A Spectral Learning Approach to Range-Only SLAM

Byron Boots
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
beb@cs.cmu.edu

Geoffrey J. Gordon
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
ggordon@cs.cmu.edu

1 Introduction

In range-only Simultaneous Localization and Mapping (SLAM), we are given a sequence of range measurements from a robot to fixed landmarks. 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; Thrun et al., 2005), multiple-hypothesis trackers (including particle filters and multiple EKFs/EIFs) (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.

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-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 desirable guarantees including statistical consistency and no local optima.

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.

2 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:

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

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

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

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

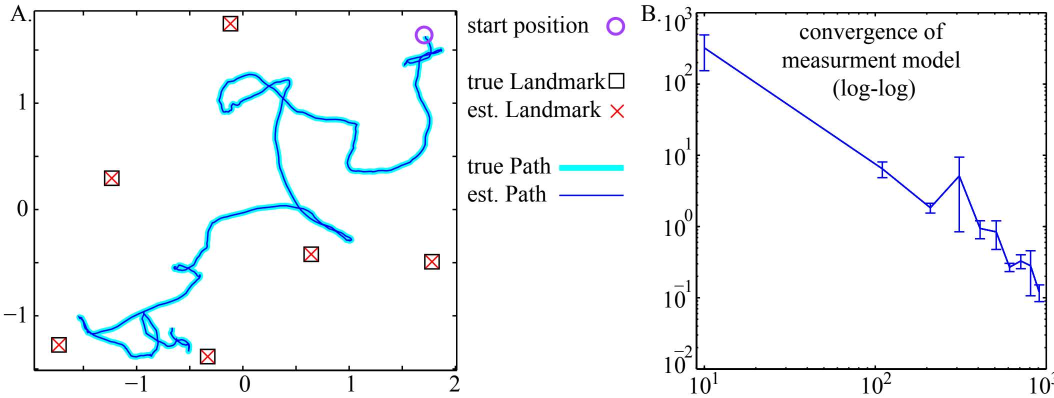

Figure 1: Spectral SLAM on simulated data. A.) Randomly generated landmarks (6 of them) and robot path through the environment (500 timesteps). A SVD of the squared distance matrix recovers a linear transform of the landmark and robot positions. Given the coordinates of 4 landmarks, we can recover the 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.

of landmarks, and $X \in \mathbb{R}^{4\times T}$ contains the positions of the robot over time:

$$C = \begin{bmatrix} (m_{1,x}^2 + m_{1,y}^2)/2 & m_{1,x} & m_{1,y} & 1 \\ (m_{2,x}^2 + m_{2,y}^2)/2 & m_{2,x} & m_{2,y} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ (m_{N,x}^2 + m_{N,y}^2)/2 & m_{N,x} & m_{N,y} & 1 \end{bmatrix} \quad X = \begin{bmatrix} 1 & \dots & 1 \\ -x_1 & \dots & -x_T \\ -y_1 & \dots & -y_T \\ (x_1^2 + y_1^2)/2 & \dots & (x_T^2 + y_T^2)/2 \end{bmatrix} \tag{3}$$

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 $Y = CX = CS^{-1}SX$. 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. 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. It turns out that Eq. (3) provides enough additional geometric constraints to do so: in the tech. report version of this paper (Boots & Gordon, 2012) 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 (3). (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 A Spectral SLAM Algorithm

The matrix factorizations of Sec. 2 suggests a straightforward SLAM algorithm, Alg. 1: build an empirical estimate $\widehat{Y}$ of Y by sampling observations as the robot traverses its environment, then apply a rank $k = 4$ thin SVD, discarding the remaining singular values to suppress noise.

$$\langle \widehat{U}, \widehat{\Lambda}, \widehat{V}^\top \rangle \leftarrow \text{SVD}(\widehat{Y}, 7) \tag{4}$$

Following Section 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.

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. 1)
2: Find the the top 4 singular values and vectors: $\langle \widehat{U}, \widehat{\Lambda}, \widehat{V}^\top \rangle \leftarrow \text{SVD}(\widehat{Y}, 4)$
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 $\widehat{C}$ to $\mathcal{C}$) or metric upgrade (see technical report (Boots & Gordon, 2012)).

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 technical report (Boots & Gordon, 2012), 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:

$$||\sin\Psi||_2 \leq \frac{Nc\sqrt{\frac{2\log(T)}{T}}}{\gamma} \tag{5}$$

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.

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 data sets collected from an autonomous lawn mowing robot (Djugash & Singh, 2008), 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. (Additional details on the experimental setup can be found in (Djugash & Singh, 2008).) 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. Then we applied the spectral SLAM algorithm of Section 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

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

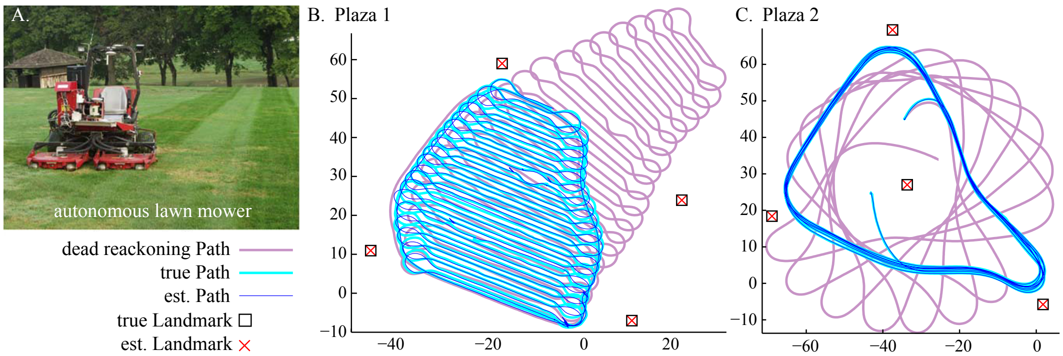

Figure 2: The autonomous lawn mower and spectral SLAM. A.) The robotic lawn mower platform. B.) In the first experiment, the robot traveled 1.9km receiving 3,529 range measurements. This path minimizes the effect of heading error by balancing the number of left turns with an equal number of right turns 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 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.

qualitative results, we quantitatively compared spectral SLAM to a number of different competing range-only SLAM algorithms. Spectral SLAM produced results that are *at least* as accurate as (and sometimes much more accurate than) competing methods while having better theoretical properties. Spectral SLAM is *statistically consistent* and fast, and the bulk of the computation is the fixed-rank SVD, so the time complexity of the algorithm is $O((2N)^2T)$ where N is the number of landmarks and T is the number of time steps. A detailed discussion of these results including a large table comparing different algorithms quantitatively can be found in (Boots & Gordon, 2012).

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 method that is closely related to SfM and spectral approaches to system identification. We provide theoretical guarantees for our algorithm, discuss how to derive an online algorithm, and show how to generalize to a full robot system identification algorithm. Finally, we demonstrate that our spectral approach to SLAM beats other state-of-the-art SLAM approaches on real-world range-only SLAM problems.

References

Biggs, M., Ghodsi, A., Wilkinson, D., & Bowling, M. (2005). Action respecting embedding. *In Proceedings of the Twenty-Second International Conference on Machine Learning* (pp. 65–72).

Boots, B., & Gordon, G. J. (2012). A spectral learning approach to range-only slam. *CoRR*, *abs/1207.2491*.

Boots, B., Siddiqi, S. M., & Gordon, G. J. (2010). Closing the learning-planning loop with predictive state representations. *Proceedings of Robotics: Science and Systems VI*.

Djugash, J., & Singh, S. (2008). A robust method of localization and mapping using only range. *International Symposium on Experimental Robotics*.

Ferris, B., Fox, D., & Lawrence, N. (2007). WiFi-SLAM using Gaussian process latent variable models. *Proceedings of the 20th international joint conference on Artifical intelligence* (pp. 2480–2485). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

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

Kantor, G. A., & Singh, S. (2002). Preliminary results in range-only localization and mapping. *Proceedings of the IEEE Conference on Robotics and Automation (ICRA '02)* (pp. 1818 – 1823).

Kehagias, A., Djugash, J., & Singh, S. (2006). *Range-only SLAM with interpolated range data* (Technical Report CMU-RI-TR-06-26). Robotics Institute.

Kurth, D., Kantor, G. A., & Singh, S. (2003). Experimental results in range-only localization with radio. *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)* (pp. 974 – 979).

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics (intelligent robotics and autonomous agents)*. The MIT Press.

Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, *9*, 137–154.

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

Van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems: Theory, implementation, applications*. Kluwer.

Yairi, T. (2007). Map building without localization by dimensionality reduction techniques. *Proceedings of the 24th international conference on Machine learning* (pp. 1071–1078). New York, NY, USA: ACM.