Robust GPS/INS-aided localization and mapping via GPS bias estimation

Paul Vernaza and Daniel D. Lee[1]

University of Pennsylvania, Philadelphia, PA 19104
{vernaza,ddlee}@seas.upenn.edu

Summary. We consider the problem of pose estimation in the context of outdoor robotic mapping. In such cases absolute position information from GPS is often available, making a full-blown SLAM implementation largely unnecessary. However, the peculiarities of GPS can lead to problems when using it in conjunction with a naive mapping system, as unpredictable biases tend to cause significant inconsistencies in the generated maps. We thus present a two-stage pose estimation system to address this problem. The first stage consists of a best-effort "blind" pose estimator based on a robust and extensible Rao-Blackwellized particle filtering framework. The estimate from this stage is then fed to a "seeing" HMM-style filter that attempts to infer the uncorrected bias of the first stage by matching stereo maps under an assumption of scene rigidity. Results are shown that demonstrate a vast improvement in pose estimates and map consistency using this method over the naive approach.

1 Motivation

Our work is motivated by the desire to create an autonomous robotic system that is able to navigate across unknown terrain as quickly and reliably as possible. Such is the scope of the Learning Applied to Ground Robots (LAGR) project, sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA). In the type of scenario studied in LAGR, our robot is required to complete multiple trips between two GPS waypoints, with no initial knowledge of the intervening terrain. We would expect the first trip to be completed in the greatest amount of time, as the robot would inevitably get caught in cul-de-sacs, necessitating backtracking and further exploration. After the first trip, with a complete map, we would expect no such delays. However, in our experiments we have discovered more often than not that the subsequent trips are even more delayed than the initial one. Analysis usually reveals that this phenomenon is caused by map and pose inconsistency between the two trips. Obstacles observed at a certain position on the first trip are observed at another position on the second trip, leading to a map filled with phantom obstacles and hence impeding progress.

These experiments were conducted using a system with separate localization and mapping components. GPS, inertial, and odometric sensors were first used to form a fused pose estimate. Given this pose information, local occupancy grids obtained from stereo vision were then overlaid into a globally-referenced occupancy grid used for navigation. The lack of feedback from the mapping process back into the pose estimation is what enabled the types of failures we observed.

In light of this, an obvious approach is to leverage existing techniques for performing simultaneous localization and mapping (SLAM), which perform the kind of feedback necessary to address the consistency issues. However, these approaches have drawbacks as well, including implementation difficulty, computational complexity, and difficulty in integrating GPS and INS measurements. This is primarily what led to our seeking a solution that would address the consistency issues associated with the naive approach while avoiding the complexities of SLAM.

We thus present our own version of this “GPS/INS and mapping” paradigm that addresses many of the failings present in the traditional approach. We decompose the problem into two parts. The first is to generate a “blind” estimate using an algorithm that is best suited for sensors such as GPS, inertial measurements, and odometry. This estimate makes use of a recent application of a Rao-Blackwellized particle filter that performs well in situations with large amounts of uncertainty and sensor drop-out, while simultaneously admitting an implementation that avoids many of the difficulties inherent in traditional GPS/INS integration. We then assume this estimate is biased in a way that is unobservable without taking vision into account. The second component then uses vision to attempt to estimate this bias. We will show in the rest of this paper how this division leads to a method for pose estimation and mapping that has practical advantages over both traditional GPS/INS solutions and SLAM-based approaches.

2 Previous Work

Typical approaches to GPS/INS integration are mainly based on nonlinear variants of the Kalman filter [van der Merwe and Wan, 2004]. However, the assumptions upon which these are based tend to fail in cases of large uncertainty and noisy sensors. In this work, we apply a Rao-Blackwellized particle filter with sampled orientations to address these issues. This aspect of the approach is summarized later in this paper and is described in detail in [Vernaza and Lee, 2006].

Another characteristic of traditional approaches to GPS/INS integration is that they usually do not address the issue of mapping, an issue that has relatively recently become one of the cornerstones of research in robotics. Approaches to mapping might be grouped roughly into those that assume known poses and those that do not (see [Thrun, 2002] for a comprehensive review of mapping techniques). When poses are assumed known, a typical approach is to build a probabilistic, grid-based representation of the environment known as an occupancy grid.

Occupancy grids have seen widespread adoption due to their simplicity and practicality. They can be made very dense and can be scaled up to very large sizes with negligible effect on computational efficiency. However, a significant drawback to occupancy grids is that maps can quickly become inconsistent if poses are assumed known when they are in fact uncertain, as is most often the case. This becomes quite a significant obstacle when it comes to autonomous navigation, as a common goal in autonomous navigation is to benefit from experience. Methods do exist for enforcing consistency in occupancy grids. Thrun provides one such method using expectation maximization [Thrun, 2003]. Unfortunately, this is an offline, computationally intensive procedure, making it unsuitable for autonomous navigation.

Autonomous navigation demands a solution that is both online and results in consistent maps. Simultaneous localization and mapping (SLAM) algorithms generally attempt to achieve both these goals by explicitly maintaining probability distributions that correlate positions of landmarks and vehicle pose. They are thus able to create consistent estimates of vehicle pose and landmark locations without an explicit absolute position sensor. However, doing so comes at a cost in both computational efficiency and ease of implementation.

Our basic motivation in this work is that the simple occupancy grid approach nearly works for GPS-aided mapping. The presence of an absolute position sensor renders a full-blown SLAM implementation largely unnecessary, since the robot's pose can typically be resolved to within error bounds of several meters with no need for any sort of visible landmark. However, we have already mentioned the caveats involved in using these uncertain estimates in a naive occupancy grid mapper. Our approach addresses the most fundamental limitation of the naive method while avoiding the complexity and other issues involved in typical SLAM. It is computationally efficient, and its computational complexity does not scale with the size of the map. However, like SLAM, it is also probabilistic and explicitly attempts to enforce map consistency in an online manner.

3 Pose estimation via GPS/INS

We first consider the problem of "blind" pose estimation without taking the map into account. We accomplish this by first considering the case where we are only concerned with orientation estimation and later considering how to integrate translational state estimation as well.

3.1 Orientation Estimation via Particle Filtering

As alluded to earlier, we take a particle filtering approach to attitude estimation. By choosing a particle filter we immediately avoid most of the representational issues that plague attitude estimators based on the Kalman filter, since there is no need to define an analytical distribution over orientations. Instead, we will represent our orientation distribution by a set of discrete particles.

In order to apply a simple particle filter, we need only to define how to sample from a state transition distribution $p(x_t|x_{t-1})$ (assuming x_t is the state at time t) and how to evaluate a measurement likelihood probability $p(z_t|x_t)$ (where z_t is the observation at time t). We then sample from the state transition distribution in order to obtain a proposal distribution. The samples in the proposal distribution are weighted according to Eq. (1) [Arulampalam et al., 2002], where $w_t^{(i)}$ is the weight of the ith sample at time t.

$$w_t^{(i)} = p(z_t|x^t, z^{t-1})w_{t-1}^{(i)} \tag{1}$$

Given that an inertial angular rate sensor is available, an obvious choice for the state transition density is based on the rotation of the inertial unit in time. Sampling from the state transition distribution thus amounts to rotating each particle by an amount specified by the inertial unit, adding small peturbations in order to reflect uncertainty in the angular rate measurement. In the absence of significant external forces besides gravity, an inertial unit's accelerometers can be used a as tilt sensor. We can therefore calculate a measurement likelihood probability by defining a Gaussian distribution on the difference between the expected gravity measurement and the actual measurement, given the orientation of each particle. Defining the state transition and measurement likelihood distributions in this way, it is straightforward to implement an attitude estimator.

3.2 Estimating Translation States via Rao-Blackwellization

Rao-Blackwellization [Doucet et al., 2000] provides a simple and computationally efficient way to include the translational states in the estimator. Given each sampled orientation, we might adequately describe the translational state distribution as Gaussian. Furthermore, we can write down linear equations for the translational state dynamics and observation models given the sampled orientation. It is therefore possible to update these Gaussians using the Kalman filter.

In order to successfully integrate this approach with the particle filtering framework, it is necessary to calculate an appropriate importance weight for each particle given observations on the translational state. The form of the importance weights can be derived in a manner very similar to the way the Kalman filter is derived; in fact, the importance weights turn out to be the likelihood of the measurement under the expected measurement distribution, whose statistics are already calculated as part of the Kalman filter update. Details on the calculation of the importance weights are given in [Vernaza and Lee, 2006].

4 Bias estimation for consistent mapping

Thus far we have only discussed the first part of the estimator. Now we turn our attention to the problem of modeling and estimating the residual bias present in the first estimate. We will assume that the majority of this bias arises from the fact that we assumed a simple Gaussian white noise model for the GPS position error. However, it is clear at the very least that the error is strongly time-correlated. Qualitatively, we might describe the error as slowly drifting in a random way. This description is captured in the following simple state space model:

$$b_{t+1} = \begin{cases} b_t + \nu_t, \nu_t \sim \mathcal{N}(0, \mathbf{B}) & \|b_t\| < \sigma \\ (1-\epsilon) b_t & \|b_t\| \geq \sigma \end{cases} \tag{2}$$

Here b_t is a two-dimensional additive bias such that the true (2D) pose is given by the sum of the first estimator's output and the bias. The bias is characterized as drifting according to a Gaussian random walk with drift covariance $\mathbf{B}$. We also assume orientation has been resolved sufficiently well to ignore any additional estimable bias in it.

As mentioned earlier, in order to observe the bias it is necessary to employ some kind of measurement beyond the GPS/INS combination. For this purpose, we propose to use a measure of consistency between an instantaneously observed local map and a continuously accumulated global map. Specifically, we can employ the normalized cross-correlation between them. Again, for the sake of efficiency, we might partially relax the normalization requirement. This suggests the following observation model:

$$p(z_t | b_t = (x, y)) = \sigma(\kappa \sum_{a,b} z_t(a, b) M(x + a, y + b)) \tag{3}$$

Here $z_t(a, b)$ is a function that provides the value of the observed local map at position (a, b) with respect to the center of the local map (after rotation into the global reference frame). $M(x, y)$ is is the value of the map under the robot assuming that the first-stage pose estimate is biased by an amount equal to (x, y) (see Fig. (1)).

Since the value is to be interpreted as a probability, we use the sigmoid function $\sigma(z) = (1 + e^{-z})^{-1}$ and an appropriate scale factor κ to ensure that the result is between zero and one. Aside from κ, the expression inside the sigmoid function is simply the cross-correlation of z_t and M.

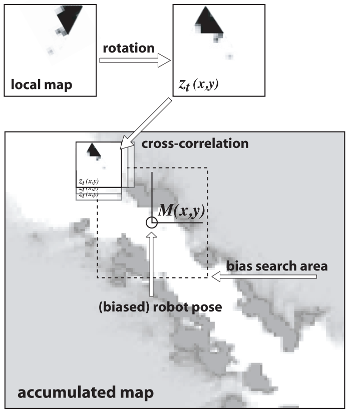

Fig. 1. Illustration of measurement model evaluation for secondary bias estimator

In order to estimate the bias from these observations, we propose exact inference via discretization of the state space. One motivation for this is the nature of the observations. Since we assumed that the bias would be bounded, discretization of the space of potential biases is easily accomplished. We should note that this idea is very similar in spirit to Markov localization (especially the correlation-based variant [Konolige and Chou, 1999]). Expressing the problem in the canonical form:

$$p(b_t|z^t) = \alpha p(z_t|b_t) \sum_{b_{t-1}} p(b_t|b_{t-1}) p(b_{t-1}|z^{t-1}) \tag{4}$$

This suggests an update rule that is $O(N^2)$ in the number of distinct biases we would like to evaluate. However, if the transition matrix that specifies $p(b_t|b_{t-1})$ is relatively sparse, the asymptotic complexity will be much reduced. Fortunately, this is the case for our scenario. Given the Gaussian random walk model in (2), $p(b_t|b_{t-1})$ is given by a Gaussian distribution with mean b_{t-1}. Given that we are dealing with a finite, discretized state space, we can ultimately implement $p(b_t|b_{t-1})$ as a look-up table T such that $p(b_t|b_{t-1}) = T(b_t - b_{t-1})$. Since the vast majority of the mass of the Gaussian is contained within a few standard devations of the mean, the look-up table can be made fairly small. Substituting this into (4) yields:

$$p(b_t|z^t) = \alpha p(z_t|b_t) \sum_{b_{t-1}} T(b_t - b_{t-1}) p(b_{t-1}|z^{t-1}) \tag{5}$$

The inner summation evaluated at each b_t is thus equivalent to convolution of the prior with a small kernel. The fact that both the observation and process models can be implemented as discrete convolutions is appealing not only for its simplicity, but also due to the availability of special purpose hardware and techniques that could be employed to perform these convolutions at high speed.

```
while $z_t$ = new local map (rotated to global frame) do
    M: the global map centered on zero bias
    $\star$: the cross-correlation operator
    $p(z_t|b_t) \leftarrow (z_t \star M)(b_t)$
    T: Gaussian kernel
    $*$: the convolution operator $p(b_t|z^{t-1}) \leftarrow \sigma(\kappa(T * p(b_{t-1}|z^{t-1}))(b_t))$
    pairwise multiplication of sequences: $p(b_t|z^t)/\alpha \leftarrow p(z_t|b_t)p(b_t|z^{t-1})$
    $\alpha \leftarrow (\sum_{b_t} p(b_t|z^t)/\alpha)^{-1}$
    $p(b_t|z^t) \leftarrow \alpha(p(b_t|z^t)/\alpha)$
end
```

Algorithm 1: GPS bias estimator

5 Implementation Results

For simplicity, we tested the blind and seeing components of the estimator separately. We first ran the blind component on simulation data, producing results summarized in table (1), where $E|\tilde{x}|$ represents mean absolute error, and $\sqrt{E|\tilde{x}^2|}$ represents the standard deviation of the absolute error. Different sets of sensors were enabled in different scenarios to test robustness in conditions of sensor unavailability. A "dead-reckoning" case employing only gyroscopes and odometry is provided as a control. As expected, results were best with all sensors enabled, yielding a very small mean error. However, disabling other sensors had a comparatively minimal effect on performance.

	Attitude error (deg)		Position error (m)	
Scenario	$\mathbf{E}\|\tilde{x}\|$	$\sqrt{\mathbf{E}\|\tilde{x}^2\|}$	$\mathbf{E}\|\tilde{x}\|$	$\sqrt{\mathbf{E}\|\tilde{x}^2\|}$
Dead-reckoning	125.39	32.85	32.79	15.94
All sensors (no magnetometer)	4.86	3.98	1.04	0.39
No GPS position	7.09	4.68	3.42	1.84
No GPS velocity or accelerometers	11.82	6.71	1.40	0.70

Table 1. Summary of simulation results

We then tested the blind estimator again by running it onboard a small, remote-controlled ground robot equipped with GPS, odometry, and an inertial measurement unit (IMU). The test scenario consisted of driving the robot in a closed-loop path around a relatively open field. The path was approximately 250 meters long. Reported GPS error averaged about 11 meters. We ran the filter twice on the same data: once with all sensors enabled, and once with GPS velocity and accelerometer measurements disabled. The resulting 2D position tracks are displayed in Fig. (2). Closing the loop without GPS position measurements resulted in a 2.41 meter error at the end of the loop, while closing the loop with all sensors resulted in a 1.45 meter error. Raw GPS closed the loop with a 2.33 meter error.

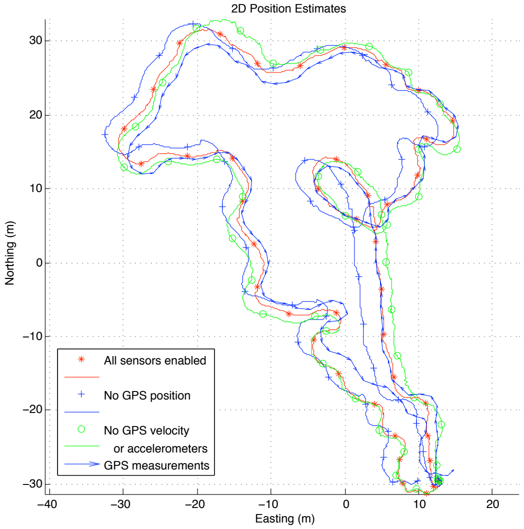

Fig. 2. Position estimates obtained by running pose filter on data obtained from real sensors. Different tracks indicate different scenarios.

We tested the performance of the second-stage estimator in a separate, offline procedure. We first ran our robot along a certain course, logging pose-tagged images of the course along the way. We repeated this procedure, obtaining a second set of images. The first set of images was then processed to create stereo disparity maps, which we combined with logged pose to build a map of the course. The resulting map is displayed in the leftmost figure of Fig. (3).

We then repeated the map-building process with the second set of images, but with a few changes. First, we initialized the procedure with the first map, allowing views from the second to overwrite it where available. Second, we added an artificial, drifting bias to the pose estimate before placing the views in the second map. This allowed us to simulate the effect of a randomly drifting GPS bias while also knowing its precise value. The logs used were specifically chosen to minimize apparent GPS drift, so that any bias in the pose would be dominated by the artificially injected one. The map built as a result of this procedure is displayed in the center figure of Fig. (3). Finally, the last procedure was repeated with the secondary bias estimator enabled. The resulting map is displayed in the rightmost figure of Fig. (3).

It is clear from Fig. (3) that the secondary bias estimation has a dramatic effect on mapping performance. The most common symptom of GPS drift is a map such as that in the center of Fig. (3), in which a shifted view of the corridor appears alongside the view generated in the previous map. By contrast, the map generated with bias estimation in effect is sharp and detailed, indicating that the true bias was successfully resolved. Plots of the injected bias against the estimated bias for this experiment are displayed in Fig. (4). As expected, these indicate that the bias was tracked closely for most of the experiment, with the exception of a short period near the end of the course. This is also to be expected, since the course ends in a clearing with few features to match, which limits the estimator's ability to distinguish between different biases.

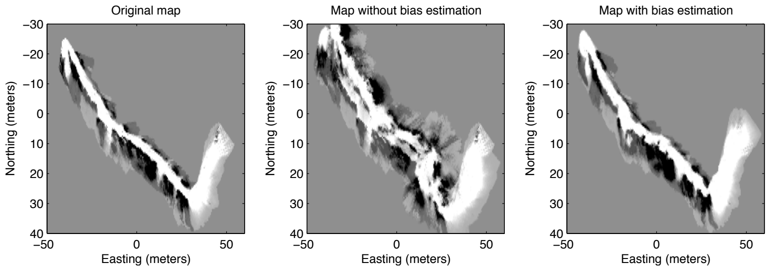

Fig. 3. Comparison of original map with those generated on a second traversal of the same course, with and without secondary bias estimation.

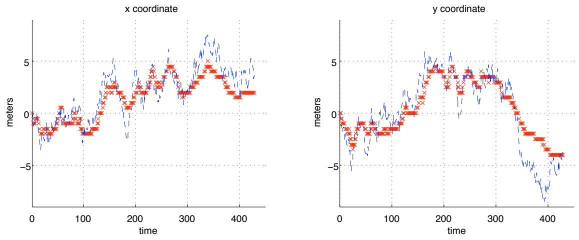

Fig. 4. Plots illustrating performance of secondary bias estimator in simulation. Dashed line indicates magnitude of artificially injected bias. Crossed line indicates estimated bias. Left and right figures indicate x and y components of bias, respectively.

6 Conclusions

We have presented a practical and comprehensive solution to the problem of pose estimation and mapping for an autonomous vehicle. Our approach combines the strengths of multiple inference techniques: for orientation, particle filtering; for translation, Kalman filtering; and simple discrete Bayesian filtering for GPS bias estimation.

Our experiments have shown that the filter is very robust with respect to conditions of high sensor uncertaity, initial state uncertainty, and sensor unavailability. We have also shown that it is feasible to track the GPS bias accurately with our method and to hence obtain a dramatic improvement in map consistency over the naive occupancy grid mapping approach. Finally, the method is efficient enough to run in real-time on modest hardware, and its complexity does not scale with map size.

We have noted that our approach may have more in common with simple "GPS/INS + occupancy grid" approaches than it does with full SLAM. Though there are advantages to this, it is also clear that there are many ideas from SLAM from which we could benefit. Thus, in the future we would like to further examine how SLAM-like features could be integrated with more traditional approaches to pose estimation. We feel that ultimately, a combination of the two is what will enable autonomous vehicles to succeed in real environments.

References

[Arulampalam et al., 2002] Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions of Signal Processing*, 50:174–188.

[Doucet et al., 2000] Doucet, A., de Freitas, N., Murphy, K., and Russell, S. (2000). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *The Sixteenth Conference on Uncertainty in Artificial Intelligence*.

[Konolige and Chou, 1999] Konolige, K. and Chou, K. (1999). Markov localization using correlation. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*.

[Thrun, 2002] Thrun, S. (2002). Robotic mapping: A survey. In Lakemeyer, G. and Nebel, B., editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.

[Thrun, 2003] Thrun, S. (2003). Learning occupancy grids with forward sensor models. *Autonomous Robots*.

[van der Merwe and Wan, 2004] van der Merwe, R. and Wan, E. (2004). Sigma-point Kalman filters for integrated navigation. In *60th Annual Meeting of the Institute of Navigation*.

[Vernaza and Lee, 2006] Vernaza, P. and Lee, D. D. (2006). Rao-Blackwellized particle filtering for 6-DOF estimation of attitude and position via GPS and inertial sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation*.