Robust Vehicle Localization in Urban Environments Using Probabilistic Maps

Jesse Levinson, Sebastian Thrun
Stanford Artificial Intelligence Laboratory
{jessel,thrun}@stanford.edu

***Abstract*— Autonomous vehicle navigation in dynamic urban environments requires localization accuracy exceeding that available from GPS-based inertial guidance systems. We have shown previously that GPS, IMU, and LIDAR data can be used to generate a high-resolution infrared remittance ground map that can be subsequently used for localization [4]. We now propose an extension to this approach that yields substantial improvements over previous work in vehicle localization, including higher precision, the ability to learn and improve maps over time, and increased robustness to environment changes and dynamic obstacles. Specifically, we model the environment, instead of as a spatial grid of fixed infrared remittance values, as a probabilistic grid whereby every cell is represented as its own gaussian distribution over remittance values. Subsequently, Bayesian inference is able to preferentially weight parts of the map most likely to be stationary and of consistent angular reflectivity, thereby reducing uncertainty and catastrophic errors. Furthermore, by using offline SLAM to align multiple passes of the same environment, possibly separated in time by days or even months, it is possible to build an increasingly robust understanding of the world that can be then exploited for localization.**

We validate the effectiveness of our approach by using these algorithms to localize our vehicle against probabilistic maps in various dynamic environments, achieving RMS accuracy in the 10cm-range and thus outperforming previous work. Importantly, this approach has enabled us to autonomously drive our vehicle for hundreds of miles in dense traffic on narrow urban roads which were formerly unnavigable with previous localization methods.

I. INTRODUCTION

Interest in autonomous vehicles and advanced driver assistance systems continues to increase rapidly. In particular, the DARPA Grand Challenges of 2004 and 2005, in which vehicles competed to autonomously navigate through desert terrain, and the DARPA Urban Challenge of 2007, in which vehicles competed to autonomously navigate through a mock urban environment amidst other traffic, have generated considerable enthusiasm and research interest in the field of autonomous driving [15], [16].

The 2007 Urban Challenge was the first significant demonstration of vehicles driving themselves through a city-like environment. As important as this result was, many simplifications were made for the purposes of the competition, and the course itself, while more difficult in many ways than previous challenges, featured roads wide enough to accomodate even military-sized vehicles.

Research into autonomous driving in real environments has been ongoing for many years, but much of it has focused on specific environments (e.g. highways with obvious lane markers) [3]. Some positive results have been achieved for map-based driving, though significant limitations remained. In order for a vehicle to handle a variety of environments, including ones with dense traffic, it must be able to localize itself in such situations without relying on particular patterns or features. In fact, in order to enable autonomous driving, a localization system should be able to handle situations where the environment has changed since the map was created.

Fig. 1. Our autonomous vehicle. Velodyne HD-LIDAR 64-beam scanner is circled in red.

In this paper we present a new method of map-based driving that extends previous work by considering maps as probability distributions over environment properties rather than as fixed representations of the environment at a snapshot in time. As in [4], we build infrared reflectivity maps of the environment and align overlapping portions of the same or disparate trajectories with GraphSLAM, using similar offline relaxation techniques to recent SLAM methods [2], [1], [5], [6], [8]. However, by extending the format of the map to encapsulate the probabilistic nature of the environment, we are able to represent the world more accurately and localize with fewer errors. In particular, previous methods often use a binary classification for deciding whether or not to incorporate a piece of evidence into a map. That is, a sensor reading such as a laser scan is either assumed to be a legitimate part of the map, or it is assumed to be spurious and therefore ignored. In [4], exactly this approach was used; any scans that were thought to be vertical according to a binary classification were thrown out.

However, instead of having to explicitly decide whether each measurement either is or is not part of the static envi-

ronment, we propose as an alternative considering the sum of all observed data and modeling the variances observed in each part of the map. This new approach has several advantages compared to the non-probabilistic alternative. First, while much research incorporating laser remission data has assumed surfaces to be equally reflective at any angle of incidence, this approximation is often quite poor [10], [11], [12]. Whereas Lambertian and retroreflective surfaces have the fortuitous property that remissions are relatively invariant to angle of incidence, angular-reflective surfaces such as shiny objects yield vastly different returns from different positions. Instead of ignoring these differences, which can lead to localization errors, we now implicitly account for them.

A further advantage of our proposed method is an increased robustness to dynamic obstacles; by modeling distributions of reflectivity observations in the map, dynamic obstacles are automatically discounted in localization via their trails in the map. Finally, in addition to capturing more information about the environment, our approach enables a remarkably straightforward probabilistic interpretation of the measurement model used in localization.

Our vehicle is equipped with an Applanix LV-420 tightly coupled GPS/IMU system that provides both intertial updates and global position estimates at 200 Hz. The environment is sensed by a Velodyne HD-LIDAR laser rangefinder with 64 separate beams; the entire unit spins at 10 Hz and provides approximately one million 3-D points and associated infrared intensity values per second. This vast quantity of data ensures that most map cells are hit multiple times, thereby enabling the computation of intensity variances on a per-cell basis.

We first present the details of our mapping algorithm, including a novel unsupervised laser calibration routine, and then explain how the localizer uses incoming data and a probabilistic map to localize the vehicle. Finally, we show results from several experiments that demonstrate the accuracy and robustness of our approach.

II. Probabilistic Maps

Our ultimate goal in building a map is to obtain a grid-cell representation of the observed environment in which each cell stores both the average infrared reflectivity observed at that location as well as the variance of those values. We generate such a map in three steps: first, we post-process all trajectories so that areas of overlap are brought into alignment; second, we calibrate the intensity returns of each laser beam so that the beams have similar response curves; and finally, we project the calibrated laser returns from the aligned trajectories into a high-resolution probabilistic map. Each of these steps is described in detail below.

A. Map alignment using GraphSLAM

Given one or more logfiles containing GPS, inertial, and laser data, we wish to refine the trajectories in order to bring areas of overlap into alignment. Specifically, when there exist sections of the logfiles that are spatially near but temporally separated, the vehicle's pose in these sections must be corrected so that the two sections align properly. This task has been successfully accomplished in previous work [4]. Briefly, GraphSLAM is used to optimize an objective function in which adjacent vehicle poses are linked by inertial and odometry data, vehicle poses are linked to their estimated global position, and matched sections from the logfile (e.g. for loop closure) are linked by their computed alignment offsets.

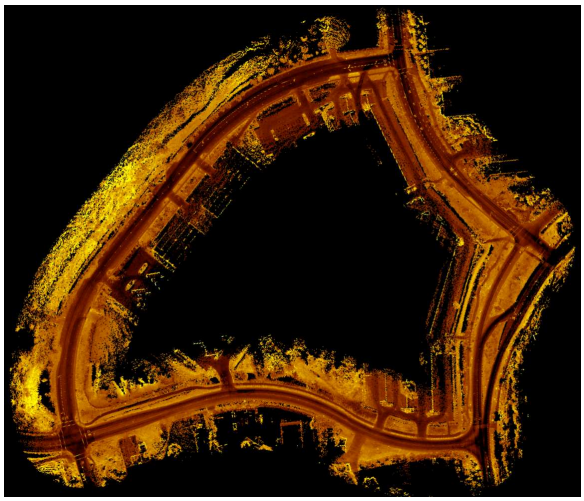


Fig. 2. An infrared refectivity map of a large urban block showing the average reflectivity of each 15x15cm cell. Many features are visible in this spectrum.

In this implementation, alignment offsets between matched poses are computed using iterative closest point on a cluster of 5 adjacent 360-degree laser scans from each of the two sections; the x, y, z, yaw, pitch, and roll are all optimized. Once a list of matches has been computed, the GraphSLAM objective function is minimized and the vehicle trajectories are updated accordingly.

B. Laser calibration

Before we generate our map, it is important to calibrate the separate laser beams so that they respond similarly to the objects with the same brightness. With a well calibrated laser, this step can be skipped without effect, but creating a probabilistic map from a poorly calibrated laser suffers from two disadvantages: first, the intensity averages for each cell will depend heavily on which beams happened to hit it; and second, the computed intensity variances will significantly overstate the reality. In practice, we find that it is not necessary to recalibrate our laser every time we use it, but using the factory calibration is unquestionably detrimental. Thus in our implementation the following calibration procedure can be performed only once and the resulting calibration table can be used for all further mapping and localization.

A single 360-degree scan from the uncalibrated laser can be seen in Fig. 3(a). As is readily apparent, some beams are generally too bright and others are generally too dark. Rather than compute a single parameter for each beam (which would still an improvement over the uncalibrated data), we instead compute an entire response curve for every

beam, so that we have a complete mapping function for each beam from observed intensity to output intensity. This more sophisticated approach is superior because, due to the particularities associated with the hardware of the laser, each beam has its own unique nonlinear response function.

Rather than using a fixed calibration target, we present an unsupervised calibration method that can be performed by driving once through an arbitrary environment. To compute the calibrated response functions for each beam, we project laser measurements from a logfile as the vehicle proceeds through a series of poses and, for every map cell, store the intensity values and associated beam ID for every laser hit to the cell. Then, for every 8-bit intensity value a observed by each of the beams j, the response value for beam j with observed intensity a is simply the average intensity of all hits from other beams to cells for which beam j returned intensity a.

Specifically, let T be the set of observations $\{z_1,\ldots,z_n\}$ where z_i is a four-tuple $\langle b_i, r_i, a_i, c_i\rangle$ containing the beam ID, range measurement, intensity measurement, and map cell ID of the observation, respectively. Then we have:

$$\begin{aligned} T &= \{z_1,\ldots,z_n\} \\ z_i &= \langle b_i, r_i, a_i, c_i\rangle \\ b_i &\in [0,\ldots,63] \\ r_i &\in \mathbb{R}_+ \\ a_i &\in [0,\ldots,255] \\ c_i &\in [0,\ldots,N\cdot M-1] \end{aligned}$$

where N and M are the dimensions of the map. Then the calibrated output $c(a,j)$ of beam j with observed intensity a is computed in a single pass as:

$$c(j,a) := \mathbb{E}_{z_i\in T}[a_i \mid ((\exists k : c_i = c_k, b_k = j, a_k = a), b_i \neq j)] \quad (1)$$

That is, the calibrated output when beam j observes intensity a is the conditional expectation of all other beams' intensity readings for map cells where beam j observed intensity a.

This equation is computed for all 64x256 combinations of j and a. Thus a calibration file is a 64-by-256 intensity mapping function; values which are not observed directly can be interpolated from those which are. These calibrated response functions need only be computed once and their results can be stored compactly in a lookup table. The result of calibration can be seen in Fig. 3(b). In contrast to many calibration algorithms, our unsupervised approach has the desirable property that it does not require a particular calibration environment, but instead adapts automatically to any environment. Due to the abundance of laser data and the averaging over many values, even the presence of dynamic objects does not significantly reduce the quality of the calibration.

C. Map creation

Given a calibrated laser and one or more logfiles with properly aligned trajectories, it is now possible to generate a high resolution map.

(a) Uncalibrated sensor.

(b) After intensity calibration.

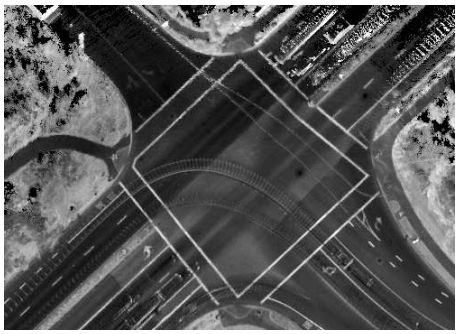
Fig. 3. Without calibration (a), each of the 64 beams has a significantly different response to reflectivity values. After calibration (b), the beams are much better matched.

In order to create a probabilistic map, we store not only the average laser intensity for each map cell, but also the variance of those values. Thus, the map has two channels of data: an intensity, and a variance. Explicitly estimating the actual reflectance properties as [13] do with multiple photographs is not necessary for our approach; simply accounting for the observed variance of infrared remittance values at each map cell is enough for our mapping and localization goals.

The algorithm for generating a probabilistic map is straightforward. As the vehicle transitions through its series of poses, the laser points are projected into an orthographic x,y representation in which each map cell represents a 15x15cm patch of ground. Every cell maintains the necessary intermediate values to update its intensity average and variance with each new measurement. Infrared reflectivity is an rich source of environmental data, as can be seen in Fig. 4(a). Unlike camera-based data, this data is immune from shadows and other artifacts caused by passive lighting. Upon close inspection, trails from passing cars can be seen; because the maps are simply averages over time, observations of dynamic obstacles will taint the map. Rather than attempt to delete these, which is impossible to do perfectly, we instead take advantage of the fact that dynamic obstacles tend to leave a signature by causing large intensity variances for the cells in which they pass.

Indeed, Fig. 4(b) shows the standard deviations of each cell, in which the dynamic trails stand out very visibly. Here, the probabilistic map encodes the fact that its intensity estimation for those cells is uncertain, so that when the map

(a) Average infrared reflectivity.



(b) Standard deviation of infrared reflectivity values.

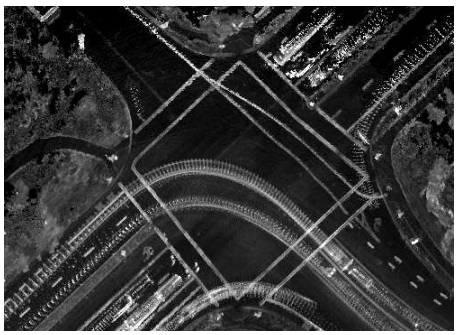


Fig. 4. The two channels of our probabilistic maps. In (a) we see the average infrared reflectivity, of brightness, of each cell. The innovation of this paper is to also consider (b), the extent to which the brightness of each cell varies. Note that the trails of the passing vehicles are much more prominent in (b).

is later used for localization, intensity values from incoming sensor data that do not match those in the map will not be overly punished. It is also interesting to note that, while it appears at first glance that the lane markings also have high variances, upon closer inspection it is clear that it is actually the edges of the markings which have high variance. This observation is easily explained by the fact that slight sensor miscalibrations and pose errors will cause the cells near a large gradient to have larger range of intensity returns.

III. Online Localization

Once we have built a map of the environment, we can use it to localize the vehicle in real time. We represent the likelihood distribution of possible x and y offsets with a 2-dimensional histogram filter. [1] As usual, the filter is comprised of two parts: the motion update, to reduce confidence in our estimate based on motion, and the measurement update, to increase confidence in our estimate based on sensor data.

[1]Although particle filters are a popular alternative method, having GPS available allows us to constrain our search space to within several meters of the GPS estimate and thus to calculate directly the probability of all possible offsets at a 15-cm cell size. This method confers the significant advantage of not having to worry that a particle is missing near the correct location. Further, the possibility of achieving accuracy much better than the size of one grid cell would afford us no additional advantage given the other sources of error in our system (e.g. sensor miscalibration, controller error, etc.)

A. Motion update

Our GPS/IMU system reports both inertial updates and a global position estimate at 200Hz. By integrating the inertial updates we maintain a "smooth coordinate" system which is invariant to jumps in GPS pose but which, necessarily, diverges arbitrarily over time. Fortunately, because the smooth coordinate system is updated by integrating velocities, its offset from the true global coordinate system can be modeled very accurately by a random walk with Gaussian noise. Of course, recovering the offset between the two coordinate frames is equivalent to knowing our true global position, so it is this offset that we strive to estimate.

As a result, the motion model for our filter is surprisingly simple; rather than needing to model the uncertainty of the motion of the vehicle itself as is typically done, we need only model the drift between the smooth and global coordinate systems. We note that the car's motion model is not actually being ignored; rather, it is used internally in the tightly-coupled GPS/IMU system precisely to minimize the rate at which the smooth and global coordinate systems drift apart. Vehicle dynamics are discussed in [9].

Because the smooth coordinate system's drift is modeled as a Gaussian noise variable with zero-mean, the motion model updates the probability of each cell as follows:

$$\overline{P}(x,y) = \eta \cdot \sum_{i,j} P(i,j) \cdot \exp\left(-\frac{1}{2}(i-x)^2(j-y)^2/\sigma^2\right) \qquad (2)$$

where $\overline{P}(x,y)$ is the posterior probability, after the motion update, that the vehicle is in cell (x,y), η is the normalizing constant, and σ is the parameter describing the rate of drift of the smooth coordinate system.

Although this update is theoretically quadratic in the number of cells, and thus quartic in the search radius, because the drift rate is relatively low and the update frequency can be arbitrarily high, it is in practice perfectly acceptable to only consider consider neighboring cells with a distance of two or three from the cell to be updated. For instance, the probability that the smooth coordinate frame drifts more than 45cm in .1 seconds is vanishingly small, even though such jumps are relatively common for the global GPS estimate. We process the motion update at a rate proportional to the speed of the vehicle, as the expected drift in the smooth coordinate system is roughly proportional to the magnitude of the vehicle's velocity.

B. Measurement update

The second component of the histogram filter is the measurement update, in which incoming laser scans are used to refine the vehicle's position estimate.

The way in which we process incoming laser scans is identical to the mapping process described in the previous section. That is, rather than treating every laser return as its own observation, we instead build a rolling grid from accumlated sensor data in the exact same form as our map. This method enables us to directly compare cells from our sensor data to cells from the map, and avoids overweighting

cells which have a high number of returns (e.g. trees and large dynamic obstacles).

If z is our sensor data and m is our map, and x and y are possible offsets from the GPS pose, then Bayes' Rule gives:

$$P(x,y|z,m) = \eta \cdot P(z|x,y,m) \cdot P(x,y) \tag{3}$$

We may approximate the uncertainty of the GPS/IMU pose estimate by a Gaussian with variance σ^2_{GPS}, so we may estimate $P(x,y)$ simply as a product of the GPS Gaussian and the posterior belief after the motion update:

$$P(x,y) = \eta \cdot \exp\left(\frac{x^2+y^2}{-2\sigma^2_{GPS}}\right) \cdot \overline{P}(x,y) \tag{4}$$

To calculate the probability of sensor data z given an offset (x,y) and map m, we take the product over all cells of the probability of observing the sensor data cell's average intensity given the map cell's average intensity and both of their variances. This value is then raised to an exponent $\alpha < 1$ to account for the fact that the data are likely not entirely independent, for example due to systemic calibration errors or minor structural changes in the environment which are measured in multiple frames. [7]

Let us call the two-dimensional arrays of the standard deviations of the intensity values in the map and sensor data m_σ and z_σ, respectively. Then, for example, the standard deviation of the intensity values in the map cell .45m east and 1.2m north of the GPS estimate would be expressed as $m_{\sigma_{(.45,1.2)}}$.

We will use the same notation for the average intensity value, with r denoting the average intensity (reflectivity) of the cell. Again, to use the same example, the average intensity seen in the map at the cell .45m east and 1.2 north of the GPS estimate would be expressed as $m_{r_{(.45,1.2)}}$.

Thus, we have:

$$P(z|x,y,m) = \prod_{i,j} \exp\left(\frac{-(m_{r_{(i-x,j-y)}} - z_{r_{(i,j)}})^2}{2(m_{\sigma_{(i-x,j-y)}} + z_{\sigma_{(i,j)}})^2}\right)^\alpha \tag{5}$$

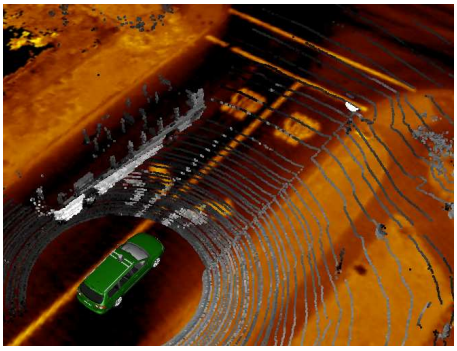
Putting it all together, we obtain:

$$P(x,y|z,m) = \eta \cdot \prod_{i,j} \exp\left(\frac{-(m_{r_{(i-x,j-y)}} - z_{r_{(i,j)}})^2}{2(m_{\sigma_{(i-x,j-y)}} + z_{\sigma_{(i,j)}})^2}\right)^\alpha \cdot \exp\left(\frac{x^2+y^2}{-2\sigma^2_{GPS}}\right) \cdot \overline{P}(x,y) \tag{6}$$

Towards the end of achieving robustness to partially outdated maps, we further impose a minimum on the combined standard deviation for the intensity values of the map and sensor data. This implicitly accounts for the not-unlikely phenomenon that an environment change since the map acquisition simultaneously enabled a low variance in both the map and the sensor data, yet with the two showing significantly different intensity values.

For computational reasons we restrict the computation of $P(x,y|z,m)$ to cells within several meters of the GPS estimate; however, this search radius could easily be increased if a less accurate GPS system were used.

(a) GPS localization induces ≥1 meter of error.



(b) No noticeable error after localization.

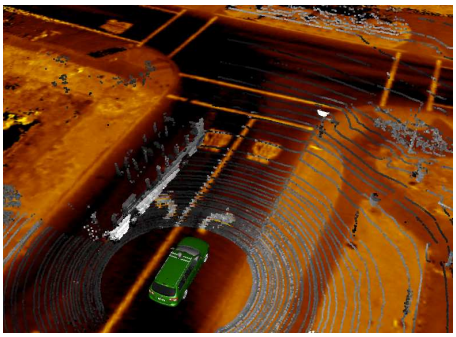


Fig. 5. Incoming laser scans (grayscale) superimposed on map (gold). (a) GPS localization is prone to error, even (as shown here) with a high-end integrated inertial system and differential GPS using a nearby stationary antenna. (b) With localization there is no noticeable error, even in the presence of large dynamic obstacles such as this passing bus.

C. Most likely estimate

Given the final posterior distribution, the last step is to select a single x and y offset that best represents our estimation. Taking the offset to be $\max_{x,y} P(x,y)$ is, by definition, probabilistically optimal at any given instant, but such an approach could add unnecessary danger as part of the pipeline in an autonomous vehicle. Because the maximum of a multimodal distribution can easily jump around discontinuously, using that approach may cause the vehicle to oscillate under unfortuitous circumstances, even if the vehicle's navigation planner performed some variety of smoothing. An alternative approach would be to choose the center of mass of the posterior distribution; this would improve consistency, but would tend to cause the chosen offset to be biased too much towards the center. As a compromise, we use the center of mass with the variation that we raise $P(x,y)$ to some exponent $\alpha > 1$, as follows:

$$x = \frac{\sum_{x,y} P(x,y)^\alpha \cdot x}{\sum_{x,y} P(x,y)^\alpha} \qquad y = \frac{\sum_{x,y} P(x,y)^\alpha \cdot y}{\sum_{x,y} P(x,y)^\alpha} \tag{7}$$

This (x,y) offset is the final value which is sent to the vehicle's navigation planner. While an advanced planning and decision-making algorithm could take advantage of the entire posterior distribution over poses rather than requiring a single, unimodal pose estimate, our vehicle's planner expects

a single pose estimate and thus this equation has proven useful as it constitutes a practical compromize between the high bias of mean-filtering and the high variance of the mode.

In our vehicle this offset is computed and sent at 10 Hz. Subsequently, the vehicle is able to plan paths in global coordinates using the best possible estimate of its global position. An example of the localizer's effect is shown in Fig. 5.

IV. Experimental Results

The above algorithms were implemented in C such that they are capable of running in real time in a single core of a modern laptop CPU. Map data requires roughly 10 megabytes of data per mile of road, which enables extremely large maps to be stored on disk. Maps are stored in a tiled format so that data grows linearly with terrain covered, and RAM usage is constant regardless of map size.

We conducted extensive experiments, both manually and autonomously, with the vehicle shown in Fig. 1. We now present both quantatative results demonstrating the high accuracy of our localizer and discuss autonomous results we were unable to achieve without the present techniques.

A. Quantatative Results

In order to quantatatively evaluate the performance of our methods in the absense of known "ground truth," we employ the same offline GraphSLAM alignment described in the mapping section to align a recorded logfile against an existing map; we then compare the offsets generated in this alignment to the offsets reported by the localizer. Ideally, the offline and online methods should yield similar results, though the offline SLAM approach would likely have higher accuracy as it uses more than 100 times the amount of data. In fact, our probabilistic maps can be thought of as an efficient reduction of the entire set of laser data that, by virtue of storing the intensity variances, loses much less information than does a typical map.

For our first test, we drove around a very large urban block several times in July and used our mapping method to create a probabilistic map. We then collected a separate logfile in September of the vehicle driving the same block three times; this was approximately ten minutes of driving in dense traffic. At this point, the online localizer was used to align the September route against the July map. Separately, offline GraphSLAM was also used to align the two trajectories using all available data.

The map is shown in Fig. 2. Fig. 6 shows the lateral offsets applied by the localizer during the ten-minute September drive compared against the difference in the localizer alignment and the offline SLAM alignment. During this drive, an RMS lateral correction of 66cm was necessary, and the localizer corrected large errors of up to 1.5 meters. As the graph illustrates, the resulting error after localization was extremely low, with an RMS value of 9cm. It should be noted that this value is quite a bit less than the grid cell size of 15cm, and also that the GraphSLAM alignment itself is

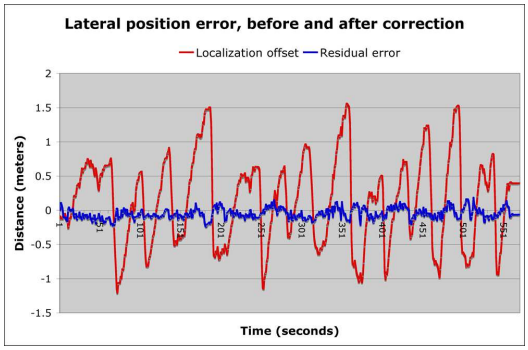


Fig. 6. Comparing the lateral offset applied by our algorithm (red) to the residual error after localization (blue) as measured by offline SLAM alignment. During these ten minutes of driving, RMS lateral error has been reduced from 66cm to 9cm.

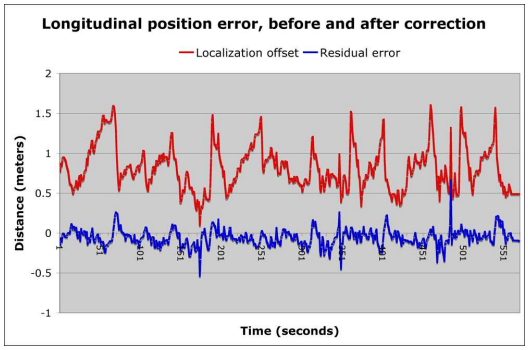


Fig. 7. Comparing the longitudinal offset applied by our algorithm (red) to the residual error after localization (blue) as measured by offline SLAM alignment. During these ten minutes of driving, RMS longitudinal error has been reduced from 87cm to 12cm. Note also the systematic bias in longitudinal error which is removed after localization.

likely to have minor errors; thus, this result is about as good as we could hope to achieve with this method of evaluation.

A companion graph for longitudinal corrections of the same drive is shown in Fig. 7; here, the localizer corrected an RMS longitudinal error of 87cm and agreed with the GraphSLAM alignment to within 12cm RMS. Interestingly, whereas the lateral corrections had roughly a zero mean, that is not the case for the longitudinal corrects. The fact that the average longitudinal correction was about 75cm forwards suggest that there was a systematic bias somewhere; perhaps the wheel encoder or GPS signal was systematically off during the mapping or localization run. In any case, the localizer was clearly able to correct for both the systematic and non-systematic effects with great success.

B. Autonomous Success

In addition to evaluating our performance quantatatively, we also ran several autonomous experiments in which the vehicle navigated autonomously in real urban environments. Using previously published localization methods, we were able to drive autonomously on moderately wide roads and only in low traffic, because turns could not be made with sufficient accuracy and narrow roads posed too great a risk.

However, using the methods presented here, we are now able to drive autonomously in several urban environments

that were previously too challenging. In one example, our vehicle participated in an autonomous vehicle demonstration in downtown Manhattan in which several blocks of 11th Avenue were closed to regular traffic. Our vehicle operated fully autonomously with other autonomous and human-driven vehicles and succesfully stayed in the center of its lane, never hitting a curb or other obstacle, despite that the environment configuration had changed considerably since the map had been acquired.

In another example, we mapped a four-mile loop around a local campus that includes roads as narrow as 10 feet, tight intersections, and speed limits up to 40 MPH. We were able on the first attempt to drive the entire loop with the vehicle completely controlling its own steering; an intervention was never necessary even amidst heavy rush-hour traffic. This route is depicted in Fig. 8.

Since performing these localization-specific experiments, the algorithms presented in this paper have already enabled our vehicle to drive several hundred miles autonomously in traffic on urban roads without a single localization-related failure.

Fig. 8. A four-mile loop around which our vehicle nagivated autonomously in traffic with no localization failures.

V. Conclusion

Localization is a critical enabling component of autonomous vehicle navigation. Although vehicle localization has been researched extensively, no system to our knowledge has yet proven itself able to reliably localize a vehicle in dense and dynamic urban environments with sufficient accuracy to enable truly autonomous operation. Previous attempts have successfully improved upon GPS/IMU systems by taking environment into account, but suffered from an inability to handle changing environments. While our approach is not infinitely adaptable and can be hindered by sufficiently severe changes in weather or environmental configuration, we believe it is a significant step forward towards allowing vehicles to navigate themselves in even the trickiest of real life situations.

By storing maps as probability models and not just expected values, we are able to much better describe any environment. Consequently, localizing becomes more robust and more accurate; compared with previous work we suffer far fewer complete localization failures and our typical localization error is significantly reduced, especially in the more challenging longitudinal direction. The fact that we were able for the first time to autonomously nagivate our vehicle in our most difficult local streets (with lanes as narrow as 10 feet), amidst rush-hour traffic, is a testament to the precision and robustness of our approach. Indeed, extensive experiments suggest that we are able to reduce the error of the best GPS/IMU systems available by an order of magnitude, both laterally and longitudinally, thus enabling decimeter-level accuracy; as we have shown, this performance is more than sufficient for autonomous driving in real urban settings.

There remain promising areas for further research on this topic. In the interest of efficiency and robustness we project points to the *xy* plane in our maps, but height information is surely useful; while infrared reflectivity is abundantly rich in information, a more complex approach could also reason in the space of map elevation. For example, ray tracing could be used in both the map-making and localization phases to explicitly process and remove dynamic obstacles, and vertical static obstacles could be incorporated into the measurement model.

References

[1] T. Duckett, S. Marsland, and J. Shapiro. Learning globally consistent maps by relaxation. ICRA 2000.

[2] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *IJRR*, 23(12), 2004.

[3] E.D. Dickmanns. Vision for ground vehicles: history and prospects. *IJVAS*, 1(1) 2002.

[4] J. Levinson, M. Montemerlo, S. Thrun. *Map-Based Precision Vehicle Localization in Urban Environments*. RSS, 2007.

[5] J. Folkesson and H. I. Christensen. Robust SLAM. *ISAV* 2004.

[6] U. Frese, P. Larsson, and T. Duckett. A multigrid algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, 2005.

[7] S. Thrun, W. Burgard and D. Fox. Probabilistic Robotics. *MIT Press*, 2005.

[8] S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *IJRR*, 25(5/6), 2005.

[9] T.D. Gillespie. Fundamentals of Vehicle Dynamics. *SAE Publications, 1992.*

[10] W. Burgard and R. Triebel, H. Andreasson. Improving Plane Extraction from 3D Data by Fusing Laser Data and Vision. *Intelligent Robotics and Systems*, 2005.

[11] B. Lamond and G. Watson. Hybrid rendering - a new integration of photogrammetry and laser scanning for image based rendering. *Proc. of Theory and Practice of Computer Graphics(TPCG)*, 2004.

[12] B. R. Harvey and D. D. Lichti. The effects of reflecting surface material properties on time-of-flight laser scanning measurements. *International Society for Photogrammetry and Remote Sensing*, 2002.

[13] P. Debevec et al. Estimating Surface Reflectance Properties of a Complex Scene under Captured Natural Illumination. Technical Report ICTTR-06.2004, University of Southern California Institute for Creative Technologies Graphics Laboratory, 2004.

[14] C. Urmson, J. Anhalt, M. Clark, T. Galatali, J.P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, P.L. Koon, K. Peterson, B.K. Smith, S. Spiker, E. Tryzelaar, and W.L. Whittaker. High speed navigation of unrehearsed terrain: Red Team technology for the Grand Challenge 2004. TR CMU-RI-TR-04-37, 2004.

[15] DARPA. DARPA Grand Challenge rulebook, 2004. On the Web at http://www.darpa.mil/grandchallenge05/Rules_8oct04.pdf.

[16] DARPA. DARPA Urban Challenge rulebook, 2006. On the Web at http://www.darpa.mil/grandchallenge/docs/Urban_Challenge_Rules_121106.pdf.