 11 time steps (and up to 500 steps) between range readings for the worst landmark. We first interpolated the missing range readings with the method of Section 3.4. Then we applied the rank-7 spectral SLAM algorithm of Section 3.3; the results are depicted in Figure 2B-C. Qualitatively, we see that the robot's localization path conforms to the true path.

In addition to the qualitative results, we quantitatively compared spectral SLAM to a number of different competing range-only SLAM algorithms which have been used on the benchmark data sets. The localization root mean squared error (RMSE) in meters for each algorithm is shown in Figure 3. The baseline is dead reckoning (using only the robot's odometry information). Next are several standard online range-only SLAM algorithms, including the Cartesian EKF, FastSLAM (Montemerlo et al., 2002) with 5,000 particles, and the ROP-EKF algorithm (Djugash & Singh, 2008), which achieved the best prior results on the benchmark datasets. These previous results only reported the RMSE for the last 10% of the path, which is the *best* 10% of the path (since it gives the most time to recover from initialization problems). The full path localization error can be considerably worse, particularly for the initial portion of the path—see Fig. 5 (right) of (Djugash & Singh, 2008).

We also compared to batch nonlinear optimization, via the quasi-Newton BFGS method as implemented in Matlab's `fminunc` (see (Kehagias et al., 2006) for details).[2] This approach to solving the range-only SLAM problem can be very data efficient, but is subject to local optima and is computationally intensive. We followed the suggestions of (Kehagias et al., 2006) and initialized with the dead-reckoning estimate of the robot's path. For Plaza 1 and Plaza 2, the algorithm took roughly 2.5 hours and 45 minutes to converge respectively.

Finally, we ran our spectral SLAM algorithm on the same data sets. In contrast to BFGS, spectral SLAM is *statistically consistent*, and much faster: the bulk of the computation is the fixed-rank SVD, so the time complexity of the algorithm is $O((2N)^2 T)$ where N is the number of landmarks and T is the number of time steps. Empirically, spectral SLAM produced results that were comparable to batch optimization in 3-4 *orders of magnitude* less time (see Figure 3).

Spectral SLAM can also be used as an initialization procedure for nonlinear batch optimization. This strategy combines the best of both algorithms by allowing the locally optimal nonlinear optimization procedure to start from a theoretically guaranteed good starting point. Therefore, the local optimum found by

[1] http://www.frc.ri.cmu.edu...
.../projects/emergencyresponse/RangeData/index.html

[2] We also tried Preconditioned Conjugate Gradient and Levenberg–Marquardt, but found BFGS to be the most efficient and stable nonlinear optimization procedure for this particular problem.

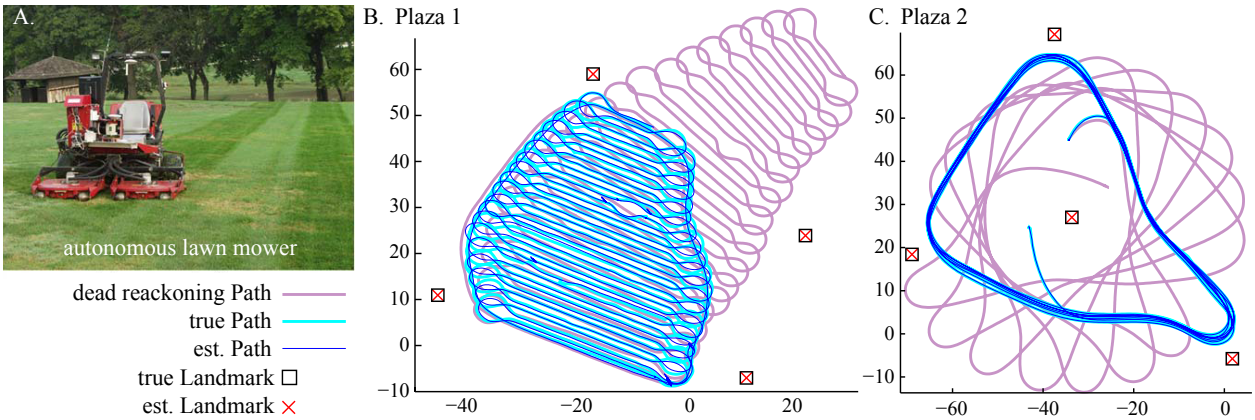

Figure 2. The autonomous lawn mower and spectral SLAM. A.) The robotic lawn mower platform. B.) In the first experiment, the robot traveled 1.9km receiving occupying 9,658 poses and receiving 3,529 range measurements. This path minimizes the effect of heading error by balancing the number of left turns with an equal number of right turns in the robot's odometry (a commonly used path pattern in lawn mowing applications). The light blue path indicates the robot's true path in the environment, light purple indicates dead-reckoning path, and dark blue indicates the spectral SLAM localization result. C.) In the second experiment, the robot traveled 1.3km occupying 4,091 poses and receiving 1,816 range measurements. This path highlights the effect of heading error on dead reckoning performance by turning in the same direction repeatedly. Again, spectral SLAM is able to accurately recover the robot's path.

nonlinear batch optimization should be *no worse* than the spectral SLAM solution and likely much better than the batch optimization seeded by dead-reckoning. Empirically, we found this to be the case (Figure 3). If time and computational resources are scarce, then we believe that spectral SLAM is clearly the best approach; if computation is not an issue, the best results will likely be found by refining the spectral SLAM solution using a nonlinear batch optimization procedure.

5. Conclusion

We proposed a novel solution for the range-only SLAM problem that differs substantially from previous approaches. The essence of this new approach is to formulate SLAM as a factorization problem, which allows us to derive a local-minimum free *spectral learning* algorithm that is closely related to SfM and recent spectral methods for system identification. Additionally, we discuss how to contend with missing data and how to derive an online algorithm. We are able to prove statistical consistency and sample complexity bounds for our algorithm, while getting competitive error in practice compared to methods like full batch optimization of the likelihood on several benchmark data sets. Finally, we show that our algorithm is *fast*: we analyze the time complexity and demonstrate empirically that our algorithm can be orders of magnitude faster than competing nonlinear batch optimization procedures.

Acknowledgements

Byron Boots and Geoffrey J. Gordon were supported by ONR MURI grant N00014-09-1-1052. Byron Boots was supported by NSF grant EEEC-0540865.

References

Biggs, Michael, Ghodsi, Ali, Wilkinson, Dana, and Bowling, Michael. Action respecting embedding. In *In Proceedings of the Twenty-Second International Conference on Machine Learning*, pp. 65–72, 2005.

Boots, Byron and Gordon, Geoff. Predictive state temporal difference learning. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R.S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23*, pp. 271–279. 2010.

Boots, Byron, Siddiqi, Sajid M., and Gordon, Geoffrey J. Closing the learning-planning loop with predictive state representations. In *Proceedings of Robotics: Science and Systems VI*, 2010.

Boots, Byron, Siddiqi, Sajid, and Gordon, Geoffrey. An online spectral learning algorithm for partially observable nonlinear dynamical systems. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI-2011)*, 2011.

Brand, Matthew. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415(1):20–30, 2006.

Candès, Emmanuel J. and Plan, Yaniv. Matrix completion with noise. *CoRR*, abs/0903.3131, 2009.

Djugash, Joseph. Geolocation with range: Robustness, efficiency and scalability. PhD. Thesis, Carnegie Mellon University, 2010.

Djugash, Joseph and Singh, Sanjiv. A robust method of localization and mapping using only range. In *International Symposium on Experimental Robotics*, July 2008.

Djugash, Joseph, Singh, Sanjiv, and Corke, Peter Ian. Further results with localization and mapping using range from radio. In *International Conference on Field and Service Robotics (FSR '05)*, July 2005.

Method	Plaza 1	Plaza 2
Dead Reckoning (full path)	15.92m	27.28m
Cartesian EKF (last, best 10%)	0.94m	0.92m
FastSLAM (last, best 10%)	0.73m	1.14m
ROP EKF (last, best 10%)	0.65m	0.87m
Batch Opt. (worst 10%)	1.04m	0.45m
Batch Opt. (last 10%)	1.01m	0.45m
Batch Opt. (best 10%)	0.56m	0.20m
Batch Opt. (full path)	0.79m	0.33m
Spectral SLAM (worst 10%)	1.01m	0.51m
Spectral SLAM (last 10%)	0.98m	0.51m
Spectral SLAM (best 10%)	0.59m	0.22m
Spectral SLAM (full path)	0.79m	0.35m
Spectral + Batch Optimization (worst 10%)	**0.89m**	**0.40m**
Spectral + Batch Optimization (last 10%)	**0.81m**	**0.32m**
Spectral + Batch Optimization (best 10%)	**0.54m**	**0.18m**
Spectral + Batch Optimization (full path)	**0.69m**	**0.30m**

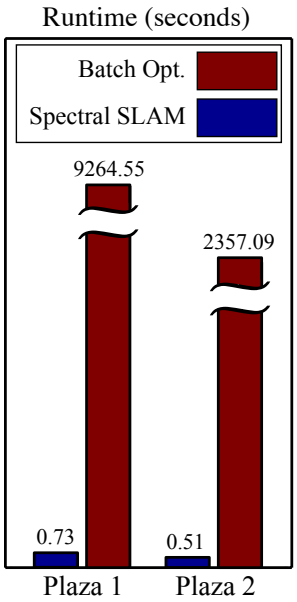

Figure 3. Comparison of Range-Only SLAM Algorithms. The table shows Localization RMSE. Spectral SLAM has localization accuracy comparable to batch optimization on its own. The best results (boldface entries) are obtained by initializing nonlinear batch optimization with the spectral SLAM solution. The graph compares runtime of Gauss-Newton batch optimization with spectral SLAM. Empirically, spectral SLAM is 3-4 *orders of magnitude* faster than full batch optimization of the likelihood on the autonomous lawnmower datasets.

Ferris, Brian, Fox, Dieter, and Lawrence, Neil. WiFi-SLAM using Gaussian process latent variable models. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pp. 2480–2485, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

Hsu, Daniel, Kakade, Sham, and Zhang, Tong. A spectral algorithm for learning hidden Markov models. In *COLT*, 2009.

Kanade, T. and Morris, D.D. Factorization methods for structure from motion. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1740):1153–1173, 1998.

Kantor, George A and Singh, Sanjiv. Preliminary results in range-only localization and mapping. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA '02)*, volume 2, pp. 1818 – 1823, May 2002.

Kehagias, Athanasios, Djugash, Joseph, and Singh, Sanjiv. Range-only SLAM with interpolated range data. Technical Report CMU-RI-TR-06-26, Robotics Institute, May 2006.

Kurth, Derek, Kantor, George A, and Singh, Sanjiv. Experimental results in range-only localization with radio. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, volume 1, pp. 974 – 979, October 2003.

Montemerlo, Michael, Thrun, Sebastian, Koller, Daphne, and Wegbreit, Ben. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 593–598. AAAI, 2002.

Rosencrantz, Matthew, Gordon, Geoffrey J., and Thrun, Sebastian. Learning low dimensional predictive representations. In *Proc. ICML*, 2004.

Siddiqi, Sajid, Boots, Byron, and Gordon, Geoffrey J. Reduced-rank hidden Markov models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-2010)*, 2010.

Thrun, Sebastian, Burgard, Wolfram, and Fox, Dieter. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

Tipping, Michael E. and Bishop, Chris M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

Tomasi, Carlo and Kanade, Takeo. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9: 137–154, 1992.

Triggs, B. Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pp. 845–851. IEEE, 1996.

Van Overschee, P. and De Moor, B. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer, 1996.

Yairi, Takehisa. Map building without localization by dimensionality reduction techniques. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pp. 1071–1078, New York, NY, USA, 2007. ACM.